ДАРНИЦЬКА РАЙОННА В М. КИЄВІ ДЕРЖАВНА АДМІНІСТРАЦІЯ

УПРАВЛІННЯ ОСВІТИ

ЦЕНТР ТЕХНІЧНОЇ ТВОРЧОСТІ ТА ПРОФЕСІЙНОЇ ОРІЄНТАЦІЇ ШКІЛЬНОЇ МОЛОДІ



**ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ
МОВОЮ ПАСКАЛЬ**

***Практикум***

 Автори: керівник гуртка «Програмування»

 Стеценко Антоніна Іванівна;

 керівник гуртка «Програмування»

 Зінзюк Олена Вячеславівна

 Київ - 2018

Зміст

[1. Вступ 3](#_Toc514159415)

 [1.1. Предмет «Інформатика» та його основні напрями в гуртковій роботі…. 3](#_Toc514159416)

[1.2. Від технологій до творчості 7](#_Toc514159417)

[1.3. Інструменти для навчання 8](#_Toc514159418)

[2. Основна частина 10](#_Toc514159419)

[2.1. Збірник презентаційних програм 10](#_Toc514159420)

[2.2. Практичні роботи 10](#_Toc514159421)

[2.3. Збірник задач з розв’язками 11](#_Toc514159422)

[2.4. Збірник задач для самостійного виконання. 11](#_Toc514159423)

[3. Комплекс тестування ADSoft Tester для перевірки теоретичних знань вихованців 13](#_Toc514159424)

[Висновки 16](#_Toc514159425)

[Література 17](#_Toc514159426)

[Додаток А 2](#_Toc514159427)4

[Додаток Б 50](#_Toc514159428)

Додаток В…………………………………………………………………………...56

Додаток Г……………………………………………………………………………99

# **Вступ**

**«**Програміст  повинен  володіти  здібністю  першокласного  математика до абстракції та логічного мислення у поєднанні з талантом Едісона споруджувати все, що завгодно, з нуля і одиниць. Він повинен поєднувати акуратність бухгалтера з прозорливістю розвідника, фантазію автора детективних романів – з тверезою практичністю економіста. А крім того, програміст повинен мати смак до колективної роботи, розуміти інтереси користувача і багато чого іншого… »

***Андрій Петрович Єршов***

# **Предмет «Інформатика» та його основні напрями в гуртковій роботі**

Майже 30 років тому в загальноосвітніх навчальних закладах з’явився новий обов’язковий для вивчення предмет «Інформатика». За цей час від лозунгу «Програмування – друга грамотність» прийшли до кардинально відмінного підходу у викладанні інформатики: «Основне – компетентний користувач, а програмування – для вузьких спеціалістів». Останнім часом програмування замінили на вивчення «як зареєструвати електронну пошту та відправити email». По суті, набір молодого клерка. До недавнього часу дітей вчили просто користуватися комп’ютером: заходити в Інтернет, копіювати файли, редагувати документи в Word, Excel і таке інше.

Однією з перешкод в розвитку комп’ютерних наук та програмування є той факт, що суспільство та батьки сприймають програмування як суто чоловічу справу, нудну та позбавлену творчого елементу. Але це лише стереотип: програмування є дуже творчим ремеслом, що вимагає життєвого досвіду і розвиненої інтуїції. Цей усталений стереотип потрібно зламати, адже через нього потенційно хороші фахівці обирають інші галузі діяльності. Втім, останнім часом ситуація має тенденцію до змін на краще. Все більше жінок, а також колишніх офісних та банківських працівників примикають до рядів програмістів, в корені ламаючи усталений стереотип образу «програміста-ботаніка».

Дуже часто батьки задають питання: «Навіщо вчити мою дитину програмуванню?». Тим більше, якщо у них росте майбутній лікар або футболіст. Вміння програмувати розширює кругозір, розвиває логічне мислення та допомагає краще розуміти динамічний світ. Зрештою, починаючи писати або читати, ви не прагнете заробляти цим на життя. Дуже мало людей стає професійними письменниками, але писати й читати повинен вміти кожен. Те саме можна сказати і про програмування в XXI столітті.

Не зменшуючи ролі вивчення ІКТ, можна бути впевненим, що країна без ІТ-спеціалістів, обізнаних в алгоритмізації та програмуванні, у сучасному інформаційному світі приречена бути лише покупцем і користувачем, залежним від створеного кимось програмного продукту. Вже зараз з’являється все більше професій на стику програмування, математики, фізики, біоінженерії, економіки. Ми маємо підготувати наших дітей до майбутнього, в якому вони будуть жити.

Дітям подобається проводити багато часу з новими технологіями, але частіше за все їх можна застати за такими заняттями, як ігри, спілкування в чатах з друзями, пошук в Інтернеті та іншим активним користуванням. При цьому вони зовсім не орієнтовані на свідоме використання цих технологій та самореалізацію з їх допомогою. Це схоже на те, якби вони вміли читати, але не вміли писати.

Паралельно зі школами в закладах позашкільної освіти набули поширення гурткові заняття з інформатики.

Роботі з комп’ютером у гуртку можна легко надати цікавого ігрового характеру, тому такі заняття приваблюють дітей. При цьому, однак, потрібно мати на увазі, що роботу гуртка не можна будувати тільки на захопленні молодого покоління комп’ютерною грою.

Зазначимо основні напрями проведення гурткових занять з інформатики:

• основи роботи на комп’ютері для запуску прикладних програм навчального призначення та подальшого їх використання для розв’язування різних завдань;

• правила роботи з електронною поштою та браузерами для перегляду гіпертекстових сторінок, форумами для подальшої роботи в телекомунікаційних проектах;

• пошук інформації в глобальній мережі за допомогою використання пошукових систем;

• мова розмітки гіпертексту та створення веб-сторінок;

• принципи створення графічних зображень;

• основи алгоритмізації та програмування.

Гурткова робота дає можливість враховувати і розвивати здібності та нахили вихованців. У цьому одна із її переваг перед звичайними заняттями. Різноманітність пристроїв, які вивчаються та використовуються в гуртках, різний рівень складності і трудомісткості питань і завдань, необхідність широких теоретичних і практичних знань дають змогу шляхом продуманого добору тем задовольнити інтереси всіх членів гуртка незалежно від їхніх знань, досвіду і здібностей.

Базовою програмою для вивчення основ алгоритмізації та програмування в нашому закладі є навчальна програма з позашкільної освіти науково-технічного напряму інформаційно-технічного профілю «Основи програмування». Це спеціально модифікований варіант програми, рекомендованої Міністерством освіти і науки України (лист Міністерства освіти і науки України від 15.09.2014 №1/11-14728) для спецкурсів, факультативів, пропедевтичних курсів, гуртків, адаптованої до умов навчального закладу позашкільної освіти.

Сучасних дітей потрібно вчити основам алгоритмізації та програмування: знайомити з базовими алгоритмічними конструкціями, їх поєднаннях в програмах, написання програм рішення основних (фундаментальних) задач типу: мінімальне/максимальне (з 2-х, 3-х, в масиві, ... ); сортування (всі способи); робота з рядками; читання/запис у файл, тощо. На їх основі ‒ рішення задач з алгебри, геометрії, фізики (бажано за шкільною програмою, щоб показати переваги використання комп’ютерної програми). А мова реалізації великого значення не має (на початковому етапі знайомства з ОАП), головне –навчити дітей логічно думати і вміти поставити себе на місце виконавця своєї програми (це, мабуть, найголовніше).

Оволодівши необхідними знаннями та навичками в області алгоритмізації та програмування, у людини з’являється можливість створювати власні та вдосконалювати існуючі доробки у сфері інформаційних технологій. Виходячи з означення інноваційного потенціалу, алгоритмізація та програмування може використовуватись у освітньому процесі як один із засобів для підвищення інноваційного потенціалу особистості як педагога, так і вихованця.

Вивчення алгоритмізації може мати два цільових напрями:

* розвивальний напрям, під яким розуміється розвиток алгоритмічного мислення вихованців;
* програмістський напрям, що передбачає в собі вивчення технології створення програм.

Останній напрям можна розділити на два цільових аспекти:

* Перший аспект пов’язаний з посиленням фундаментальної компоненти курсу інформатики. Дітям дається уявлення про те, що таке мови програмування, що представляє собою програма мовою програмування високого рівня, як створюється програма в середовищі сучасної системи програмування.
* Другий аспект носить профорієнтаційний характер. Вивчення програмування в рамках гурткової роботи дозволяє вихованцям випробувати свої здібності до різного роду діяльності і, при бажанні, вибрати у майбутньому відповідний професійний шлях.

Алгоритмізація відповідає методу структурного програмування і є підготовчим етапом до вивчення об’єктно-орієнтованого програмування, актуального на сучасному етапі розвитку програмування.

Стандартна програма вивчення основ алгоритмізації передбачає наступну послідовність тем:

* складання лінійних алгоритмів;
* складання циклічних алгоритмів;
* використання розгалужень в алгоритмах;
* опис і використання допоміжних алгоритмів.

На початковому етапі вивчення певної структури алгоритму доцільно використовувати блок-схеми, які наочно демонструють базові структури алгоритмів та дають можливість сформувати правильну уяву про механізм роботи кожної із них.

## **Від технологій до творчості**

Вміння програмувати ‒ це набагато більше, ніж просто технічні навички. Написання програмних кодів розвиває аналітичне мислення, вчить мислити логічно, розвиває уяву, креативність. Діти вчаться розуміти, як влаштований світ, встановлюють логічний ланцюжок подій і можуть передбачити, що буде далі. Чим раніше починається навчання, тим винахідливішою і креативнішою стає людина. Декому це дивно чути, але в майбутньому всі будуть навчатись програмуванню. Як сказав Джон Дьюї, американський філософ, психолог та реформатор освіти, «Якщо ми будемо вчити сьогодні так, як вчили вчора, ми вкрадемо в наших дітей завтра». Ми маємо допомогти нашим дітям стати активними, а не пасивними користувачами технологій, а також розвинути знання, які потрібні в XXI столітті. Необхідно ламати застарілі стереотипи та змінювати країну разом.

## **Інструменти для навчання**

На сьогоднішній день існує велика кількість навчальних мов програмування, кожна з яких володіє своїми перевагами. Однак, яку б мову програмування не обрав керівник гуртка інформатики для вивчення в закладі позашкільної освіти, він має розуміти, що вибір має ґрунтуватися, перш за все, на перспективних, динамічно розвиваючих напрямах науки і техніки, на розвитку в вихованцях інженерно-технічного, креативного та логічного мислення.

Вибір мови програмування однозначно визначений завданнями щодо набуття компетентності щодо мовлення й опанування алгоритмів. Наразі такою мовою програмування для початківців може бути Паскаль. Основну увагу потрібно приділяти змістовому наповненню задач і їхній складності, а не вивченню структур мови програмування.

Проте останнім часом мова Паскаль стала непопулярною, адже використовується лише для навчання учнів та студентів і вважається базовою. Паскаль може використовуватися для написання реальних програм та ігор. Раніше він мав достатню популярність, але зараз зберігся лише, по суті, для навчальних цілей. Перше, що привертає потенційних програмістів в Паскалі, ­ це можливості писати, зберігаючи властивості англійської мови, а не ламати голови над використанням спеціальних схем, які прийняті в мовах, подібних С.

**Перевагами мови програмування *Паскаль* є**:

* простий синтаксис мови – програми на Паскалі легко читати та редагувати, оскільки вони містять невелику кількість базових понять;
* досить низькі апаратні та системні вимоги як для роботи самого середовища програмування, так і для програм, записаних мовою Паскаль;
* універсальність мови, тобто можливість використовувати цю мову програмування для розв’язування задач різних класів та сфер діяльності;
* підтримка структурного та об’єктно-орієнтованого програмування.

Розпочати програмувати на Паскалі досить просто. Найпростіша програма складається всього з трьох рядків і не потребує введення складних понять. Мабуть, найбільшою перевагою цієї мови є той факт, що це мова програмування зі строгими правилами при використанні різних типів даних. Це означає, що середовище програмування не дозволить використовувати рядкову величину для присвоєння числовій змінній без явної вказівки програміста. Таку помилку можна буде побачити ще на етапі компіляції.

Вивчення розділу «Основи алгоритмізації та програмування» з курсу «Інформатика» відбувається в такій послідовності: алгоритми та їх виконавці; алгоритми з повторенням і розгалуженням; алгоритми роботи з величинами; табличні величини та алгоритми їх опрацювання; складні структуровані типи даних (множини, записи, файли). На першому рівні вивчення передбачається формування алгоритмічного мислення вихованців і розробка алгоритмів, на другому рівні – ознайомлення з конкретною мовою програмування та реалізація алгоритмів в цьому програмному середовищі, тобто здійснення переходу до вивчення мови програмування Паскаль, так як у ньому багато концепцій програмування свідомо спрощені, що дозволяє використовувати його на більш ранніх етапах навчання.

Як показує практика, навички, отримані під час вивчення Паскаля дозволяють легко перейти на будь-яку сучасну мову програмування.

# **2. Основна частина**

## **2.1. Збірник презентаційних програм**

Для оптимізації навчального процесу та зручності викладання матеріалу створено електронні презентації в середовищі PowerPoint інтегрованого пакету MSOffice 2016. До збірника входять такі роботи:

* лінійні програми;
* розгалуження;
* цикли;
* масиви одновимірні;
* масиви двовимірні;
* підпрограми;
* рядки;
* множини;
* записи;
* файли.

Всі презентації виконані в одному стилі та за однаковою структурою, що полегшує сприйняття й засвоєння матеріалу. Після оголошення теми заняття здійснюється актуалізація опорних знань вихованців – перевіряється рівень їх теоретичних знань, пропонується знайти помилки у фрагментах кодів програми, запропоновано виконати низку вправ на закріплення матеріалу. Після чого приводяться коди програм, в яких детально реалізуються алгоритми різного ступеня складності, а також на екрані представляється результат роботи даної програми.

Перейти до папки «Презентації практикум» можна за посиланням [презентації\_практикум](../%D0%9F%D1%80%D0%B5%D0%B7%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D1%96%D1%97_%D0%BF%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%BA%D1%83%D0%BC).[[1]](#footnote-1)

Один із зразків такої презентації можна переглянути в *Додатку А*.

## **2.2. Практичні роботи**

Після перегляду презентації вихованці отримують практичну роботу з даної теми у вигляді друкованого документу. Всі практичні роботи оформлені в одному стилі і мають таку структуру: тема та мета даної практичної роботи, коротке викладення теоретичного матеріалу, підбірка задач, причому кожен вихованець отримує свою задачу зі списку запропонованих в залежності від ступеня своїх можливостей і підготовки ­ вибір здійснює керівник гуртка, використовуючи індивідуальний диференційований підхід до кожної дитини.

Перейти до папки «Практичні роботи» можна за посиланням [практичні\_роботи](../%D0%9F%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D1%87%D0%BD%D1%96_%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B8).

Практичні роботи в друкованому вигляді подані в *Додатку Б*.

## **2.3. Збірник задач з розв’язками**

 В даному розділі представлено підбірку задач з розв’язками. Задачі мають прикладний характер, різний рівень складності та відносяться до різних розділів навчальної програми, а саме: цілочисельна арифметика, проста математика, довга арифметика, оператори розгалуження, циклічні оператори, робота з масивами та файлами. Кожна задача містить такі характеристики: (Оптимальний час вирішення задачі, пам’ять, необхідна для виконання даної задачі, складність алгоритму. У всіх задачах вхідні дані зчитуються з текстового файлу, а також результат роботи програми записується до файлу.

Перейти до папки «Задачі з розв’язками» можна за посиланням [задачі\_розв’язки](../%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D1%96%20%D0%B7%20%D1%80%D0%BE%D0%B7%D0%B2%E2%80%99%D1%8F%D0%B7%D0%BA%D0%B0%D0%BC%D0%B8).

Задачі з розв’язками подані в *Додатку В.*

## **2.4. Збірник задач для самостійного виконання.**

В цьому розділі вихованцям пропонуються задачі для самостійного виконання. Задачі розділені на такі розділи, які відповідають основним темам навчальної програми:

* лінійні програми;
* розгалуження;
* цикли;
* масиви одновимірні;
* масиви двовимірні;
* підпрограми;
* рядки;
* множини;
* записи;
* файли.

Задачі мають різний рівень складності, носять прикладний характер та сприяють закріпленню знань вихованців з основ алгоритмізації та програмування.

Перейти до папки «Збірник задач для самостійного виконання» можна за посиланням [збірник\_задач.](../%D0%97%D0%B1%D1%96%D1%80%D0%BD%D0%B8%D0%BA%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%20%D0%B4%D0%BB%D1%8F%20%D1%81%D0%B0%D0%BC%D0%BE%D1%81%D1%82%D1%96%D0%B9%D0%BD%D0%BE%D0%B3%D0%BE%20%D1%80%D0%BE%D0%B7%D0%B2%E2%80%99%D1%8F%D0%B7%D0%BA%D1%83)

Задачі з для самостійного виконання подані в *Додатку Г.*

(*Повний перелік задач з усього курсу можна знайти за вказаним раніше посиланням*).

# **Комплекс тестування ADSoft Tester для перевірки теоретичних знань вихованців**

Комплекс тестування ADSoft Tester - це вільно поширювана програма. Сайт розробників програми <http://www.adtester.org.>  Комплекс складається з трьох частин - Адміністратор, Редактор тестів, Тестер.

**Можливості пакету програм:**

* необмежена кількість запитань у тесті та кількість відповідей у ньому;
* п’ять типів запитань: одиночний та множинний вибір, введення відповіді з клавіатури, відповідність, порядок;
* два режими роботи: контроль та навчання;
* кожне запитання має свою вагу в балах;
* кожне запитання може мати підказку, яка з’являється при неправильній відповіді в режимі навчання;
* використання шрифтів, кольорів тексту, стилів;
* запитання можуть містити зображення, таблиці, діаграми, різні OLE-документи;
* тестування може бути обмежене в часі;
* настроювана шкала оцінювання для кожного тесту;
* файл тесту зашифрований, його не можна прочитати звичайним текстовим редактором. Редагування тесту може бути захищене паролем.

Посилання на [ADSoft Tester](../ADTester) (запуск тесту за допомогою додатку ***tester***).



 Програма Адміністратор зберігає списки вихованців гуртка, результати тестування, робить аналіз результатів.



Конструктор тестів – створює тести.



Програма для тестування



# **Висновки**

Даний практикум може бути використаний керівниками гуртків інформатики, які працюють з вихованцями за напрямом «Основи алгоритмізації та програмування». Передбачається, що спочатку педагог проводить теоретичне заняття з конкретної теми, потім – презентація-практикум, яка допомагає закріпити набуті знання (актуалізація опорних знань, вправи та завдання на пошук помилок у фрагментах коду програми), далі ­ розгляд конкретних прикладних задач (алгоритм і код програми), аналіз результатів. Після цього вихованцям можна запропонувати пройти тест ADSoft Tester на знання теоретичного матеріалу з даної теми.

Другим етапом є виконання практичної роботи з даної теми. Кожен вихованець отримує друкований текст з ходом роботи, а також коротким викладом теоретичних знань та задачі, які викладач розподіляє між дітьми індивідуально за різними рівнями складності.

Вдома у вільний час, на канікулах вихованці можуть виконувати завдання для самостійного розв’язку, які можна отримати у педагога в друкованому або електронному вигляді. Це закріплює знання та навички, здобуті на заняттях. Якщо в процесі роботи над цими завданнями виникають питання, на допомогу можуть прийти задачі з готовими розв’язками, які також можна отримати у керівника гуртка.

# **Література**

1. Глинський Я. М. Паскаль. TURBOPASCAL. DELPHI : навч. пос. для студ. / Я. М. Глинський, В. Є. Анохін, В. А. Ряжська. – Львів : « 2002», – 144 с.
2. Каліон В. А. Основи інформатики. Структурне програмування на Паскалі : практикум / В. А. Каліон, О. І. Черняк, О. М. Харитонов. – Київ : Центр учбової літератури, 2007. – 248 с.
3. Скляр І.В. Базовий курс програмування для фізико-математичних шкіл : навчальний посібник / І. В. Скляр, А. Б. Ставровський. – Київ : «Агентство «Україна», 2014. – 272 с.
4. Караванова Т. П. Інформатика. Основи алгоритмізації та програмування : навчальний посібник / Караванова Т. П. – Київ : «Ґенеза», 2006. – 286 с.

# Додаток А

# **Презентації\_Практикум**

#

#

#

#















# Додаток Б

**Збірник практичних робіт**

**Практична робота № 1**

 **«Створення лінійних програм. Створення найпростіших діалогових програм із використанням команд вводу-виводу»**

**Мета**:

1. Закріплення навичок роботи у середовищі програмування.
2. Розвиток інформаційної компетентності (навичок аналізу та локалізації помилок).
3. Формування компетентності саморозвитку та самоосвіти.

 **Хід роботи:**

***Теоретична частина***

***Структура програми***

***Program<ім’я програми>;***

***Uses< ім’я1, ім’я2,... >; {список використовуваних бібліотечних модулів }***

***Label<мітка1, мітка2,...>; {розділ міток}***

***Const<ідентифікатор=значення>; {розділ констант}***

***Type<ідентифікатор=опис типу>; {розділ типів}***

***Var< ідентифікатор:тип >; {розділ змінних}***

***Procedure< ім’я підпрограми >; {розділ процедур}***

***Function< ім’я функції >; { та функцій}***

***Begin***

***<оператор1>;***

***... {серія команд}***

***<операторN>***

***End.***

***Оператори введення та виведення на мові Pascal***

***Загальний вигляд процедури введення інформації:***

* ***Read(<список елементів вводу>);***

 ***або***

* ***Readln(<список елементів вводу>);***
* ***Елементами вводу можуть бути одна або кілька змінних, записаних через кому.***
* ***Значення, що вводяться відокремлюються один від одного пробілами або розміщуються на різних рядках (при натискуванні Enter).***
* ***Readln відрізняться від Read тим, що після її виконання автоматично відбувається перехід до чергового рядка даних, тобто наступна процедура вводу буде зчитувати дані з нового рядка.***

***Загальний вигляд процедури виведення інформації:***

* ***Write(<список елементів виводу>);***

 ***або***

* ***Writeln(<список елементів виводу>);***

***Список виведення може містити:***

* ***ідентифікатори величин, значення яких будуть виведені на екран;***
* ***вирази, значення яких спочатку будуть обчислені, а потім виведені на екран***
* ***Константи - числові та символьні (беруться в лапки)***
* ***Використання процедури Writeln без параметрів забезпечує пропуск порожнього рядка при виведенні результатів.***

***Форматування при виведенні результатів:***

***<ім’я змінної>:n:m;***

***n – загальна кількість позицій, що надається дійсному числу при виведенні на екран, враховуючи знак числа і десяткову крапку;***

***m – кількість позицій, що відводиться під дробову частину.***

***Наприклад,***

***Writeln(f:10:6);***

 ***Writeln(d:6:3,g:7:4);***

***Актуалізація опорних знань:***

1. Для чого в Паскалі служить крапка з комою?

* 1. Чим '7' відрізняється від 7?
	2. Який тип даних потрібно використати для запису вартості товару в гривнях?
	3. У чому полягає відмінність між константою й змінною?
	4. Чи завжди зберігає змінна своє значення в ході виконання програми?
	5. Чи правильний запис:

**const a=25;**

**…**

**a=a+1;**

7. Чи можна використати різні типи даних в одній операції? Якщо так, то як це відіб'ється на результаті операції?

8. Чи правильно зроблено перестановку значень двох змінних:

**…**

**а:=12; b:=21; a=b; b=a;**

***Практична частина***

1. У квадрат вписане коло. Визначити площу заштрихованої частини фігури. Довжину сторони квадрата увести з клавіатури.



1. Дано трицифрове число. Знайти число, отримане при прочитанні його цифр у зворотному порядку.
2. Квіткова клумба має форму круга. Обчислити її периметр і площу за радіусом, значення якого вводиться з клавіатури.
3. Обчислити периметр трикутника за сторонами, довжини яких вводяться з клавіатури.
4. Обчислити довжину кола і площу круга за діаметром, значення якого вводиться з клавіатури.
5. Ділянка лісу має форму рівнобічної трапеції. Обчислити її периметр за заданими сторонами, які вводяться з клавіатури.
6. Ресторан закуповує щодня масло **m кг** по **а грн**. за кілограм, сметану - **n кг** по **b грн.,** вершки - **f кг** по **c грн**. Визначити суми, потрібні для купівлі окремих продуктів, і загальну суму.
7. Скільки секунд мають доба, тиждень, рік?
8. Місто А знаходиться в ***х*** милях від Нью-Йорку. Напишіть програму, яка обчислює цю відстань в кілометрах.
9. Петрик запропонував своє тлумачення «щасливого» квитка. Для нього він «щасливий» тільки тоді, коли сума його цифр ділиться на 5. Перевірити, чи куплений Петриком квиток є щасливим.
10. Складіть програму розв'язування рівняння виду ax+b=0**.**
11. Визначити площу прямокутника, якщо відомо, що одна сторона більша за другу на х см, а периметр дорівнює р см. Вказівка: після спрощення формула для визначення площі прийме вигляд S=(p2-x2)/16.
12. Знайти суму перших n членів арифметичної прогресії {An}, якщо відомі А***1*** і D.
13. Хлопчик, що торгує на базарі газетами, заробляє а грн. від продажу перших 100 газет. На кожній з інших проданих газет він заробляє по х коп. Напишіть програму, яка визначає заробіток за день хлопчика, якщо він продасть у газет (у>100).
14. Реактивний літак перевозить групу футбольних болільників на чемпіонат світу з Києва до Парижу. Три чверті пасажирів мають білет другого класу вартістю х доларів кожен. Всі інші пасажири мають квитки першого класу, які вдвічі дорожчі від білетів другого класі. Напишіть програму, яка виводить суму (в гривнях), що отримує авіакомпанія від продажу квитків на цей рейс.

**Практична робота № 2**

**«Створення програм з розгалуженням та вибори варіантів»**

**Мета**:

1. Закріплення навичок роботи у середовищі програмування.
2. Розвиток інформаційної компетентності (навичок аналізу та локалізації помилок).
3. Формування компетентності саморозвитку та самоосвіти, виховувати інформаційну культуру вихованців, уважність та відповідальність

**Хід роботи:**

***Теоретична частина***

* ***Оператор розгалуження в Паскалі має вигляд (повний оператор):***

***IF<логічний вираз>THEN<оператор 1>ELSE<оператор 2>;***

***(IF- якщо, THEN- то, ELSE- інакше – службові слова)***

* ***Оператор розгалуження в Паскалі має вигляд (неповний оператор):***

***IF<логічний вираз>THEN<оператор 1> ;***

* ***Якщо логічний вираз істинний, то виконується ні оператор 1, якщо – то оператор 2.***
* ***Якщо необхідно після службового слова записати більше одного оператора, то необхідно обмежити цю послідовність команд службовими словами BEGINEND.***

***Умова (логічний вираз) – вираз булівського типу, який може приймати одно з двох значень: ”так” або “ні”.***

* ***Умова може бути простою або складеною.***
* ***Проста умова утворюється за допомогою операцій відношення***
* ***(=,<>, >=, <=, >, <)***
* ***Наприклад, Х<0 В=7***

***Складені умови утворюються з простих за допомогою логічних операцій:***

* ***and (необхідне виконання всіх умов)***
* ***or (необхідне виконання хоча б однієї умови)***
* ***not (заперечення умови).***
* ***Наприклад: (X>4) and (X<10)***

***Конструкція “Вибір” узагальнює конструкцію “Розгалуження” у випадку, коли є багато шляхів розгалуження.***

***Загальний вигляд команди:***

***CASE<вираз>OF***

***<список значень 1>:<команда 1>;***

 ***.......***

***<список значень N>:<команда N>***

***ELSЕ < команда N+1>***

***END;***

* ***Дія команди. Якщо значення виразу збігається зі значенням з деякого списку, то виконується відповідна команда, а інші команди не виконуються. Список значень може складатися з одного чи кількох елементів, які перераховуються через кому.***
* ***Складової частини ELSЕ може не бути.***
* ***Вираз – це проста змінна цілого, символьного, переліковного або логічного типу.***
* ***Не можна використовувати змінні дійсного (real) та рядкового (string) типів.***
* ***Списки значень – сталі або діапазони, тип яких збігається з типом виразу.***
* ***В списку вибору треба вживати складений оператор Begin…End, коли виконується група команд***

Вкладене розгалуження дозволяє перевірити дані за більш, ніж двома умовами

При вкладеному розгалуженні аналогічно послідовному спочатку перевіряється перша умова, але за результатом перевірки цієї умови може бути обраний як оператор, так і перевірка іншої умови.

***Актуалізація опорних знань:***

1. Що таке умовний вираз (умова)? Для чого потрібні умовні вирази?
2. Який тип мають умови в умовних конструкціях?
3. Описати загальний вигляд і виконання інструкцій розгалуження.
4. Що таке складена інструкція?
5. Якими операціями записуються складені умови?
6. Описати загальний вигляд і виконання інструкції вибору варіантів.
7. Чи може селектор варіантів мати дійсний тип?

***Практична частина***

1. Скласти програму знаходження більшого серед: а) двох чисел; б) трьох чисел; в) чотирьох чисел.
2. На площині задано трикутник координатами своїх вершин. Знайти найменшу сторону трикутника.
3. Точку А задано на площині своїми координатами (X,Y). Знайти номер координатної чверті, якій належить дана точка.
4. Дві точки на площині А (X1,Y1 )і В(X2,Y2)задано своїми координатами. Перевірити, чи лежать ці точки в одній координатній чверті.
5. Дано дійсні додатні числа X, Y, Z**.** Чи існує трикутник з довжинами сторін X, Y, Z.
6. Три точки на площині задано своїми координатами (X**1,**Y**1**),(X**2,**Y**2**),(X**3,**Y**3**). Знайти радіус найменшого кола (з центром в початку координат), якому належать данні точки.
7. Дано коло радіусом R і прямокутник зі сторонами A**,** B. Визначити, чи поміститься коло в прямокутнику.
8. Дано натуральне число N(N < 100), що визначає вік людини в роках. Після цього числа потрібно дописати одне з слів: "рік", "роки" або "років". Наприклад: 1 рік, 23 роки, 97 років.
9. Три точки на площині задано координатами A(X**1,**Y**1**),B(X**2**,Y**2**),C(X**3,**Y**3**). Чи належить початок координат трикутнику ABC.
10. Задано розміри прямокутних дверей А**,** B та розміри шафи, що має форму прямокутного паралелепіпеда X**,**Y**,**Z. Перевірити, чи можна шафу пронести через двері. а) Проносити шафу через двері дозволяється лише так, щоб кожне з її ребер було паралельно або перпендикулярно кожній з сторін дверей. б) Проносити шафу крізь двері можна під кутом.
11. На площині в прямокутній системі координат задано три точки Вважаючи, що вони є вершинами деякого прямокутника, знайти координати четвертої вершини.

**Практична робота № 3**

**«Створення програм з використанням різних видів циклів»**

**Мета**:

1. Закріплення навичок роботи у середовищі програмування.
2. Розвиток інформаційної компетентності (навичок аналізу та локалізації помилок).
3. Формування компетентності саморозвитку та самоосвіти.

**Хід роботи:**

***Теоретична частина***

***Цикли ­–– це оператори, які можуть повторювати дію декілька разів.***

 ***Для організації повторень призначені операторами: For, Repeat і While.***

***WHILE –– вказівка повторення з перед умовою***

***REPEAT –– вказівка повторення з після умовою***

***FOR –– вказівка повторення з параметром***

***While –– оператор циклу з передумовою.***

 ***Формат оператора:***

 ***While< умова >do***

 ***Begin***

 ***оператор 1; оператор 2; …***

 ***end;***

***На початку оператора вводиться умова. Поки вона вірна, оператор виконується. Якщо умова невірна з самого початку, то цикл не виконається жодного разу. Цей оператор зручно використовувати при роботі з файлами і якщо кількість повторень наперед невідома.***

***Repeat – оператор циклу з післяумовою.***

***Формат оператора:***

***Repeat***

***оператор 1; оператор 2; …***

***Until<умова>;***

***Repeat – оператор циклу з після умовою. Умова вводиться в кінці циклу. Якщо умова не вірна з самого початку, то цикл виконається один раз.***

***For – оператор циклу з параметром.***

***Для використання оператора в програмі треба записати:***

***For<лічильник> := <мін знач лічильника>to<макс. знач. ліч.>do***

 ***Begin***

 ***оператор 1; оператор 2; …***

 ***end;***

***Оператор For збільшує (to) чи зменшує (downto) лічильник на 1, доки його значення менше максимального значення і виконує оператори, записані після DO.***

***Цей оператор дуже зручно використовувати для роботи з масивами і тоді, коли відома кількість повторень.***

***Актуалізація опорних знань:***

1. Назвіть типи інструкцій циклу.
2. Опишіть загальний вигляд інструкцій циклу в мові Паскаль.
3. Яким може бути тип параметру **for**-циклу.
4. У чому різниця між інструкціями **for-to** і **for-downto**?
5. У чому полягає різниця між інструкціями циклу з відомою і невідомою кількістю повторів?
6. Якими можуть бути причини зациклювання програми?
7. Поясніть різницю між інструкціями циклу з передумовою та з постумовою.

***Практична частина***

1. Створити програму, яка виводить на екран квадрати чисел від 10 до 20.
2. Створити програму, яка обчислює суму чисел от 1 до **N**. Значення **N** вводиться з клавіатури.
3. Створити програму, яка обчислює добуток чисел от 1 до **N**. Значення **N** вводиться з клавіатури.
4. З клавіатури вводяться **N** чисел. Створити програму, яка визначає кількість від’ємних, додатніх та кількість нулів серед введених чисел. Значення **N** вводиться з клавіатури.
5. В ощадкасу на трьохпроцентний вклад поклали **S** гривень. Якою стане сума вкладу через **N** років?
6. Ввести с клавіатури 10 пар чисел. Порівняти числа в кожній парі й надрукувати більше з них.
7. Дані натуральні числа від 20 до 50. Надрукувати ті з них, які діляться на 3, але не діляться на 5.
8. Знайти двозначне число, якщо воно в три рази більше суми своїх цифр.
9. Знайти всі двозначні числа, сума цифр яких менша за їх добуток на 3.
10. Знайти всі трьохзначні числа, в яких сума цифр, збільшена на 2, в 4 рази менша їх добутку.
11. **Організувати безперервне введення чисел з клавіатури, доки користувач не введе 0. Після введення нуля вивести на екран кількість чисел, які були введені, їх суму та середнє арифметичне. Необхідно оголосити змінну–лічильник, яка буде рахувати кількість введених чисел, та змінну, яка буде накопичувати суму.**
12. **Необхідно сумувати всі непарні цілі числа в діапазоні, який вводиться з клавіатури.**

**Практична робота № 4**

**«Створення програм з використанням одновимірних масивів»**

**Мета**:

1. Закріплення навичок роботи у середовищі програмування.
2. Розвиток інформаційної компетентності (навичок аналізу та локалізації помилок).
3. Формування компетентності саморозвитку та самоосвіти.

**Хід роботи:**

***Теоретична частина***

* ***Масив – це структура даних, що являє собою однорідну (за типом), фіксовану (за розміром) сукупність елементів, упорядкованих за номерами.***
* ***Масив визначається ім’ям (ідентифікатором), кількістю елементів та індексами (номерами), що потрібні для вказівки місцезнаходження необхідного елемента масиву.***

***Опис одновимірного масиву в розділі Var :***

* ***Var<ім’я масиву>: array[нижня\_межа..верхня\_межа] of<тип елементів>;***
* ***Наприклад:***
* ***Var а:array[1..100] of integer;***
* ***b:array[1..10] of real;***
* ***Найпрактичніше здійснювати опис так:***
* ***Const n=30;***
* ***Var t:array[1..n] of integer;***

***До окремого елемента масиву можна звертатися за допомогою індекса. У якості індексів можуть використовуватись константи та змінні порядкових типів.***

***Одновимірний масив (вектор) – це масив, у якому елементи нумеруються одним індексом.***

* ***Задання масиву типізованою константою: const x: array[1..5] of integer=(1,3,5,6,2);***
* ***Попередній опис типу масиву:***

***type<ім’я типу>= array[нижня\_межа..верхня\_межа] of<базовий тип елементів>;***

***var<ім’я масиву>:< ім’я типу >;***

***Наприклад,***

***type t\_vector=array[1..100] of real;***

 ***var mas:t\_vector;***

***Заповнення даними одновимірного масиву***

* ***З клавіатури:***
* ***for i:=1 to n do read(a[i]);***
* ***for i:=1 to n do readln(a[i]);***
* ***for i:=1 to n do***

 ***begin write(‘введіть а[’,i,’]:’); readln(a[i]) end;***

* ***Випадковим чином:***

***randomize; {ініціалізація генератора випадкових чисел}***

***For i:=1 to n do a[i]:=random(100);***

* ***Через оператор присвоєння:***

***Var p:array[1..3] of integer;***

***Begin***

***p[1]:=100; p[2]:=200; p[3]:=300;***

***З використанням формул:***

***For i:=1 to n do a[i]:=100\* i-1/ i;***

***Виведення масиву***

* ***У рядок через пропуск-розділювач:***

***For i:=1 to n do write(a[i],’ ’);***

* ***У стовпчик:***

 ***For i:=1 to n do writeln(a[i]);***

* ***З заданням формату:***

***For i:=1 to n do write(a[i]:4);***

***Актуалізація опорних знань:***

* Що таке масив? Як він оголошується?
* Яким може бути тип елементів масиву?
* Яким може бути тип індексів масиву?
* Які є операції для обробки масивів?
* Які обмеження накладаються на розмір і кількість елементів масиву в мові Pascal?
* Як зберігається багатовимірний масив в пам’яті комп’ютера?
* Які масиви називають квадратними і які вони мають особливості?

***Практична частина***

1. Ввести одновимірний масив. На місці максимального елемента поставити число 1000. Масив вивести на екран.
2. Ввести одновимірний масив. На місці мінімального елемента поставити число 100. Масив вивести на екран.
3. Скласти програму пошуку заданого числа, кількості його входжень в масив та номер першого елемента.
4. Скласти програму вставки елемента до масиву. (Задано одновимірний масив, номер, після якого необхідно вставити елемент та сам елемент для вставки). Масив вивести на екран.
5. Скласти програму видалення елемента з масиву. (Задано одновимірний масив, номер, після якого необхідно видалити елемент). Масив вивести на екран.
6. В одновимірному масиві знайти мінімальний елемент та розділити кожен елемент масиву на індекс мінімального елемента. Масив вивести на екран.
7. В одновимірному масиві знайти максимальний елемент та розділити кожен елемент масиву на індекс максимального елемента. Масив вивести на екран.
8. В одновимірному масиві знайти максимальний елемент та розділити кожен елемент масиву на максимальний елемента. Масив вивести на екран.
9. В одновимірному масиві знайти мінімальний елемент та розділити кожен елемент масиву на мінімальний елемента. Масив вивести на екран.
10. Дано масив, що складається з 0 та 1. Підрахувати, чого в масиві більше 0 чи 1.
11. Дано два одинакові за довжиною одновимірні масиви. Злити їх у третій масив, чергуючи елементи першого та другого масиву.
12. Дано два одинакові за довжиною одновимірні масиви. Злити їх у третій масив, спочатку занести елементи першого масиву, а потім – другого.

**Практична робота № 5**

**«Створення програм з використанням двовимірних масивів»**

**Мета**:

1. Закріплення навичок роботи у середовищі програмування.
2. Розвиток інформаційної компетентності (навичок аналізу та локалізації помилок).
3. Формування компетентності саморозвитку та самоосвіти.

**Хід роботи:**

***Теоретична частина***

* ***Двовимірний масив – це масив , у якому зберігається таблиця значень (матриця), його елементи нумеруються двома індексами – номером рядка та номером стовпчика відповідно.***

***Опис двовимірного масиву в розділі Var :***

***Var<ім’я масиву>: array[нижня\_межа1..верхня\_межа1, нижня\_межа2..верхня\_межа2] of<тип елементів>;***

***Наприклад:***

***Var а:array[1..100,1..100] of integer;***

 ***b: array[1..10,1..5] of real;***

***або***

***Const n=30;m=10;Vart:array[1..n,1..m] of integer;***

* ***Задання масиву типізованою константою:***

***const x: array[1..2,1..3] of integer=(1,3,5),(6,2,4);***

* ***Попередній опис типу масиву:***

***type<ім’я типу>= array[нижня\_межа1..верхня\_межа1, нижня\_межа2..верхня\_межа2] of<базовий тип елементів>;***

***var<ім’я масиву>:< ім’я типу >;***

***Наприклад,***

***type t\_vector=array[1..100,1..100] of real; var mas:t\_vector;***

***Заповнення даними двовимірного масиву***

* ***З клавіатури:***

***- З коментарем***

 ***for i:=1 to n do***

 ***for j:=1 to m do***

 ***begin***

 ***writeln(‘введіть а[’,i,’,’,j,’]:’);***

 ***readln(a[i,j]);***

 ***end;***

***-Через пропуск пустого рядка між рядками матриці***

***for i:=1 to n do***

 ***begin***

 ***for j:=1 to m do***

 ***read(a[i,j]);***

 ***writeln; { пропуск рядка між рядками матриці }***

***end;***

***-Заповнення в стовпчик***

 ***for i:=1 to n do***

 ***for j:=1 to m do***

 ***readln(a[i,j]);***

* ***Випадковим чином:***

***randomize; {ініціалізація генератора випадкових чисел}***

***for i:=1 to n do***

 ***for j:=1 to m do***

 ***a[i,j]:=random(100);***

* ***Через оператор присвоєння:***

***Var p:array[1..3,1..2] of integer;***

***Begin***

 ***p[1,1]:=100; p[1,2]:=200; p[2,1]:=300; p[2,2]:=200; p[3,1]:=400; p[3,2]:=250;***

 ***……………..***

* ***З використанням формул:***

***for i:=1 to n do for j:=1 to m do a[i,j]:=100\* i-1/j;***

***Виведення масиву***

***В стандартній формі запису – по рядках та стовпцях:***

***For i:=1 to n do***

***begin***

 ***For j:=1 to m do***

 ***write(a[i,j]:4);***

 ***writeln;***

***end;***

***- Виведення в рядок через пропуск-розділювач:***

***For i:=1 to n do***

***begin***

 ***For j:=1 to m do***

 ***write(a[i,j],’ ’);***

 ***writeln;***

 ***end;***

***Актуалізація опорних знань:***

* Що таке масив? Як він оголошується?
* Яким може бути тип елементів масиву?
* Яким може бути тип індексів масиву?
* Які є операції для обробки масивів?
* Які обмеження накладаються на розмір і кількість елементів масиву в мові Pascal?
* Як зберігається багатовимірний масив в пам’яті комп’ютера?
* Які масиви називають квадратними і які вони мають особливості?

***Практична частина***

1. Замінити нулем елементи К-го стовпця та L-го рядка, залишивши елемент на їх перетині.
2. Замінити нулем діагональні елементи квадратної матриці, залишивши елемент на перетині діагоналей.
3. Замінити нулями рядки, що складаються тільки з від’ємних елементів.
4. Замінити нулями рядки, що містять хоча б один від’ємний елемент.
5. Замінити нулями ті рядки, у яких є нулі.
6. Замінити одиницями ті рядки, у яких є хоча б один від’ємний елемент.
7. Замінити нулями ті рядки та стовпчики, де є нульові елементи.
8. Замінити одиницею елементи К-го стовпця та L-го рядка, залишивши елемент на їх перетині.
9. Замінити одиницею діагональні елементи квадратної матриці, залишивши елемент на перетині діагоналей.
10. Замінити нулями рядки, що складаються тільки з додатніх елементів.
11. Замінити нулями рядки, що містять хоча б один додатній елемент.
12. Замінити нулями ті рядки, у яких є елементи більші 1000.
13. Замінити одиницями ті рядки, у яких є хоча б один елемент, більший з 1000.
14. Замінити нулями ті рядки та стовпчики, де є елементи, кратні 3.

**Практична робота № 6**

**«Створення програм з використанням підпрограм».**

**Мета**:

1. Закріплення навичок роботи у середовищі програмування.
2. Розвиток інформаційної компетентності (навичок аналізу та локалізації помилок).
3. Формування компетентності саморозвитку та самоосвіти.

**Хід роботи:**

***Теоретична частина***

***Підпрограма – логічно закінчена група операторів мови програмування, яку можна викликати для виконання по імені багаторазово з різних місць програми.***

***Розрізняють різні типи підпрограм:***

* ***Підпрограми-процедури***
* ***Підпрограми-функції***

***Підпрограма-процедура може мати будь-яку кількість вхідних і вихідних значень.***

***Підпрограма-функція має будь-яку кількість вхідних значень і видає тільки один результат-значення***

***Структура процедури***

***Procedure ім’я\_процедури(список форм. параметрів);***

***label***

***const Опис локальних міток, констант,***

***type типів та змінних***

***var***

***procedure Опис внутрішніх процедур та функцій***

***function***

***Begin***

 ***оператори ;***

***end.***

***У заголовку процедури вказується ім’я-ідентифікатор процедури та описується список формальних параметрів. Присутність формальних параметрів необов‘язкова (процедура без параметрів).***

***Опис локальних об’єктів (змінних, констант, внутрішніх процедур та функцій) та оператори складають внутрішню частину процедури. Команди, які повинні виконуватися під час запуску процедури, містяться в операторній частині модуля процедури.***

***Запуск процедури в основній програмі здійснюється за допомогою оператора процедури, який містить ім’я процедури та необхідні фактичні параметри. Виклик процедури є лінійним оператором. Наприклад,***

 ***ім’я\_процедури (список фактичних параметрів);***

***Структура функції***

***Function ім’я\_функції(список форм. параметрів): тип\_результату;***

***label***

***const Опис локальних міток, констант,***

***type типів та змінних***

***var***

***procedure Опис внутрішніх процедур та функцій***

***function***

***Begin***

 ***оператори ;***

***<ім’я\_функції>:=<вираз>;***

***end.***

***Формальні параметри – це параметри, що описуються в заголовку процедури чи функції. Фактичні параметри підставляються замість формальних під час виклику підпрограми. Кількість, тип і порядок формальних і фактичних параметрів повинні бути у строгій відповідності (співпадати).***

***Локальні дані можуть використовуватися й змінюватися тільки операторами даної програми.***

 ***Глобальні дані – це дані, оголошені в розділі описів головної програми . Вони доступні як основній програмі, так і всім процедурам і функціям, що в ній описані.***

***Актуалізація опорних знань:***

* Для чого призначені процедури?
* Що містить у собі заголовок процедури?
* Чим відрізняються формальні й фактичні параметри?
* Чим відрізняються локальні й глобальні змінні?
* Для чого призначені функції?
* Що містить у собі заголовок функції?
* Яка різниця між процедурою та функцією?

***Практична частина***

***Варіант 1***

**1.** Ввести прізвище, ім’я та по батькові та як одне дане типу рядок. Вивести ім’я та кількість букв у третьому слові (Використати процедуру).

**2.** Дано двовимірний масив. Елементи масиву, що дорівнюють нулю замінити на 1 (Використати процедуру).

 Знайти суму елементів під головною діагоналлю.(Використати функцію).

**3.** Дано сторони для N трикутників. Обчислити площу кожного трикутника, якщо трикутник існує (Використати процедуру).

***Варіант 2***

**1.** Ввести прізвище, ім’я та по батькові як одне дане типу рядок. Визначити, скільки букв „а” є у прізвищі (Використати функцію).

**2.** Дано двовимірний масив. Обчислити добуток мінімального і максимального елементів масиву (Використати функцію).

**3.** Дано сторони для N прямокутників. Обчислити площу та периметр кожного (Використати процедуру).

***Варіант 3***

**1.** Ввести прізвище, ім’я та по батькові як одне дане типу рядок. Вивести довжини прізвища та імені (Використати процедуру).

**2.** Дано двовимірний масив. Обчислити суму від’ємних елементів (Використати функцію). Знайти максимальний елемент та його індекс (Використати процедуру).

**3.** Дано радіуси для N кіл. Обчислити довжину та площу кожного (Використати процедуру).

***Варіант 4***

**1.** Ввести прізвище, ім’я та по батькові як одне дане типу рядок. Визначити, скільки букв „о” є в імені (Використати функцію).

**2.** Дано двовимірний масив. Обчислити суму елементів масиву над головною діагоналлю (Використати функцію).

Визначити індекси мінімального елемента (Використати процедуру).

**3.** Обчислити корені N лінійних рівнянь (Використати процедуру).

**Практична робота № 7**

**«Створення програм з використанням рядкових величин»**

**Мета**:

1. Закріплення навичок роботи у середовищі програмування.
2. Розвиток інформаційної компетентності (навичок аналізу та локалізації помилок).
3. Формування компетентності саморозвитку та самоосвіти.

**Хід роботи:**

***Теоретична частина***

***Загальний вигляд опису символьних змінних:***

***Var<ідентифікатор>: char;***

 ***Наприклад, VarХ: char;***

 ***Для символів означено всі порівняння, причому вони впорядковані так само, як їхні номери. Справджуються такі нерівності:***

 ***‘ ‘<‘\_’<‘0’<‘1’<…<‘9’<‘A’<‘B’<…’Z’<‘a’<’b’<’c’<’d’<…<’z’***

 ***Символьну змінну можна використовувати як параметр циклу.***

 ***наприклад: For s:=‘A’ to ‘H’ do …***

***Введення символьних даних:***

***Read(f); Readln(a,b,c);***

***Виведення символьних даних:***

***Write (‘a’); {Виведення літери ‘а’}Write (a); {Виведення значення змінної а}***

***Функції для обробки змінних типу char***

* ***Chr(x:byte):char; {повертає символ, відповідний коду ASCII числа х}***
* ***Ord(x:char):byte; {повертає код ASCII символу х}***
* ***Upcase(x: char):char; {перетворює символи із великих на маленькі}***
* ***Pred(x: char):char; {повертає символ, що передує символу х}***
* ***Succ(x: char):char; {повертає символ, наступний за х}***

***В Паскалі використовуються багатосимвольні текстові дані.***

***Рядок – це взята в апострофи скінченна послідовність будь-яких символів.***

***Наприклад, ‘ABC’, ‘12345’.***

***Порожній рядок , тобто послідовність символів довжини 0, позначається константою ’’ (два апострофи без проміжку).***

* ***Базовий рядковий статичний тип String.***
* ***Пам’ять під змінні цього типу виділяється на початку виконання програми або підпрограми, її розмір 256 байтів.***
* ***Змінна цього типу є масивом елементів типу charз індексами від 0 до 255.***
* ***Поточне рядкове значення, зображене змінною, може мати довжину від 0 до 255 і займати ділянку масиву, починаючи з індексу 1.***

***Оголошення рядкових величин***

***Var<ідентифікатор>:string;***

***Var<ідентифікатор>:string[n]; де n – кількість символів у рядку. По замовчуванні n - 255 символів***

***Функції для роботи з рядками***

* ***Length(a) – довжина рядка, кількість символів у рядку а;***
* ***Copy(a,p,n) – копіює n символів рядка а, починаючи із символу під номером p;***
* ***Concat(a,b,…) – з’єднує рядки в один рядок;***
* ***Pos(a,b) – визначає номер символу, з якого починається входження тексту а в текст b;***

***Процедури для роботи з рядками***

* ***Insert(a,b,p) – вставка підрядка а в рядок b, починаючи від позиції з номером p***
* ***Delete(b,p,m) – вилучення m символів з рядка b, починаючи з позиції під номером p***
* ***Str(n,b) - перетворює числове значення n в рядковий тип змінної b***
* ***Val(a,<змінна 1>,<змінна 2>) – засилає у числовузмінну1 числовий образ рядка а. Якщо це коректно, то змінна2 приймає значення 0, інакше – числове значення номера першого недопустимого символу заданого рядка***

***Актуалізація опорних знань:***

* Що таке рядок і якою структурою даних він зображується?
* Які операції застосовуються до рядків?
* Як порівнюються рядкові величини?
* Особливості введення й виведення рядкових величин.
* Які процедури та функції можна застосовувати до рядкових величин?
* Які операції можна виконувати з кожним символом рядкової величини?

***Практична частина***

1. Перевірити, чи входить у заданий рядок символ «\*». Якщо входить, то надрукувати символ, що знаходиться безпосередньо перед ним, але тільки тоді, коли цей символ є цифра.
2. Перевірити, чи входить у заданий рядок символ «\*». Якщо входить, то надрукувати символ, що знаходиться безпосередньо перед ним, але тільки тоді, коли цей символ є цифра і піднісши його до квадрата.
3. У даній формулі підрахувати кількість входжень символів «+» та «-».
4. Обчислити загальну кількість символів «\*»,«+» та «-» у заданому тексті.
5. Дано текст. Визначити, які символи зустрічаються частіше «а» чи «о». Якщо якийсь із символів відсутній, повідомити про це.
6. У заданому тексті замінити всі символи «:» на «-» і навпаки.
7. У заданому тексті замінити всі символи «.» на «…». Якщо в тексті зустрічається підряд три крапки, то залишити їх без змін.
8. У даному тексті замінити всі послідовності крапок на одну крапку.
9. Дано деякий текст, в якому є хоча б одна кома. Визначити порядковий номер першої коми в тексті та останньої коми.
10. Дано текст-формулу, яка містить лише арифметичні операції +,-,\*,/ і не містить дужок та функцій. Визначити загальну кількість арифметичних дій, передбачених у цій формулі, та кількість чисел , над якими вони здійснюються.
11. Дано деякий текст. Відредагувати текст таким чином, щоб після кожного розділового знака стояв хоча б один пробіл.
12. Скласти програму, яка після кожної цифри в тексті вставляє в дужках її текстовий еквівалент. Наприклад, 0(нуль), 1(один),…
13. Дано текст-формулу. Визначити коректність формули щодо кількості відкритих і закритих дужок. Вважається, що закриті дужки не стоять перед відкритими. Якщо дужок немає – повідомити про це.

**Практична робота № 8**

**«Створення програм з використанням множин»**

**Мета**:

1. Закріплення навичок роботи у середовищі програмування.
2. Розвиток інформаційної компетентності (навичок аналізу та локалізації помилок).
3. Формування компетентності саморозвитку та самоосвіти.

**Хід роботи:**

***Теоретична частина***

***Множини – це структурований тип даних, який являє собою набір взаємопов’язаних за якою-небудь ознакою або групою ознак об’єктів, котрі можна розглядати як одне ціле. Тобто множина в Паскалі – це обмежений, невпорядкований набір різних елементів одного типу.***

***Кожен об’єкт в множині називається елементом множини.***

***Всі елементи в множині повинні належати одному із скалярних типів, крім дійсного; цей тип називається базовим типом. Базовий тип задається діапазоном або переліком.***

***Під потужністю множини розуміється число елементів, яке міститься в даній множині. Потужність пустої множини дорівнює нулю.***

***Всій множині дається ім’я, яке утворюється так, як імена простих змінних і масивів.***

***На відміну від масивів до елементів множини немає прямого доступу (по індексах цих елементів). Тому введення-виведення множин виконується з використанням спеціальних операцій.***

***В Паскаль-програмі множина описується у вигляді списку значень в квадратних дужках. Наприклад, [1,2,5]; [‘a’,’b’,’c’].***

***Кількість елементів у множині не може перевищувати 256 і множина не повинна мати елементи з порядковими номерами менше 0 і більше 255.***

***Один елемент множини займає один біт пам’яті, максимальний розмір множини - 32 байта.***

* ***Якщо в множині елемент повторюється, то вважаємо, що він туди входить один раз. Наприклад, множини [2,5,2] і [2,5] еквівалентні.***
* ***Порядок елементів у множині не має значення. Наприклад, множини [5,2] і [2,5] еквівалентні.***
* ***Пуста множина позначається [].***
* ***Для опису змінних типу множина використовують службове слово set of (множина із...).***

***Формат:***

***Type <Ім’я типу>=set of < базовий тип>;***

***Var <ідентифікатор>:<ім’я типу>;***

***Type***

 ***Prost=set of (3,5,7,11,14);***

 ***Nomer=set of 1..31;***

***Var***

 ***Pr:Prost;***

***N: Nomer;***

***В даному прикладі змінна Pr може приймати значення 3,5,7,11,14; змінна N – будь-яке значення з діапазону 1..31***

* ***Можна задавати тип множини і без попереднього опису:***

***Var <ідентифікатор>:set of < базовий тип>;***

 ***Var***

***Bukva: set of (‘a’,’b’,’c’);***

***Операції над множинами***

* ***ОпераціяIN. Використовується для перевірки належності якого-небудь елемента вказаній множині.***
* ***Об’єднанням двох множин (+) є третя множина, яка містить елементи обох множин.***
* ***Перетином двох множин (\*) є третя множина, яка містить елементи, що входять одночасно в обидві множини.***
* ***Різницею двох множин (-) є третя множина, яка містить елементи першої множини, що не входять до другої множини.***

***Актуалізація опорних знань:***

* Що таке множина?
* Як множини зображаються на мові Паскаль?
* Які вирази позначають тип множини?
* Які типи можуть бути базовими для типів множин?
* Які операції застосовані до множин у мові Паскаль та який їх пріоритет?
* Як порівнюються множини в операціях порівняння?

***Практична частина***

1. Прочитати від клавіатури два слова, складені малими латинськими літерами, та визначити, чи можна утворити друге з них, використовуючи лише букви першого слова. Кількість повторень букв у словах не має значення. Наприклад, з букв слова bao можна утворити слова boa та baobab, а слово bay — ні.
2. Скласти програму, що виконує і виводить на екран дисплея набори випадкових чисел для гри в "Спортлото 5 з 36".
Для заповнення кожної картки спортлото необхідно одержати набір з п'яти випадкових чисел. До цих чисел пред'являються дві вимоги:
–    числа повинні знаходитися в діапазоні 1..36;
–    числа не повинні повторюватися.
3. З рядка тексту відібрати голосні літери.
4. З масиву цілих чисел відібрати прості сформувати з них множину.
5. З масиву дійсних чисел відібрати цілі сформувати з них множину.
6. Ввести рядок символів, який закінчується символом ’**.**’. Визначити чи є у введеному рядку цифри.
7. Задати три довільні множини одного базового типу. Сформувати нову множину. X=(Y1-Y2)+(Y2\*Y3). Перевірити чи включено множину Y3 в X . Підрахувати скільки разів в множинах зустрічаються літери від А до К.
8. Дано рядок з рядкових латинських букв, за яким йде точка. Визначити, яких букв - голосних (a, e, i, o, u) або приголосних - більше в цьому тексті.
9. Дано рядок з рядкових латинських букв, за яким йде точка. Визначити, яких букв - голосних або приголосних - більше в цьому тексті.
10. Дано рядок символів. Підрахувати скільки в цьому рядку цифр.
11. Дано рядок символів. Підрахувати скільки в цьому рядку розділових знаків.
12. Дано рядок слів. Почати кожне слово в рядку з великої букви (враховувати наявність в рядку розділових знаків).
13. Дано рядок символів. Підрахувати кількість символів латинського алфавіту і кількість цифр.
14. Дано рядок. Перетворити в ньому усі рядкові букви в прописні, а прописні – в рядкові

**Практична робота № 9**

**«Створення програм з використанням записів»**

**Мета**:

1. Закріплення навичок роботи у середовищі програмування.
2. Розвиток інформаційної компетентності (навичок аналізу та локалізації помилок).
3. Формування компетентності саморозвитку та самоосвіти.

**Хід роботи:**

***Теоретична частина***

***Запис – це структурований тип даних, що складається з фіксованої кількості компонентів одного або декількох типів, що описуються і опрацьовуються як одне ціле Компоненти запису називаються полями. Різні поля можуть мати різний тип. Поля можуть бути як простими даними, так і складеними (масивами або записами).***

***Опис типу запис починається словом RECORD і закінчується словом END. Між ними розміщується список компонентів, які називаються полями з указанням ідентифікаторів полів та типу кожного поля.***

***Формат:***

***Type <назва типу> = record***

***<ідентифікатор поля1>:<тип компонента>;***

***<ідентифікатор поля2>:<тип компонента>;***

***...***

***<ідентифікатор поля N>:<тип компонента>;***

***End;***

***var <список змінних>:<назва типу>;***

***Наприклад:***

***Type mash = record***

 ***Nomer: integer;***

 ***Marka: string[20];***

 ***Fl: string[40];***

 ***Adress: string[60]***

***End;***

***Var k,car:mash;***

***Доступ до полів запису здійснюється за допомогою складних назв, що складаються з ідентифікатора змінної та ідентифікатора поля, розділених крапкою.***

***Наприклад, car.Fl, car.marka i т.д.***

***Значення полів запису можна використовувати у виразах та операторах.***

***Часто в задачах зручно використовувати масиви записів. Їх можна описати таким чином:***

***Type***

 ***person=record***

 ***fio:string[20];***

 ***age:1..99;***

 ***prof:string[30]***

 ***end;***

***Var list:array[1..50] of person;***

***Для спрощення звертання до полів запису за допомогою складних імен Паскаль має оператор WITH такого формату:***

***With<список змінних типу запис>do<оператор>;***

***Так оператор присвоєння з попереднього прикладу можна замінити на:***

***With car do begin***

***Marka:=’zaporojhec’;nomer:=234522;***

***End;***

***Паскаль допускає вкладення запису один в один (тобто поле запису може бути записом).***

***Відповідно операції With теж можуть бути вкладеними.***

***Актуалізація опорних знань:***

* У яких ситуаціях зручніше використовувати тип record?
* Чи можуть бути компоненти записів різних типів?
* Як називаються елементи, з яких складаються записи?
* Яким чином можна звертатися до значень полів змінної типу «запис»?
* У чому полягає зручність використання оператора приєднання?

***Практична частина***

1. Ввести інформацію по 10-ти студентах ( порядковий номер, ПІБ і три оцінки) визначити кількість відмінників і середній бал кожного студента.
2. Ввести інформацію 10-ти виборцях, визначити кількість людей, що проживають на вулиці Польова.
3. Ввести інформацію 1-ти спортсменах (ПІБ, зріст, вага), визначити кількість людей, чия вага перевищує 70 кг.
4. Ввести інформацію про товар (найменування, ціна, кількість, поле вартість розрахувати як ціна\* кількість). Вивести підсумкову вартість кожного товару.
5. Ввести інформацію про рейс літака (N рейсу, час вильоту, час прильоту, пункт призначення). Вивести інформацію про всі рейси в задане місто.
6. Скласти програму, що виводить на екран меню дитячого кафе (найменування виробу, вага, вартість).
7. Скласти програму, що виводить на екран розклад руху поїздів (N поїзда, станція відправлення, станція прибуття, час прибуття, час в дорозі).
8. Скласти програму, що виводить на екран список книг домашньої бібліотеки (автор, назва книги, видавництво, рік видання, вартість). Провести пошук по авторові книги.
9. Скласти програму, що виводить на екран розклад іспитів і заліків (предмет, вигляд звітності, число, викладач).
10. Скласти програму, що виводить на екран зведення про періодичні видання (найменування видання, наклад, річна вартість).
11. Скласти програму, що виводить на екран розклад вчителя (номер уроку, час початку уроку, клас, предмет, номер кабінету).
12. Скласти програму, що виводить на екран перелік товарів, наявних в продажі в магазині "Океан" (найменування, одиниця виміру, ціна, кількість).
13. Скласти програму, що виводить на екран "Телефонний довідник" (ПІБ., адреса, номер телефону). Виробити пошук по ПІБ– видати адресу і телефон.
14. Скласти програму, що виводить на екран графік відпусток (ПІБ дата початку відпустки, дата виходу на роботу, кількість днів).
15. Скласти програму, що виводить на екран інвентаризаційну відомість (інвентаризаційний номер, найменування, дата обліку) .
16. Скласти програму, що виводить на екран відомість нарахованої заробітної плати (ПІБ., посада, дата народження, заробітна плата).
17. Скласти програму, що виводить на екран зміни температури по днях тижня (дата, день тижня, температура, вологість).
18. Ввести інформацію 10-ти учасників змагань (ПІБ, рік народження, результат). Визначити переможця змагань.
19. Ввести інформацію по 10-ти перевезеннях (N рейсу, пункт призначення, найменування, вага). Знайти сумарний об'єм всіх перевезень.
20. Ввести інформацію по 10-ти школярах (ПІБ, стать, рік народження). Визначити кількість хлопчиків і дівчаток. Вивести список кожних.

**Практична робота № 10**

**«Створення програм з використанням файлів»**

**Мета**:

1. Закріплення навичок роботи у середовищі програмування.
2. Розвиток інформаційної компетентності (навичок аналізу та локалізації помилок).
3. Формування компетентності саморозвитку та самоосвіти.

**Хід роботи:**

***Теоретична частина***

***Файл – це сукупність даних будь-якого типу. Файл зберігається на диску, кількість даних при його описуванні не вказується, елементи файлу не мають індексів.***

***В мові Паскаль є три види файлів:***

* ***типізовані,***
* ***текстові,***
* ***нетипізовані***

***Опис файлової змінної (або в розділі типів або в розділі змінних):***

***Type<ім’я типу>=file of<базовий тип>;***

***Var<ім’я файлової змінної>:<ім’я типу>;***

***або***

***Var<ім’я файлової змінної>: file of<базовий тип>;***

***Так як число елементів файлу не задається, то в Паскалі введена стандартна функція***

***EOF(var F:file):boolean;***

***Коли буде досягнуто кінця файлу, EOF(F) матиме значення true, в протилежному випадку – false.***

***Так як кількість записів у файлі невідома, то при читанні даних використовується цикл “доки”:***

***WHILE NOT EOF(F) DO – поки не зустрілась ознака кінця файлу, виконувати цикл.***

***Стандартні процедури для роботи з файлами***

***ASSIGN (var F : file, Name : string);***

***Зв’язування імені файлової змінної F в програмі з іменем зовнішнього файлу на диску Name.***

***RESET (var F : file); Підготовка до читання файлу F***

***REWRITE (var F: file); Підготовка до запису на початок файлу F***

* ***READ (ім’я файлової змінної, ім’я області пам’яті); - Забезпечує передачу даних***

 ***з файлу в оперативну пам’ять.***

***WRITE (ім’я файлової змінної, ім’я області пам’яті); - Забезпечує передачу даних з оперативної пам’яті у файл.***

* ***WRITE (var F : file ; X1, X2, … , XN);***
* ***CLOSE (var F : file); Закриття відкритого файлу***
* ***SEEK (var F: file, N : Longint); Дає змогу здійснювати прямий доступ до елементів файлу F***
* ***FILEPOS (var F : file) : Longint; Дає змогу визначити поточне положення вказівника (номер елемента).***
* ***FILESIZE (var F : file) : Longint; Дає розмір файлу, тобто число елементів файлу F.***

***Опис текстового файла має вигляд:***

* ***Var < ім’я файлу> : text;***
* ***Для визначення кінця рядка використовується функція : EOLN (var F : text) : Boolean;***

***Для роботи з текстовими файлами поряд з розглянутими вище (крім SEEK, FILEPOSE, FILESIZE) використовуються такі стандартні процедури:***

* ***APPEND (var F : file); - відкриття уже існуючого текстового файлу F для доповнення даних в кінець файлу.***
* ***WRITE (var F : file; Х1, Х2, ..., ХN); - запис у файл F.***
* ***READ (var F : file; Х1, Х2, ..., ХN); - читання з файлу F.***
* ***WRITELN (var F : file; Х1, Х2, ..., ХN); - запис рядка в текстовий файл F значень змінних Х1, Х2, ..., ХN.***
* ***READLN (var F : file; Х1, Х2, ..., ХN); - читання текстового рядка файлу F.***

***Актуалізація опорних знань:***

* Що таке файл і що таке файлова змінна?
* Які методи доступу до файлів ви знаєте? Чим вони відрізняються?
* Що таке файловий вказівник і для чого він використовується?
* Які різновиди типів файлів є в мові Паскаль і чим вони відрізняються?
* Як можна відкрити текстовий файл?
* Які наслідки можливі, якщо не закрити файл, відкритий для запису?
* Чим відрізняються процедури **read** та **readln**?

***Практична частина***

1. Скопіювати з одного файлу в інший тільки певні символи (наприклад, ряд голосних) і порахувати їх загальну кількість.

2. Дано рядок символів завдовжки n. Необхідно замінити в ньому групу букв *foot* групою букв *feet.*

3. Дано текстовий рядок, що складається із заголовних латинських букв. Необхідно знайти підрядок з трьох букв, який зустрічається в цьому рядку найчастіше. Наприклад, в рядку DEFDEFABCABCZABCDEFDEF найчастіше 4 рази) зустрічається підрядок DEF.

4. Створити файл X, компонентами якого являються елементу масиву M. Переписати з файлу X у файл Y усі компоненти, які стоять на парних місцях. Прочитати файл Y. M = (0,1; 5,6; 0.8; 2,4; 7,3; 6,8; 0,5; 2,3;)

5.Даний файл з учбовим розкладом. Написати програму,яка визначає

кількість лекційних, практичних і лабораторних занять. Вивести їх кількість на екран.

Примітка:*в кожному рядку файлу може зустрічатися тільки одно з трьох  слів ("лекц"., "практ". і "лаб".), то якщо знаходиться перше слово, немає  потреби шукати інші.*

6. Записати в текстовий файл усі числа з іншого текстового файлу, більших за їх середнє арифметичне.

7.Дано текстовий файл. Необхідно переписати його вміст в іншій
текстовий файл, додаючи в початок кожного рядка його порядковий номер
(десятковий запис цілого числа шириною в 5 символів, вирівняний
по правій межі поля) і пропуск.

8.Дано текстовий файл. Необхідно переписати в інший текстовий файл
усі його непорожні рядки, які починаються і закінчуються
однаковим символом.

9.Дано текстовий файл. Необхідно переписати в інший текстовий файл
його вміст, окрім непорожніх рядків, що розпочинаються з точки.

10.Дано текстовий файл. Знайти кількість порожніх рядків в ньому.
11. Дано текстовий файл. Знайти довжину його максимального рядка.
12. Дано текстовий файл. Знайти кількість рядків в ньому, що починаються із
заданого символу.
13. Дано текстовий файл. Знайти кількість рядків в ньому, що закінчуються
заданим символом.

14.Дано текстовий файл. Знайти кількість рядків в ньому, що складаються з
однакових символів.

15. Дано текстовий файл. Видалити в ньому усі рядки максимальної довжини.

17. Дано текстовий файл. Переписати його вміст в інший текстовий
файл, доповнюючи справа усі його рядки, менші по довжині 80 символів
пропусками справа до 80 символів.

18. Дана послідовність символів, що закінчується точкою і вводиться
користувачем з клавіатури. Необхідно записати її в текстовий файл,
вставляючи в неї символи перекладу рядка через кожні 40 символів.

19. Дані два текстові файли. Перевірити їх вміст нарівність.

20. Дано текстовий файл, що складається з однієї абодекількох рядків.

Кожен рядок файлу містить числа, розділені пропусками.Замінити  пропуски, що йдуть підряд, одним пробілом

21.Дано текстовий файл, що складається з одного або декількох рядків. Кожен
рядок файлу містить числа, розділені пропусками. Перевірити
чи міститься у файлі задане число. Знайти суму і кількість
парних чисел файлу. Результати з відповідними коментарями
дописати в початковий файл.

# Додаток В

**Задачі з розв’язками**

(Програми на Pascal створені в середовищі FreePascal.

 Програми на С++ створені в середовищі CodeBlocks.)

**Цілочисельна арифметика. Проста математика. Довга арифметика.**

**Зарплата**

*(Час 1 сек. Пам’ять16 Мб. Складність 5%)*

У відділі працюють 3 співробітники, які отримують заробітну плату в гривнях. Визначити, на скільки зарплата найбільш високооплачуваного з них відрізняється від найбільшнизькооплачуваного.

**Вхідні дані**

У єдиному рядку вхідного файлу **INPUT.TXT** записані розміри зарплат усіх співробітників через пропуск. Кожна заробітна плата – це натуральне число, що не перевищує **105**.

**Вихідні дані**

У вихідний файл **OUTPUT.TXT** необхідно вивести одно ціле число –різницю між максимальною і мінімальною зарплатою.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 100 500 1000 | 900 |
| 2 | 3600 1100 2000 | 2500 |

**Код програми на С++**

#include <iostream>

#include <fstream>

using namespace std;

int main ()

{

ifstream fin("input.txt");

ofstream fout("output.txt");

int a,b,c,max,min;

 fin >> a>>b>>c;

min=a; max=a;

if (b>max) max=b;

if (c>max) max=c;

if (b<min) min=b;

if (c<min) min=c;

fout <<(max-min);

 return 0;

}

**Код програми на Pascal**

var a,b,c, min,max:integer;

begin

assign(input, 'input.txt'); reset(input);

assign(output, 'output.txt'); rewrite(output);

read (a,b,c);

min:=a; max:=a;

if b>max then max:=b;

if c>max then max:=c;

if b<min then min:=b;

if c<min then min:=c;

write (max-min);

end.

**Розворот**

*(Час 1 сек. Пам’ять16 Мб. Складність 9%)*

Дано натуральне число **N** та послідовність із **N** елементів. Треба вивести цю послідовність у зворотному порядку.

**Вхідні дані**

В першому рядку вхідного файлу **INTPUT.TXT** записано натуральне число**N (N<=103** ). У другому рядку вхідного файлу через пробіл записані **N**цілих чисел, які по модулю не перевищують **103**  - елементи послідовності.

**Вихідні дані**

В вихідний файл **OUTPUT.TXT** вивести задану послідовність у зворотному порядку.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 31 2 3 | 3 2 1 |

**Код програми на С++**

#include <iostream>

#include <fstream>

using namespace std;

int main ()

{

ifstream fin("input.txt");

ofstream fout("output.txt");

int a[1001], n, i;

 fin >> n;

 for (i=1;i<=n;i++) {fin >> a[i];}

 for (i=n;i>0;i--) { fout << a[i] << " "; }

fin.close();

fout.close();

return 0;

}

**Код програми на Pascal**

program razvorot;

var n,i:integer;

a:array[1..100] of integer;

input,output:text;

begin

assign(input, 'input.txt'); reset(input);

assign(output, 'output.txt'); rewrite(output);

readln (input,n);

for i:=1 to n do

 read(input,a[i]);

 for i:=n downto 1 do

 write(output,a[i],' ');

close (input);

close (output);

end.

**Чотирьохзначний паліндром**

*(Час 1 сек. Пам’ять16 Мб. Складність 10%)*

Необхідно написати програму, яка визначає, чи є чотирьохзначне натуральне число **N** паліндромом, тобто числом, яке одинаково читається зліва направо і справа наліво.

**Вхідні дані**

Вхідний файл **INTPUT.TXT** містить натуральне число **N (1000<=N<=9999)**.

**Вихідні дані**

В вихідний файл **OUTPUT.TXT** треба вивести слово “**YES**”, якщо число **N** є паліндромом, або “**NO**” – якщо ні.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 6116 | YES |
| 2 | 1231 | NO |

**Код програми на С++**

#include <ostream>

#include <fstream>

using namespace std;

int main(){

 ifstream fcin("input.txt");

 ofstream fcout("output.txt");

 int n,a1,a2,tmp;

 fcin >>n;

 a1=(n%10)\*10+((n/10)%10);

 tmp=(n/100);

 a2=(tmp%10)+((tmp/10)%10)\*10;

 if (a1==a2) fcout <<"YES";

else fcout <<"NO";

 fcin.close();

 fcout.close();

return 0; }

**Код програми на Pascal**

var f1,f2:text;

x:integer;

x1,x2,x3,x4:real;

begin

assign(f1,'input.txt');

assign(f2,'output.txt');

reset(f1);

rewrite(f2);

read(f1,x);

 x1 := x div 1000;

 x2 := x div 100 mod 10;

 x3 := x div 10 mod 10;

 x4 := x mod 10;

 if(x1=x4) and (x2=x3)then write(f2,'YES') else write(f2,'NO');

close(f1);

close(f2);

end.

**День програміста**

*(Час 1 сек. Пам’ять16 Мб. Складність 13%)*

День програміста відзначається в 255-й день року (при цьому 1 січня вважається нульовим днем). Написати програму, яка визначить дату (місяць і число григоріанського календаря), на яку випадає День програміста в заданому році. У григоріанському календарі високосним є:

- рік, номер якого ділиться без остачі на 400;

- рік, номер якого ділиться на 4, але не ділиться на 100.

**Вхідні дані**

У єдиному рядку вхідного файлу **INPUT.TXT** записане ціле число від 1 до 9999 включно, яке означає номер року нашої ери.

**Вихідні дані**

У єдиний рядок вихідного файлу **OUTPUT.TXT** треба вивести дату Дня програміста у форматі **DD/MM/YYYY**, де **DD** - число, **MM** - номер місяця (01 - січень, 02 – лютий, ..., 12 - грудень), **YYYY** - рік в десятковому записі.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 2000 | 12/09/2000 |
| 2 | 2009 | 13/09/2009 |
| 3 | 2017 | 13/09/2017 |
| 4 | 2018 | 13/09/2018 |

**Код програми на С++**

#include<iostream>

#include<fstream>

#include <string>

using namespace std;

int main(){

int n;

string s;

ifstream fcin("input.txt");

ofstream fcout("output.txt");

fcin>>n;

fcin.close();

fcin.open("input.txt");

fcin >>s;

 while (s.length()<4) s="0"+s;

 if (((n % 4==0)&&(n % 100!=0))||(n % 400==0)) fcout<<"12/09/"<<s; else fcout<<"13/09/"<<s;

return 0;}

**Код програми на Pascal**

var n:longint;

 s:string;

begin

 assign(input,'INPUT.TXT');

 reset(input);

 assign(output,'OUTPUT.TXT');

 rewrite(output);

read(n);

 str(n,s);

 while length(s)<4 do s:='0'+s;

 if ((n mod 4=0)and(n mod 100<>0))or(n mod 400=0) then write('12/09/',s) else write('13/09/',s);

end.

**Кавуни**

*(Час 0,5 сек. Пам’ять16 Мб. Складність 14%)*

Дід Панас прийшов на ринок і вирішив купити два кавуни: один для себе, а другий для сусіда. Собі треба вибрати кавун важкий, а для сусіда легший. Але кавунів дуже багато і серед них треба вибрати найважчий та найлегший. Допоможемо йому.

**Вхідні дані**

В першому рядку вхідного файлу **INTPUT.TXT** записано натуральне число**N –** кількість кавунів. У другому рядку вхідного файлу через пробіл записані **N**цілих чисел, де кожне число – це маса відповідного кавуна. Всі числа натуральні і не перевищують 30000.

**Вихідні дані**

В вихідний файл **OUTPUT.TXT** треба вивести два числа через пробіл: масу кавуна, який дід Панас купить сусіду і масу кавуна, який він купить собі.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 5 | 1 9 |
| 2 | 5 1 6 5 9 |  |

**Код програми на С++**

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream fcin("input.txt");

ofstream fcout("output.txt");

int i,n,x,min=30000,max=0;

 fcin >>n;

for (i=1;i<=n;i++) {fcin >>x;

 if (x>max){max=x;};

 if (x<min){min=x;}}

 fcout <<min<<" "<<max;

fcin.close();

fcout.close();

return 0;

}

**Код програми на Pascal**

var i,j,min,max,tmp: integer;

 t: text;

begin

 assign(t,'input.txt');

 reset(t);

 readln(t,i);

 min:=30000;

 for j:=1 to i do begin

 read(t,tmp);

 if min>tmp then min:=tmp;

 if max<tmp then max:=tmp;

 end;

 close(t);

 assign(t,'output.txt');

 rewrite(t);

 writeln(t,min,' ',max);

 close(t);

end.

**Статистика**

*(Час 1 сек. Пам’ять16 Мб. Складність 15%)*

Вася не любить англійську мову, але кожного разу намагається отримати хоча б четвірку за семестр. У поточному семестрі Вася помітив наступну закономірність: по непарних днях місяця він отримував трійки, а по парних–четвірки. Так само він пам'ятає, в які дні він отримував ці оцінки. Тому він виписав на папірці усі ці дні для того, щоб оцінити, скільки у нього трійок і скільки четвірок. Допоможіть Васе це зробити, розташувавши парні і непарні числа в різних рядках. Вася може розраховувати на оцінку 4, якщо четвірок не менше, ніж трійок.

**Вхідні дані**

У першому рядку вхідного файлу **INPUT.TXT** записане число**N**–кількість елементів цілочисельного масиву (**1 ≤ N ≤ 100**). Другий рядок містить **N** чисел, що представляють заданий масив. Кожен елемент масиву – натуральне число від **1** до **31**. Усі елементи масиву розділені пропуском.

**Вихідні дані**

У перший рядок вихідного файлу **OUTPUT.TXT** треба вивести числа, які відповідають дням місяців, в які Вася отримав трійки, а в другому рядку відповідно розташувати числа місяця, в які Вася отримав четвірки. У третьому рядку треба вивести "**YES**", якщо Вася може розраховувати на четвірку і "**NO**" інакше. У кожному рядку числа слід виводити в тому ж порядку, в якому вони йдуть у вхідних даних. При виведенні, числа відділяються пропуском.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 54 16 19 31 2 | 19 314 16 2YES |
| 2 | 829 4 7 12 15 17 24 1 | 29 7 15 17 14 12 24NO |

**Код програми на С++**

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream fcin("input.txt");

ofstream fcout("output.txt");

 int n,i,tmp,c,nc;

fcin >>n;

c=0; nc=0;

for (i=1; i<=n; i++) {

 fcin >>tmp;

 if (tmp % 2==1) { nc=nc+1; fcout <<tmp<<' '; }}

fcin.close();

fcin.open("input.txt");

fcin >>n;

fcout <<'\n';

for (i=1; i<=n; i++) {

 fcin >>tmp;

 if (tmp % 2==0) { c=c+1; fcout <<tmp<<' ';}}

if (c>=nc) fcout <<'\n'<<"YES"; else fcout<<'\n'<<"NO";

 fcin.close();

 fcout.close();

return 0;

}

**Код програми на Pascal**

varinp,o: text;

n,i,tmp,c,nc: integer;

begin

 assign(inp,'input.txt');

 assign(o,'output.txt');

 reset(inp);

 rewrite(o);

 readln(inp,n);

 c:=0; nc:=0;

 for i:=1 to n do begin

 read(inp,tmp);

 if tmp mod 2=1 then begin nc:=nc+1; write(o,tmp,' '); end;

 end;

 writeln(o);

 close(inp);

 reset(inp);

 readln(inp,n);

 for i:=1 to n do begin

 read(inp,tmp);

 if tmp mod 2=0 then begin c:=c+1; write(o,tmp,' '); end;

 end;

 if c>=nc then writeln(o,#13#10+'YES')

else writeln(o,#13#10+'NO');

 close(inp);

 close(o);

end.

**Перепис**

*(Час 1 сек. Пам’ять16 Мб. Складність 15%)*

В будинку проживає N мешканців. Одного разу вирішили переписати всіх жильців і склали список, в якому вказали вік та стать кожного мешканця. Потрібно знайти номер найстаршого жильця чоловічої статі.

**Вхідні дані**

Вхідний файл **INTPUT.TXT** в першому рядку містить натуральне число**N –** кількість мешканців(**N<=100**). В наступних **N** рядках знаходиться інформація про всіх мешканців: кожен рядок містить два цілих числа: **V** і **S** – вік та стать людини (**1<=V<=100, S – 0** або **1**). Чоловічій статі відповідає значення **S**=1, а жіночій – **S**=0.

**Вихідні дані**

Вихідний файл **OUTPUT.TXT** повинен містити номер найстаршого чоловіка у списку. Якщо таких жильців декілька, то треба вивести найменший номер. Якщо жильців чоловічої статі немає, то вивести 1.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 4 | 2 |
|  | 25 1 |  |
|  | 70 1 |  |
|  | 100 0 |  |
|  | 3 1 |  |

**Код програми на С++**

#include <iostream>

#include<fstream>

#include<cstdio>

using namespace std;

int main()

{

freopen("INPUT.TXT","r",stdin);

freopen("OUTPUT.TXT","w",stdout);

int a,b,c=-1,i,n,k=0,m=-1;

cin>>n;

 for(i=0;i<n;i++)

 {

 cin>>a>>b;

 k++;

if(b==1 && c<a) { c=a; m=k; }}

cout<<m<<endl;

return 0;

}

**Код програми на Pascal**

var i,j,a,b,max,res: integer;

 f: text;

begin

 assign(f,'input.txt');

 reset(f);

 readln(f,i);

 res:=101;

 for j:=1 to i do begin

 readln(f,a,b);

 if (b=1)and(max<a) then max:=a;

 end;

 close(f);

 if max<>0 then begin

 reset(f);

 readln(f,i);

 for j:= 1 to i do begin

 readln(f,a,b);

 if b=1 then

 if a=max then

 if res>j then res:=j;

 end;

 close(f);

 end;

 assign(f,'output.txt');

 rewrite(f);

 if res=101 then writeln(f,'-1') else writeln(f,res);

 close(f);

end.

**Нулі**

*(Час 1 сек. Пам’ять 16 Мб. Складність 16%)*

Необхідно знайти найдовший безперервний ланцюжок нулів в послідовності нулів і одиниць.

**Вхідні дані**

У єдиному рядку вхідного файлу **INPUT.TXT** записана послідовність нулів і одиниць (без пропусків). Сумарна кількість цифр від **1** до **100.**

**Вихідні дані**

У єдиний рядок вихідного файлу **OUTPUT.TXT** треба вивести шукану довжину ланцюжка нулів.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 00101110000110 | 4 |

**Код програми на С++**

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream fcin("input.txt");

ofstream fcout("output.txt");

 char s;

 int c,max;

c=max=0;

 while(!fcin.eof()){

 fcin.get(s);

 if(s=='0') c=c+1; else c=0;

 if(c>max) max=c;

 }

 fcout<<max;

 fcin.close();

 fcout.close();

return 0;

}

**Код програми на Pascal**

var a,s:string;

 p:integer;

begin

assign(input,'input.txt');

assign(output,'output.txt');

reset(input);

rewrite(output);

readln(s);

a:='0';

p:=-1;

while p<>0 do

begin

p:=pos(a,s);

a:=a+'0';

end;

writeln(length(a)-2);

end.

**За один крок до щастя**

*(Час 1 сек. Пам’ять 16 Мб. Складність 16%)*

Вова купив квиток у трамваї і відразу порахував суми перших трьох цифр і останніх трьох цифр номера квитка (номер у квитка шестизначний). Виявилось, що суми відрізняються рівно на одиницю. "Я в одному кроці від щастя", – подумав Вова, – "або попередній або наступний квиток точно щасливий". Чи правий він?

**Вхідні дані**

Вхідний файл **INPUT.TXT** містить в першому рядку число **K**–кількість тестів. У наступних **K** рядках записані номери квитків. Кількість тестів не більше **10**. Номер складається рівно з шести цифр, серед яких можуть бути і нулі. Гарантується, що Вова уміє рахувати, тобто суми перших трьох цифр і останніх трьох цифр відрізняються рівно на одиницю.

**Вихідні дані**

Вихідний файл **OUTPUT.TXT** повинен містити **K** рядків, в кожній з яких для відповідного тесту слід вказати "**Yes**", якщо Вова правий, і "**No**", якщо ні.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 3715068445219012200 | YesNoYes |

**Код програми на С++**

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream fcin("input.txt");

ofstream fcout("output.txt");

 long a,x,y,x1,y1,k,c,v,a1,a2,x2,x3,y3,y2,i;

 fcin >>k;

 string s;

 for (i=1; i<=k; i++)

 {

 fcin>>a;

 a1=a+1;

 a2=a-1;

 x=a1 % 1000;

 y=a1 / 1000;

 x3=x % 10+(x / 10) % 10 +x / 100;

 y3=y % 10+(y / 10) % 10 +y / 100;

 x1=a2 % 1000;

 y1=a2 / 1000;

 x2=x1 % 10+(x1 / 10) % 10 +x1 / 100;

 y2=y1 % 10+(y1 / 10) % 10 +y1 / 100;

 if ((x3==y3) || (y2==x2)) s="Yes";

 else s="No";

fcout<<'\n'<<s;

 }

 fcin.close();

 fcout.close();

return 0;

}

**Код програми на Pascal**

var a,x,y,x1,y1,k,c,v,a1,a2,x2,x3,y3,y2,i:longint;

s:string;

f1,f2:text;

begin

 assign(f1,'input.txt');

 assign(f2,'output.txt');

 reset(f1);

 rewrite(f2);

 read (f1,k);

 for i:=1 to k do

 begin

 read(f1,a);

 a1:=a+1;

 a2:=a-1;

 x:=a1 mod 1000;

 y:=a1 div 1000;

 x3:=x mod 10+(x div 10) mod 10 +x div 100;

 y3:=y mod 10+(y div 10) mod 10 +y div 100;

 x1:=a2 mod 1000;

 y1:=a2 div 1000;

 x2:=x1 mod 10+(x1 div 10) mod 10 +x1 div 100;

 y2:=y1 mod 10+(y1 div 10) mod 10 +y1 div 100;

 if (x3=y3) or (y2=x2) then s:='Yes'

 else s:='No';

 writeln(f2,s);

 end;

close(f1);

close(f2);

end.

**Загадка**

*(Час 1 сек. Пам’ять16 Мб. Складність 18%)*

Петро і Катя – брат і сестра. Петро допомагає Катіз математики. Він загадує два натуральні числа **X** і **Y** (**X, Y≤1000**), а Катя повинна їх відгадати. Для цього Петро робить дві підказки. Він називає суму цих чисел **S** і їх добуток **P**. Допоможіть Каті відгадати задумані Петром числа.

**Вхідні дані**

Вхідний файл **INPUT.TXT** містить два натуральні числа **S** і **P**, розділених пропуском.

**Вихідні дані**

У вихідний файл **OUTPUT.TXT** виведіть два числа **Х** і **Y**, загадані Петром. Числа слід вивести в порядку неспадання своїх значень, розділені пропуском.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 4 4 | 2 2 |
| 2 | 5 6 | 2 3 |

**Код програми на С++**

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream fcin("input.txt");

ofstream fcout("output.txt");

 int s,p,x,y,x1;

fcin >>s>>p;

for (x=1; x< 1000; x++){

 y=s-x;

 x1=x\*y;

 if((x<=y) && (x1==p)) fcout<<x<<" "<<y;

}

 fcin.close();

 fcout.close();

return 0;}

**Код програми на Pascal**

program zagadka;

var s,p,x:integer;y,x1:longint;

f1,f2:text;

begin

assign(f1,'input.txt');

assign(f2,'output.txt');

rewrite(f2);

reset(f1);

readln(f1,s,p);

for x:=1 to 1000 do

begin

 y:=s-x;

 x1:=x\*y;

 if(x<=y) and (x1=p) then write(f2,x,' ',y);

end;

close(f1);

close(f2);

end.

**Остання цифра АВ**

*(Час 1 сек. Пам’ять16 Мб. Складність 21%)*

Необхідно написати програму, що знаходить цифру, на яку закінчується число **АВ**.

**Вхідні дані**

Вхідний файл **INTPUT.TXT** містить один рядок, у якому є два цілих числа **А** і **В**, розділені пробілами (**1<=A, B<=10000**).

**Вихідні дані**

В вихідний файл **OUTPUT.TXT** вивести цифру, на яку закінчується число **АВ**.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 2 2 | 4 |
| 2 | 3 7 | 7 |
| 3 | 24 9 | 4 |

**Код програми на С++**

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream fcin("input.txt");

ofstream fcout("output.txt");

 int a,b;

 fcin >>a>>b;

 a=a%10;

 int res=a;

 b--;

 for (int i=0;i<b;i++)

 {

 res=res\*a;

 res=res%10;

 }

fcout <<res;

fcin.close();

fcout.close();

return 0;}

**Код програми на Pascal**

var a,b,i:integer;

c:integer;

f1,f2:text;

begin

assign(f1,'input.txt');

assign(f2,'output.txt');

reset(f1);

rewrite(f2);

read(f1,a,b);

c:=1;

for i:=1 to b do

 begin

 c:=c\*a;

 c:=c mod 10;

 end;

end.

**Дім – Школа - Дім**

*(Час 1 сек. Пам’ять16 Мб. Складність 21%)*

Хлопчик Вася щодня їздить на метро. Уранці він їде в школу, а увечері того ж дня, назад зі школи, додому. Для того, щоб трохи заощадити, він купує електронну карту на **X** поїздок. Коли він хоче зайти в метро, він прикладає карту до турнікету. Якщо на карті залишилася ненульова кількість поїздок, то турнікет пропускає Васю і списує з карти одну поїздку. Якщо ж на карті не залишилося поїздок, то турнікет не пропускає Васю, і він (Вася) вимушений купити на цій же станції нову карту на **X** поїздок та знову пройти через турнікет.

Вася помітив, що у зв'язку з тим, що ранком метро переповнено, купувати нову карту уранці невигідно за часом, і він може запізнитися до школи. У зв'язку з цим він хоче зрозуміти: чи буде такий день, що з ранку, поїхавши в школу, Вася виявить у себе на картці нуль поїздок. Вася більше нікуди на метро не їздить і тому заходить в метро тільки на станції біля будинку і на станції біля школи.

 **Вхідні дані**

У вхідному файлі **INPUT.TXT** міститься рівно 2 рядки. У першій міститься слово " **School** " або "**Home**" залежно від того, де перший раз Вася купив картку на **X** поїздок. У другому рядку міститься натуральне число **X**, **1≤X ≤ 1000**.

**Вихідні дані**

У вихідний файл **OUTPUT.TXT** слід вивести "**Yes**", якщо буде такий день, що дома вранці у Васі на картці виявиться нуль поїздок і "**No**" –інакше.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | Home1 | YES |
| 2 | School2 | NO |

**Код програми на С++**

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream fcin("input.txt");

ofstream fcout("output.txt");

 string str;

 int n;

 fcin>>str>>n;

 if (str=="Home" || n%2==1) fcout<<"Yes";

else cout<<"No";

return 0;

}

**Код програми на Pascal**

var s:string; n:longint;

begin

assign(input, 'input.txt'); reset(input);

assign(output, 'output.txt'); rewrite(output);

readln(s);

readln(n);

if (s='School') and (n mod 2=0) then writeln('No');

if (s='School') and (n mod 2=1) then writeln('Yes');

if (s='Home') and (n mod 2=0) then writeln('Yes');

if (s='Home') and (n mod 2=1) then writeln('Yes');

end.

**Ділення на 11**

*(Час 1 сек. Пам’ять16 Мб. Складність 22%)*

Для ділення числа на 11 необхідно, щоб різниця між сумою цифр, що стоять на парних місцях, та сумою цифр, що стоять на непарних місцях, ділилась на 11.

Написати програму, яка перевіряє ділення заданого числа на 11.

**Вхідні дані**

Вхідний файл **INTPUT.TXT** містить одне натуральне число**N**,ділення якого необхідно перевірити (**1<=N<=1010000** ).

**Вихідні дані**

В вихідний файл **OUTPUT.TXT** вивести ”**YES**”, якщо число **N**ділиться на 11 або “**NO**”, якщо не ділиться.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 121 | YES |
| 2 | 1211 | NO |

**Код програми на С++**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

 int even=0,odd=0,i;

 int main()

 {

 freopen("input.txt","r",stdin); // перенаправить стандартний потік вводу із вхідного файлу

 freopen("output.txt","w",stdout); // перенаправить стандартний потік виводу в вихідний файл

 string a; //стандартний клас, який представляє текстовий рядок

 cin>>a;

 for(i=0;i<a.length();i++)

 {

 {if((i + 1) % 2 == 0)even+=a[i]-'0';//a[i]-'0' переведення цифри із символьної форми в числову

 if((i + 1)%2==1)odd+=a[i]-'0';}

 }

 if((odd-even)%11==0)cout<<"YES";

 else cout<<"NO";

 return 0;

 }

**Код програми на Pascal**

Programdiv\_11;

var

k : integer;

s : longint;

c : char;

begin

assign(input,'input.txt'); reset(input);

assign(output,'output.txt'); rewrite(output);

k:=1; s:=0;

while not eoln do

begin

read(c);

s:=s+k\*(ord(c)-48);//переводимо символ-цифру в число, віднявши код нуля, наприклад, ord(‘1’)-48=49-48=1

k:=-k

end;

if s mod 11=0 then write('YES')

else write('NO');

close(output)

end.

**Сума максимумів і мінімумів**

*(Час 1 сек. Пам’ять16 Мб. Складність 26%)*

Задана послідовність цілих чисел. Числа нумеруються по порядку, починаючи з одиниці. Написати програму, яка знайде суму максимуму з чисел з парними номерами і мінімуму з чисел з непарними номерами, – max{a2, a4}+min{a1, a3}.

**Вхідні дані**

Вхідний текстовий файл **INPUT.TXT** містить в єдиному рядку послідовність від **2** до **2×105** цілих чисел, які по модулю не перевищують **10000.**

**Вихідні дані**

Вихідний текстовий файл **OUTPUT.TXT** повинен містити одно ціле число – суму максимуму з чисел з парними номерами і мінімуму з чисел з непарними номерами.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 1 2 | 3 |
| 2 | 1 -2 3 -4 5 | -1 |

**Код програми на С++**

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream fcin("input.txt");

ofstream fcout("output.txt");

int a, max, min,n;

max=-10000; min=10000; n=0;

while (!fcin.eof())

{

fcin>>a; n=n+1;

if (n%2!=0) {if (a<min) min=a;}

else if (a>max) max=a;

}

fcout<<(max+min);

fcin.close();

fcout.close();

 return 0;}

**Код програми на Pascal**

vara, max, min : integer;

n : longint;

begin

assign(input,'input.txt'); reset(input);

assign(output,'output.txt'); rewrite(output);

max:=-10000; min:=10000; n:=0;

while not eof do

begin

read(a); n:=n+1;

if odd(n) then if a<min then min:=a else

else if a>max then max:=a

end;

write(max+min)

end.

**Домашнє завдання**

*(Час 1 сек. Пам’ять16 Мб. Складність 27%)*

Петро устигає з математики краще за всіх у класі, тому учитель задав йому складне домашнє завдання, в якому треба в заданому наборі цілих чисел знайти суму усіх позитивних елементів, потім знайти, де в заданій послідовності знаходяться максимальний і мінімальний елемент і обчислити добуток чисел, розташованих між ними. Так само відомо, що мінімальний і максимальний елемент зустрічаються в заданій послідовності чисел тільки один раз. Оскільки завдань такого роду учитель дав Петру близько ста, то Петро як сильний програміст зміг написати програму, яка по заданому набору чисел самостійно знаходить рішення. Спробуйте і Ви.

**Вхідні дані**

У першому рядку вхідного файлу **INPUT.TXT** записане число **N**–кількість елементів масиву. Другий рядок містить **N** цілих чисел, що представляють заданий масив. Усі елементи масиву розділені пропуском. Кожне з чисел у вхідному файлі не перевищує **102** за абсолютною величиною.

**Вихідні дані**

У єдиний рядок вихідного файлу **OUTPUT.TXT** треба вивести два числа, розділених пропуском: суму позитивних елементів і добуток чисел, розташованих між мінімальним і максимальним елементами. Значення суми і добутку не перевищують по модулю 3\*104.

**Приклад**

|  |  |  |
| --- | --- | --- |
| № | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 5-7 5 -1 3 9 | 17 -15 |
| 2 | 83 14 -9 4 -5 1 -12 4 | 26 180 |
| 3 | 10-5 1 2 3 4 5 6 7 8 -3 | 36 5040 |

**Код програми на С++**

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

 int n = 0;

 int mul = 1, sum = 0;

 int min = 0;

 int max = 0;

ifstream fcin("input.txt");

ofstream fcout("output.txt");

 fcin>>n;

 int vector[n];

 for (int i = 0; i<n ;i++)

 {

 fcin>>vector[i];

 if (vector[i] > 0) sum += vector[i];

 if (vector[i] < vector[min]) min = i;

 if (vector[i] > vector[max]) max = i;

 }

 if (min > max)

 {

 n = min;

 min = max;

 max=n;

 }

 for (int i = min+1; i<max; i++)

mul \*= vector [i];

 fcout <<sum<<" "<<mul;

fcin.close();

fcout.close();

return 0;

}

**Код програми на Pascal**

var a: array [1..10000] of integer;

 n: integer;

 t: text;

 max,i, min, sum, p, tmp: longint;

begin

 max:=1;

 min:=1;

 p:=1;

 sum:=0;

 assign(t,'input.txt');

 reset(t);

 readln(t,n);

 for i:=1 to n do

 begin

 read(t, a[i]);

 if (a[i]>a[max]) then max:=i;

 if (a[i]<a[min]) then min:=i;

 if (a[i]>0) then sum:=sum+a[i];

 end;

 close(t);

 if (min>max) then

 begin

 tmp:=min;

 min:=max;

 max:=tmp;

 end;

 for i:=(min+1) to max-1 do

 p:=p\*a[i];

 assign(t,'output.txt');

 rewrite(t);

 writeln(t, sum, ' ', p);

 close(t);

end.

**Двовимірні масиви**

**Перестановка**

Ввести матрицю **X** розміром**n x n.** Поміняти місцями рядки матриці, номера яких вводяться з клавіатури, і вивести результат на екран.

**Код програми на С++**

#include <iostream>

#include <cstdlib>

using namespace std;

const int n = 3; //оголошуємо константу для масиву

int main ()

{

 setlocale(LC\_ALL,"rus");

int i,j,t,r1,r2; //змінні для циклу

// оголошення масив цілого типу, ініціалізація иасиву

int X[n][n] = {{11,12,13}, {21,22,23}, {31,32,33}};

cout<<"\*\*\*\*\*\*\*\*\*вхідний масив\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

for (i = 0; i < n ; i++)

 { //цикл виводу масиву

 cout<<"\n"; //перехід на наступний рядок

 for (int j = 0; j < n ; j++)

 cout<<X[i][j]<<"\t"; //на екран виводиться результат вводу, \t - горизонтальна табуляція

 }

 //ввести номери рядків для перестановки

 cout<<endl<<"введіть номери рядків для перестановки"<<endl<<"r1="; cin>>r1;

 cout<<"r2="; cin>>r2;

 // преставити місцями рядки

for (int j = 0; j < n; j++)

{

int t = X [r1] [j];

X [r1] [j] = X [r2] [j];

X [r2] [j] = t;

}

// вивести масив порядково

cout<<endl<<"\*\*\*\*\*\*\*\*\*переставлений масив\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

for (i = 0; i < n ; i++)

 { //цикл виводу масиву

 cout<<"\n";

 for (int j = 0; j < n ; j++)

 cout<<X[i][j]<<"\t"; //на екран виводиться результат \t - горизонтальна табуляція

 }

 return 0; }

**Заповнення масиву одиницями (верхній трикутник,
обмежений діагоналями)**

Ввести матрицю **X** розміром**n x n.**Заповнити масив нулями. Заповнити одиницями частину масиву, яка обмежена головною та побічною діагоналями (верхній трикутник).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **1** | **1** | **1** | **1** |
| 0 | **1** | **1** | **1** | 0 |
| 0 | 0 | **1** | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

**Код програми на С++**

#include <iostream>

#include <cstdlib>

using namespace std;

const int n = 5; //оголошуємо константу для масиву

int main ()

{

 setlocale(LC\_ALL,"rus");

int i,j; //змінні для циклу

// оголошення масиву цілого типу, ініціалізація масиву

int X[n][n];

cout<<"\*\*\*\*\*\*\*\*\*вхідний масив\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

for (i = 0; i < n ; i++)

 for (int j = 0; j < n ; j++)

 X[i][j]=0;

for (i = 0; i < n ; i++)

 { //цикл виводу масиву

 cout<<"\n";

 for (int j = 0; j < n ; j++)

 cout<<X[i][j]<<"\t"; //на екран виводиться результат \t - горизонтальна табуляція

 }

 for (i=0; i<n / 2+1; i++)

 for (j=i; j< n-i; j++)

X[i][j]=1;

cout<<endl<<"\*\*\*\*\*\*\*\*\*новий масив\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

for (i = 0; i < n ; i++)

 { //цикл виводу масиву

 cout<<"\n";

 for (int j = 0; j < n ; j++)

 cout<<X[i][j]<<"\t"; //на екран виводиться результат \t - горизонтальна табуляція

 }

 return 0; }

**Код програми на Pascal**

Constn=5;

Var x:array[1..n,1..n] of integer; i,j:integer;

begin

For i:=1 to n do

 For j:=1 to n do

 x[i,j]:=0 ;

 For i:=1 to n do

 begin

 For j:=1 to n do

 write(x[i,j]:4);

 writeln;

 end;

 writeln; writeln;

 For i:=1 to n div 2+1 do

 For j:=i to n-i+1 do

 x[i,j]:=1;

For i:=1 to n do

 begin

 For j:=1 to n do

 write(x[i,j]:4);

 writeln;

end;

end.

**Заповнення масиву одиницями (нижній трикутник,
обмежений діагоналями)**

Ввести матрицю **X** розміром**n x n.**Заповнити масив нулями. Заповнити одиницями частину масиву, яка обмежена головною та побічною діагоналями (верхній трикутник).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | **1** | 0 | 0 |
| 0 | **1** | **1** | **1** | 0 |
| **1** | **1** | **1** | **1** | **1** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

**Код програми на С++**

#include <iostream>

#include <cstdlib>

using namespace std;

const int n = 5; //оголошуємо константу для масиву

int main ()

{

 setlocale(LC\_ALL,"rus");

int i,j; //змінні для циклу

// оголошення масиву цілого типу, ініціалізація масиву

int X[n][n];

cout<<"\*\*\*\*\*\*\*\*\*вхідний масив\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

for (i = 0; i < n ; i++)

 for (int j = 0; j < n ; j++)

 X[i][j]=0;

for (i = 0; i < n ; i++)

 { //цикл виводу масиву

 cout<<"\n";

 for (int j = 0; j < n ; j++)

 cout<<X[i][j]<<"\t"; //на екран виводиться результат \t - горизонтальна табуляція

 }

 for (i=n / 2-1; i< n; i++)

 for(j=n-i-1;j< i+1; j++)

X[i][j]=1;

cout<<endl<<"\*\*\*\*\*\*\*\*\*новий масив\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

for (i = 0; i < n ; i++)

 { //цикл виводу масиву

 cout<<"\n";

 for (int j = 0; j < n ; j++)

 cout<<X[i][j]<<"\t"; //на екран виводиться результат \t - горизонтальна табуляція

 }

 return 0; }

**Код програми на Pascal**

Const n=5;

Var x:array[1..n,1..n] of integer; i,j:integer;

begin

For i:=1 to n do

 For j:=1 to n do

 x[i,j]:=0 ;

 For i:=1 to n do

 begin

 For j:=1 to n do

 write(x[i,j]:4);

 writeln;

 end;

 writeln; writeln;

 For i:=n div 2+1 to n do

 For j:=n-i+1 to i do

 x[i,j]:=1;

For i:=1 to n do

 begin

 For j:=1 to n do

 write(x[i,j]:4);

 writeln;

 end;

 end.

# Додаток Г

**Збірник задач для самостійного розв’язку**

**Арифметичні вирази**

**Варіант 1. Арифметичні вирази. Лінійні програми.**

1. Скласти програму обчислення виразу: К=.

Результат надрукувати у вигляді ЗНАЧЕННЯ К=.

2. Задано координати вершин трикутника (х1,у1),(х2,у2),(х3,у3). Скласти програму для знаходження медіан трикутника.

**Варіант 2. Арифметичні вирази. Лінійні програми.**

1. Скласти програму обчислення виразу Х=.

Результат надрукувати у вигляді РЕЗУЛЬТАТ: Х=.

2. Задано координати вершин трикутника (х1,у1),(х2,у2),(х3,у3). Скласти програму для знаходження довжин середніх ліній трикутника. (Середня лінія паралельна основі трикутника та дорівнює її половині;)

**Варіант 3. Арифметичні вирази. Лінійні програми.**

1. Скласти програму обчислення виразу М=.

2. Задано координати вершин трикутника (х1,у1),(х2,у2),(х3,у3). Скласти програму для знаходження периметру і площі трикутника.

**Варіант 4. Арифметичні вирази. Лінійні програми.**

1. Скласти програму обчислення виразу : В=.

Результат надрукувати у вигляді РЕЗУЛЬТАТ: ЗНАЧЕННЯ В=.

1. Задано координати вершин прямокутного трикутника (х1,у1),(х2,у2),(х3,у3). Скласти програму для обчислення площі, периметру і гіпотенузи прямокутного трикутника.

**Варіант 5. Арифметичні вирази. Лінійні програми.**

1. Скласти програму обчислення виразу: Z=.

2. Задано координати вершин прямокутника (х1,у1),(х2,у2),(х3,у3),(х4,у4). Скласти програму для обчислення площі і периметру прямокутника.

**Варіант 6. Арифметичні вирази. Лінійні програми.**

1. Скласти програму обчислення виразу: У=.

2. Задано координати вершин прямокутника (х1,у1),(х2,у2),(х3,у3),(х4,у4). Скласти програму для обчислення довжини діагоналей прямокутника.

**Вказівки розгалуження та повторення**

**Варіант 1. Вказівки розгалуження та повторення**

1. Скласти програму для обчислення виразу у=.

2. Написати програму для табулювання функції у=cos2x на проміжку [0;3,14] з кроком 0,25 і обчислити кількість і добуток значень, що задовольняють умові 0,5<у<1

**Додаткове завдання.** Обчисліть середнє арифметичне усіх значень функції.

**Варіант 2. Вказівки розгалуження та повторення**

1. Ввести три числа . Якщо вони рівні між собою, надрукувати одне з них, інакше - всі числа, але на окремих рядках.

2. Написати програму для табулювання функції у=sin3x на проміжку [-1;1], з кроком 0,2 і обчислити кількість і суму від’ємних значень функції.

**Додаткове завдання.** Обчисліть суму першого та максимального значень функції.

**Варіант 3. Вказівки розгалуження та повторення**

1. Написати програму обчислення виразу у=.

2. Написати програму для табуляції функції у=соs2х на проміжку [-2;2 ] з кроком 0,25 і обчислити кількість і суму додатніх значень функції.

**Додаткове завдання.** Обчисліть суму мінімального та останнього значень функції.

**Варіант 4. Вказівки розгалуження та повторення**

1. Написати програму обчислення коренів квадратного рівняння.

2. Написати програму для табулювання функції у=5хsin2x на проміжку [1;3], з кроком 0,2 і обчислити кількість і суму значень функції, які задовольняють умові –2<y<2 .

**Додаткове завдання.** Обчисліть суму першого та останнього значень функції.

**Варіант 5. Вказівки розгалуження та повторення**

1. Визначити, чи є відрізки а, в, с сторонами трикутника. Який це трикутник:

 рівносторонній, рівнобедренний чи різносторонній?

2. Написати програму для табулювання функції у=sin4xcos3x на проміжку [1;4], з кроком 0,25 і обчислити кількість і добуток значень функції, які менші за 0.

**Додаткове завдання.** Обчисліть добуток першого і максимального значень функції.

**Варіант 6. Вказівки розгалуження та повторення**

1. Знайти найменше з трьох заданих чисел х,у,с.

1. Написати програму для табулювання функції у=3х+sin4xcos3x-(х-3.6) на проміжку [1;4], з кроком 0,25 і обчислити кількість і добуток значень функції, які менші за 3 або більші за 6.

**Додаткове завдання.** Обчисліть суму останнього і мінімального значень функції.

**Лінійні задачі 1**

*1. Умова:* Дано два дійсних числа *a* та *b*. Обчислити їх суму, різницю, добуток.
Необхідні змінні:
*a*, *b* - задані числа;
*Add* - сума чисел;
*Sub* - різниця чисел;
*Multy* - добуток чисел.

2. *Умова*: Визначити, яку платню одержить на фірмі сумісник за виконану роботу, якщо йому нараховано *S* грн., а податок становить 20 відсотків.
Необхідні змінні:
*S* - сума нарахувань сумісника;
*P* - реальна платня, що він одержить у касі (за умовою вона становить 80% від нарахувань).

*3. Умова:* Якщо на одну шальку терезів посадити Даринку, яка важить *n* кг, і Наталку, яка важить на 5 кг менше, а на іншу насипати m кг цукерок, то скільки кілограмів цукерок доведеться з'їсти дівчаткам, щоб шальки терезів врівноважились.
Введемо наступні змінні для зберігання необхідних результатів:
N - вага Даринки;
M - вага цукерок;
P - вага цукерок, що необхідно з'їсти дівчинкам.

*4.Умова:* Дано гіпотенуза і один з катетів прямокутного трикутника. Знайти другий його катет.
Необхідні змінні:
*a* - катет прямокутного трикутника;
*c* - гіпотенуза прямокутного трикутника;

*5.Умова:* Дано три цілі числа. Знайти середнє арифметичне цих чисел.
Необхідні змінні:
*a* – перше число; *b* – друге число; *c*- третє число.

6. Складіть програму «Сніданок на замовлення». На екран виводиться меню з 5 страв (почергово). Проти кожної страви вказана її ціна. Програма виводить на екран рахунок за сніданок і бажає замовнику смачного.

7. Складіть програму, яка за даними про власника квартири, її площу, кількість мешканців і витрати електроенергії здійснює розрахунок платні за комунальні послуги й виводить на екран відповідне повідомлення на ім’я власника квартири.

8. Знайти суму цифр трьохзначного числа.

9. Написати програму, яка здійснює перестановку двох цілих чисел, які вводяться з клавіатури.

**Лінійні задачі 2**

1. Написати програму обчислення площі паралелограма. Нижче представлений рекомендований вигляд екрана під час роботи програми (дані, введені користувачем, виділені жирним шрифтом).
**Обчислення площі паралелограма. Введіть вихідні дані: Довжина (см)- > 9 Ширина (см)- > 7.5**
Площа паралелограма: 67.50 кв.см.

2. Написати програму обчислення об’єму паралелепіпеда. Нижче представлений рекомендований вигляд екрана під час роботи програми (дані, введені користувачем, виділені жирним шрифтом).

**Обчислення об’єму паралелепіпеда.
Введіть вихідні дані:
Довжина (см)- > 9
Ширина (см)- > 7.5
Висота (см)- > 5**

Об'єм: 337.50 куб.см.

3.Написати програму обчислення площі поверхні паралелепіпеда. Нижче представлений рекомендований вигляд екрана під час роботи програми (дані, введені користувачем, виділені жирним шрифтом).

 **Обчислення площі поверхні паралелепіпеда.
Введіть вихідні дані:
Довжина (см)- > 9
Ширина (см)- > 7.5
Висота (см)- > 5**Площа поверхні: 90.00 кв.см.

4. Написати програму обчислення об’єму куба. Нижче представлений рекомендований вигляд екрана під час роботи програми (дані, введені користувачем, виділені жирним шрифтом).

**Обчислення об’єму куба.
Введіть довжину ребра (см) і натисніть < Enter>
- > 9.5
\***Обсяг куба: 857.38 куб.см.

5. Написати програму обчислення об’єму циліндра. Нижче представлений рекомендований вигляд екрана під час роботи програми (дані, введені користувачем, виділені жирним шрифтом).

**Обчислення об’єму циліндра
Введіть вихідні дані:
Радіус підстави (см)- > 5
Висота циліндру (см)- > 10**Об'єм циліндра 1570.80 куб. см.

**Лінійні задачі 3**

* Напишіть програму, яка виводить на екран окремо цілу та дробову частину введеного дійсного числа х. (наприклад, 12,82 —>12 і 82)
* Обчисліть суму квадратів двох введених чисел, (наприклад, 2 і 3 —>13)
* Знайдіть значення функції у=2х2+7х-3, при х=3; -5,2; 14,85.
* Визначте суму і добуток цифр двозначного числа, (наприклад, 25 —>7 і 10)
* Визначте суму і добуток цифр тризначного числа, (наприклад, 111 —>3 і 1 )
* Визначте число, утворене з тризначного числа переставленням цифр його сотень та одиниць, (наприклад, 123 —» 321)
* Присвоїти цілій змінній h третю від кінця цифру в запису додатнього цілого числа k (наприклад, якщо k=130985, то h=9).
* Присвоїти цілій змінній d першу цифру з дробної частини додатнього дійсного числа х (так, якщо х=32. 597, то d=5).
* Цілій змінній s присвоїти суму цифр тризначного цілого числа k.
* Йде k-та секунда доби. Визначити, скільки повних годин (h) і повних хвилин (m) пройшло до цього моменту (наприклад, h=3 і m=40, якщо k=13257=3\*3600+40\*60+57).
* Визначити f- кут (у градусах) між положенням у h годин, m хвилин і s секунд (0<=h<=11, 0<=m, s<=59).
* Визначити h- повна кількість годин і m- повна кількість хвилин, яки минули від початку доби до того моменту (у першій половині дня), коли годинна стрілка повернулася на f градусів (0<=f<360, f- дійсне число).
* Нехай k- ціле число від 1 до 365. Присвоїти цілій змінній n значення 1, 2, ... , 6 або 7 у залежності від того, на який день тижня (понеділок, вівторок, ... , суботу або неділю) припадає k-й день невисокосного року, у якому 1 січня - понеділок.
* Змінити місцями значення цілих змінних х та у.
* Задано натуральне число n. Визначити кількість сотень у ньому.
* Якщо на одну шальку терезів посадити Даринку, яка важить *n* кг, і Наталку, яка важить на 5 кг менше, а на іншу насипати m кг цукерок, то скільки кілограмів цукерок доведеться з'їсти дівчаткам, щоб шальки терезів врівноважились.
Введемо наступні змінні для зберігання необхідних результатів:
N - вага Даринки;
M - вага цукерок;
P - вага цукерок, що необхідно з'їсти дівчинкам.
* Дано два дійсних числа *a* та *b*. Обчислити їх суму, різницю, добуток.
Необхідні змінні:
*a*, *b* - задані числа;
*Add* - сума чисел;
*Sub* - різниця чисел;
*Multy* - добуток чисел.
* Дано дійсні числа x та y. Обчислити значення виразу: 
Необхідні змінні:
*x*, *y* - задані числа;
*Rez* - результат обчислень.
* Дано гіпотенуза і один з катетів прямокутного трикутника. Знайти другий його катет і площу вписаного круга.
Необхідні змінні:
*a* - катет прямокутного трикутника;
*c* - гіпотенуза прямокутного трикутника;
*b* - довжина невідомого катета;
*S* - площа вписаного круга.
*Математична довідка:*
* другий катет прямокутного трикутника знаходиться за теоремою Піфагора *a*^2 + *b*^2 = *c*^2, звідки випливає, що катет дорівнює: 
* площа вписаного круга обчислюється за наступною формулою: 
* Обчислити відстань між двома точками з координатами (*x1*; *y1*) та (*x2*; *y2*).
Необхідні змінні:
*x1*, *y1* - координати першої точки;
*x2*, *y2* - координати другої точки
*Математична довідка:*
Відстань між двома точками обчислюється за теоремою Піфагора, як катет прямокутного трикутника, побудованого наступним чином:
* 
* Тоді відстань між точками з координатами (*x1*, *y1*) та (*x2*, *y2*) буде обчислюватись за наступною формулою: 
Враховуючи, що різниця між координатами точок підноситься до квадрату, можна не звертати увагу на послідовність координат в різниці.
* Визначити, яку платню одержить на фірмі сумісник за виконану роботу, якщо йому нараховано *S* грн., а податок становить 20 відсотків.
Необхідні змінні:
*S* - сума нарахувань сумісника;
*P* - реальна платня, що він одержить у касі (за умовою вона становить 80% від нарахувань).
* Дано значення змінних *x*, *y*, *z*. Обчислити значення змінної *t*:





Необхідні змінні:
*x*, *y*, *z* - задані числа;
*t1*, *t2*, *t3*, *t4* - результати обчислень за відповідними формулами завдання 77 підпункти 1), 2), 3), 4).
*Математична довідка:*
Для обчислення будь-якої степені будь-якого числа можна скористатися такою формулою: 

**Лінійні задачі 4**

1. Місто А знаходиться в ***х*** милях від Нью-Йорку. Напишіть програму, яка обчислює цю відстань в кілометрах.
2. Петрик запропонував своє тлумачення «щасливого» квитка. Для нього він «щасливий» тільки тоді, коли сума його цифр ділиться на 5. Перевірити, чи куплений Петриком квиток є щасливим.
3. Складіть програму розв'язування рівняння виду ax+b=0***.***
4. Визначити площу прямокутника, якщо відомо, що одна сторона більша за другу на х см, а периметр дорівнює р см. Вказівка: після спрощення формула для визначення площі прийме вигляд S=(p2-x2)/16.
5. Знайти суму перших n членів арифметичної прогресії {An}, якщо відомі А***1*** і D.
6. Хлопчик, що торгує на базарі газетами, заробляє а грн. від продажу перших 100 газет. На кожній з інших проданих газет він заробляє по х коп. Напишіть програму, яка визначає заробіток за день хлопчика, якщо він продасть у газет (у>100).
7. Реактивний літак перевозить групу футбольних болільників на чемпіонат світу з Києва до Парижу. Три чверті пасажирів мають білет другого класу вартістю х доларів кожен. Всі інші пасажири мають квитки першого класу, які вдвічі дорожчі від білетів другого класі. Напишіть програму, яка виводить суму (в гривнях), що отримує авіакомпанія від продажу квитків на цей рейс.
8. На протязі місяця продавець доставляв додому 4 л молока в день. В березні молоко коштувало х коп. за літр. З першого квітня ціна підвищилась до (х+а) коп. за літр. Скільки потрібно заплатити продавцю за все доставлене молоко в кінці квітня? Кількість молока, що купується щоденно, залишилась тією ж.
9. Хлопчик може бігати в три рази швидше, ніж ходити. Швидкість його ходьби становить 4 км/год. Він прийняв участь в марафонському забігу, але зійшов з дистанції, пробігши тільки х км. Який час він затратив на подолання цієї відстані?
10. Робітник заробляє х грн. за кожні 38 годин своєї роботи. Йому платять в 1,5 рази більше за кожну годину понад 38 годин. Яку суму він отримає, якщо пропрацює а годин (а>38)?
11. Людині потрібно з'їздити з Лондону в розміщений в 390 милях Едінбург. Вона може їхати на автомобілі марки “Роллс-Ройс” або на автомобілі марки “Форд Ескорт”. “Роллс-Ройс” витрачає 1 галон пального на кожні 15 миль шляху. “Форд Ескорт” витрачає 1 галон пального на кожні 36 миль шляху. Скільки буде коштувати поїздка до Единбургу на “Роллс-Ройсі”, якщо вартість 1 галону бензину становить х фунтів? Скільки грошей він зекономить, якщо замість цього поїде на автомобілі марки “Форд Ескорт”?
12. Магазин продає b автомобілів по ціні 5627 грн. за кожен. Знайти загальну виручку від продажу автомобілів.
13. Щоденно на молочній фермі надоюють по 1842 л молока. Знайдіть кількість молока, отриманого за довільний місяць (вводиться число днів в місяці).
14. Заробіток працівників на підприємстві склав ***х*** грн. Його потрібно поділити порівну між а працівниками. Виведіть заробіток кожного працівника.
15. В магазині продається тканина для костюму. Її ціна b грн. за кв. метр. Напишіть програму, яка підраховує і виводить на екран вартість куска цієї тканини довжиною х м і шириною 80 см.
16. Людина бажає обклеїти шпалерами довгу стінку в своїй квартирі. Довжина стінки а м, а висота b м. Рулон шпалер має довжину 12 м і ширину 80 см. Скільки будуть коштувати шпалери для всієї стінки, якщо ціна одного рулону k грн.?
17. Обчисліть заробітну плату працівника при погодинній системі оплати, якщо відомі тарифна ставка за 1 годину і кількість відпрацьованих годин за місяць.
18. Складіть програму знаходження площі трикутника за його основою та висотою.
19. Складіть програму знаходження площі ромба за його стороною і гострим кутом (кут попередньо переведіть в радіани). S=a2Sinf, де a – сторона, f – кут.
20. Складіть програму знаходження суми всіх натуральних чисел від ***1*** до n (використайте формулу суми членів арифметичної прогресії). ******, де а***1***– перший член послідовності, а d – різниця.
21. Знайдіть площу круга, якщо відомо радіус кола.
22. В прямокутному трикутнику відомі катет і гіпотенуза. Знайдіть інший катет.
23. В арифметичній прогресії відомі 1-й член послідовності і різниця. Знайдіть 30-й член послідовності і суму перших 40 членів.
24. Знайдіть координати кінців відрізку, заданого координатами його кінців.
25. За двома сторонами і кутом між ними в трикутнику АВС знайдіть два інших кути і третю сторону. , де а, b – відомі сторони, аf – кут між ними.
26. Складіть програму обчислення рівнодійної сил F, що діють на тіло масою m, яке рухається з прискоренням а. F = ma.
27. Складіть програму обчислення швидкості тіла в момент часу t при рівноприскореному русі. Відомі початкова швидкість V***0*** і прискорення а, при умові, що тіло рухається прямолінійно. V***t***=V***0***+at.
28. Складіть програму обчислення опору R в колі електричного струму за даними значеннями струму І та напруги U. .
29. Складіть програму визначення висоти h, на підніметься тіло, що має масу m, кинуте вертикально вгору з початковою швидкістю V**0**. .
30. Складіть програму обчислення об'єму прямокутного паралелепіпеда за його сторонами.
31. Складіть програму для обчислення значення виразу: . Відомо, що z № 0 i y № 0.
32. Складіть програму запиту даних та друку результатів для визначення середньої економії пального та мастил в автобусному парку за місяць, якщо відома середня економія за день по бензину, дизельному паливу та мастилам.
33. Для відливки циліндричної деталі з чавуну необхідно визначити її об'єм. Складіть програму для розв'язання цієї задачі.
34. Четверо друзів повечеряли в ресторані. Офіціант подав їм рахунок на ***х*** грн. кожному. Друзі вирішили залишити офіціанту чайові в розмірі 15% від рахунку. Складіть програму, яка виводить на екран суму чайових, яку отримав офіціант.
35. Кожен тиждень Юрко отримує гроші на дрібні витрати. З них він витрачає х коп. на солодощі, що становить одну чверть того, що він отримує щоденно. Юрко вирішив зберігати одну третину того, що залишається після придбання солодощів. Складіть програму, що яка підраховує суму, накопичену Юрком за рік.

**Типи даних**

**Варіант 16**

**Обчислити значення і визначити тип таких виразів:**

SQR(5.0);

SIN(0);

TRUNC(-3.14);

SQRT(30+INT(6.3));

**Варіант 15**

**Обчислити значення і визначити тип таких виразів:**

PRED(‘Г’);

SUCC(‘Г’);

SQR(2)+SQRT(35+1);

18/18;

**Варіант 14**

**Обчислити значення і визначити тип таких виразів:**

TRUNC(-1.5);

ROUND(-1.5);

28+1.1;

SQRT(8+INT(41.5);

**Варіант 13**

**Обчислити значення і визначити тип таких виразів:**

10+SQR(2+1);

25+ABS(29-30);

ROUND(0.5);

21 DIV (ROUND(PI+2));

**Варіант 11**

**Обчислити значення і визначити тип таких виразів:**

28 MOD 29;

28 DIV 29;

SQRT(8+INT(41.5));

PRED(30);

**Варіант 12.**

**Обчислити значення і визначити тип таких виразів:**

TRUNC(10.9);

ROUND(10.9);

21 DIV (ROUND(PI+1));

SUCC(‘Ю’);

**Варіант 10.**

**Обчислити значення і визначити тип таких виразів:**

225 DIV 0;

225 MOD 0;

SQR(4);

PRED(‘Д’);

**Варіант 9.**

**Обчислити значення і визначити тип таких виразів:**

5.9 MOD 5;

18/3;

7+7.1;

SUCC(8);

**Варіант 8.**

**Обчислити значення і визначити тип таких виразів:**

31 MOD 16;

31 DIV 16;

SQRT(16);

5-ABS(28-29);

**Варіант 7.**

**Обчислити значення і визначити тип таких виразів:**

TRUNC(21.01);

ROUND(21.01);

2 DIV 5;

SIN(0);

**Варіант 6.**

**Обчислити значення і визначити тип таких виразів:**

ROUND(7.8);

TRUNC(7.8);

18 MOD 3;

1+ABS(10-12);

**Варіант 5.**

**Обчислити значення і визначити тип таких виразів:**

TRUNC(-2.8);

ROUND(-2.8);

20 MOD 21;

1+ABS(5-8);

**Варіант 4.**

**Обчислити значення і визначити тип таких виразів:**

ROUND(0.5);

ROUND(-0.5);

3.0 MOD 3;

SQR(5.0);

**Варіант 3.**

**Обчислити значення і визначити тип таких виразів:**

TRUNC(-1.8);

ROUND(-1.8);

1+0.0;

SUCC(-2);

**Варіант 2.**

**Обчислити значення і визначити тип таких виразів:**

TRUNC(6.2);

ROUND(6.2):

20 MOD 4;

20/4;

**Варіант 1.**

**Обчислити значення і визначити тип таких виразів:**

TRUNC(6.9);

ROUND(6.9);

123 DIV 0;

SIN(0);

1. Щоб перейти по заданому гіперпосиланню, потрібно натиснути клавішуCtrl та ліву кнопку миші одночасно. [↑](#footnote-ref-1)