**І. Р. МИХАЙЛЮК, Т. О. ВАВРИК**

**ОСНОВИ ПРОГРАМУВАННЯ**

**ЛАБОРАТОРНИЙ ПРАКТИКУМ**

**Івано-Франківськ**

**2024**

**Івано-Франківський національний технічний**

**університет нафти і газу**

**Кафедра інженерії програмного забезпечення**

І. Р. МИХАЙЛЮК, Т. О. ВАВРИК

**ОСНОВИ ПРОГРАМУВАННЯ**

ЛАБОРОТОРНИЙ ПРАКТИКУМ

**Івано-Франківськ**

**2024**

УДК 681.3201

М –

**Рецензент:**

**Шекета В. І.** д. т. н, проф. каф. інженерії програмного забезпечення ІФНТУНГ

*Рекомендовано методичною радою університету*

*(протокол № від р.)*

**Михайлюк І. Р., Ваврик Т. О.** Основи програмування. Лабораторний практикум для студентів спеціальності 121 Інженерія програмного забезпечення - Івано-Франківськ: Факел, 2024.- 152 с.

МВ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ - 2024

Лабораторний практикум складається з набору задач (пов’язаних між собою за принципом “з низу до гори”), прикладу розв’язування задачі, пояснення до програми. Розв’язуючи задачі студенти набувають практичних навичок з алгоритмізації та складання програм мовою Сі. Окремі розділи містять задачі, розв’язання яких вимагає застосування процедур, функцій користувача, вміння працювати з файлами даних В кінці кожної лабораторної роботи є контрольні питання самоконтролю для закріплення вивченого матеріалу. Список використаної літератури наведено в кінці лабораторного практикуму. Студенти виконуть завдання згідно варіанту.

Адресовано студентам денної та заочної форм навчання за спеціальністю 121 Інженерія програмного забезпечення

УДК 681.3201

МВ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -2024

© ІФНТУНГ, 2024

© Михайлюк І. Р.

# © Ваврик Т. О.

**ЗМІСТ**

|  |  |
| --- | --- |
| **1 Лабораторна робота №1** Ознайомлення з середовищем розроблення програм на С. Введення/виведення даних ……………  **2 Лабораторна робота №2** Прості обчислення. Основні математичні функції стандартної бібліотеки із заголовкого файлу <math.h> ………………………………………………………………..  **3 Лабораторна робота №3** Розгалужені програми ………….……  **4 Лабораторна робота №4**  Використання керуючих конструкцій мови С. Циклічні програми …………………………………………..  5 **Лабораторна робота №5**  Використання керуючих конструкцій мови С. Циклічні програми,що містять розгалуження ……………..  6 **Лабораторна робота №6**  Масив як структурований тип. Одновимірні масиви ……………...........................................................  7 **Лабораторна робота №7.** Циклічні програми з накопиченням суми або добутку ……………………………………………………………..  8 **Лабораторна робота №8.** Масив як структурований тип. Двовимірні масиви …………………………………………..…………  9 **Лабораторна робота №9** Вкладені цикли ………………………...  10 **Лабораторна робота №10** Функції. Механізм передачі параметрів Глобальні та локальні змінні ……………………………..  11 **Лабораторна робота №11** Структури на С ………………………  12 **Лабораторна робота №12** Робота з файлами мовою С …………  **Перелік використаних джер** ……………………………………….. | 5  13  25  34  45  59  71  79  93  102  127  1  34  152 |

**Лабораторна робота №1**

**ОЗНАЙОМЛЕННЯ З СЕРЕДОВИЩЕМ РОЗРОБЛЕННЯ ПРОГРАМ НА С. ВВЕДЕННЯ/ВИВЕДЕННЯ ДАНИХ**

**Мета роботи:** навчитися створювати прості програми на С, вивчити типи даних та операції введення/виведення.

**Теоретичні відомості**

***Типи даних*.** Базовими (фундаментальними, вбудованими) типами даних мови С є такі:

|  |  |  |  |
| --- | --- | --- | --- |
| **тип** | **назва** | **Розмір(біти)** | **Інтервал** |
| char | символьний | 8 | -128 до 127 |
| int | цілий | 16 | -32768 до 32767 |
| float | дійсний | 32 | 3.4Е-38 до 3.4Е+38 |
| double | дійсний з подвійною точністю | 64 | 1.7Е-308 до 1.7Е+308 |

На основі цих типів додаванням модифікаторів будуються інші типи даних:

Signed – знаковий, unsigned – без знака,

Long – довгий, short – короткий.

**Наприклад**: signed short int.

Окремого **логічного** типу в С++ не має, але будь-який цілочисельний тип може бути застосований як логічний: 0 – „хибність”, 1 – „істина”.

***Оголошення даних*** .Усі змінні перед використанням у програмі потрібно об’явити.

**Тип <список\_змінних>;**

Тип – будь-який тип, який існує в С++;

<список\_змінних> - перелік ідентифікаторів через кому.

**Наприклад:** int i,j;

float x,y;

***Стандартне введення та виведення*.** Функція **printf()** здійснює виведення за форматом. Синтаксис функції:

*printf(“форматний рядок”, список\_змінних);*

*форматний рядок* складається з:

* символів, які виводяться на екран,
* керуючі символи (\<символ>),
* команди формату( %<код\_формату>).

Функція **scanf()** здійснює введення за форматом. Синтаксис функції:

*scanf(“форматний рядок”,адреса змінної\_1,...,адреса змінної\_n);*

*форматний рядок* складається тільки з команди формату( %<код\_формату>).

**Коди формату**:

%с – символ,

%d – ціле десятичне число,

%f, %F – дійсне число з комою(хх.ххх),

%e, %E – дійсне число в експоненціальній формі(х.ххх е+хх),

%s – рядок символів.

**Керуючі символи:**

\n – новий рядок,

\t – горизонтальна табуляція.

**Структура програми.** Будь – яка програма на Сi складається з функцій. Головна функція має ім’я main().

*# include <stdio.h>*

*// stdio.h містить прототипи функцій scanf(), pintf(),getchar()*

*void main()*

*{ int i; float f;*

*char ss[5];*

*// форматоване введення з консолі цілого числа,*

*//дійсного числа та рядка символів з 5 елементів*

*scanf("%d %f %s",&i,&f,&ss);*

*// форматоване виведення з консолі цілого числа,*

*//дійсного числа та рядка символів з 5 елементів*

*printf("i=%5d f= %8.3f ss=%10s",i,f,ss);*

*// очікування вводу символу з консолі*

*getchar();*

*}*

**1.2 Індивідуальні завдання**

**Завдання 1.**

Розгляньте програму наведену нижче. Поясніть роботу програми. Доробіть її так, щоб вона обчислювала кількість пам’яті в байтах, що займають дані типу long double або long int .

*#include <stdio.h>*

*void main() { printf("Дані типу char займають %d байт. \n", sizeof(char) );*

*printf("Дані типу int займають %d байт. \n", sizeof(int) );*

*printf("Дані типу float займають %d байт. \n", sizeof(float) );*

*}*

**Завдання 2**

Напишіть програму, яка вводить ціле число, а далі виводить його на екран за різними форматами: %d , %4d , %8d , %1d , %-7d .

**Завдання 3**

**Варіант 1**

а) Напишіть програму, яка б виводила на екран Ваше прізвище та ім’я на двох рядках. Колір тексту та фону оберіть за власним бажанням;

б) Напишіть інструкцію виводу значень змінних a, b і c (типу float) з трьома цифрами цілої частини та чотирма – дробової у вигляді:

а = значення b = значення с=значення;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Встановіть точність, ширину поля та зафіксуйте крапку.

**Варіант 2**

а) Напишіть програму, яка б виводила на екран Вашу адресу та телефон на двох рядках. Колір тексту та фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень змінних h та l (типу float), які містять висоту та довжину прямокутника, у форматі з фіксованою крапкою. Перед значенням змінної повинно бути пояснення (висота=, ширина =), а після – одиниці вимірювання (см);

в) Виведіть значення змінних з попереднього пункту у стовпчик. Встановіть точність, ширину поля та зафіксуйте крапку.

**Варіант 3**

а) Напишіть програму, яка б виводила на екран уривок вірша. Колір тексту та фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень трьох змінних типу float, з двома цифрами цілої та трьома дробової частини;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Встановіть точність, ширину поля та зафіксуйте крапку.

**Варіант 4**

а) Напишіть програму, яка б виводила на екран ”дні тижня” в стовпчик, виділяючи кожне слово відповідним кольором;

б) Напишіть інструкцію вводу та виводу значень змінних типу float з фіксованою крапкою (формат задайте самі). При виводі вказують також імена змінних та “=”;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Встановіть точність, ширину поля та зафіксуйте крапку.

**Варіант 5**

а) Напишіть програму, яка б виводила куплет гімну України. Колір тексту та фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень змінних типу float, з різною кількістю цифр цілої та дробової частини;

в) Виведіть числові значення змінних з попереднього пункту у стовпчик. Встановіть точність, ширину поля та зафіксуйте крапку.

**Варіант 6**

а) Напишіть програму, яка б виводила на екран прислів’я на декількох рядках. Колір тексту та фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень змінних типу float, з однаковою кількістю цифр цілої та дробової частин;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Встановіть точність, ширину поля та зафіксуйте крапку.

**Варіант 7**

а) Напишіть програму, яка б виводила на екран у стовпчик типи констант мови Сі. Кожен тип виділіть окремим кольором. Колір фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень трьох беззнакових цілих чисел;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Вирівняйте по лівому полю і встановіть ширину поля.

**Варіант 8**

а) Напишіть програму, яка б виводила на екран у стовпчик лексеми мови Сі. Кожну лексему виділіть окремим кольором. Колір фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень трьох цілих чисел зі знаком;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Вирівняйте по правому полю і встановіть ширину поля.

**Варіант 9**

а) Напишіть програму, яка б виводила на екран у стовпчик структуру програми на мові Сі. Кожен елемент виділіть окремим кольором. Колір фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень сьогоднішнього дня і місяця (у вигляді змінної типу float, наприклад 13.09) та року;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Вирівняйте по лівому полю. Визначіть точність та ширину поля.

**Варіант 10**

а) Напишіть програму, яка б виводила на екран у стовпчик перелік предметів з Вашого розкладу на сьогодні. Кожен предмет позначте окремим кольором. Колір фону оберіть за власним бажанням;

б) Напишіть розклад дзвінків для першої, другої та третьої пари. Час представити як змінну типу float, наприклад: 9.30;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Встановіть точність, зафіксуйте крапку, визначіть ширину поля.

**Варіант 11**

а) Напишіть програму, яка б виводила на екран у стовпчик пори року, кожну з яких позначте відповідним для неї кольором. Колір фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу цін на три види товару у форматі Назва товару – ціна (змінна типу float);

в) Виведіть значення цін з попереднього пункту у стовпчик. Встановіть точність, ширину поля та вирівнювання.

**Варіант 12**

а) Напишіть програму, яка б виводила на екрані перелік спеціальностей ІФНТУНГу . Використайте для цього довільні позначки відповідного кольору. Колір фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу курсів валют у форматі Назва грошової одиниці - курс (змінна типу float);

в) Виведіть значення змінних з попереднього пункту у стовпчик. Вирівняйте по лівому полю, встановіть точність та ширину поля.

**Варіант 13**

а) Напишіть програму, яка б виводила вищі навчальні заклади міста у стовпчик. Використайте для цього позначки відповідного кольору. Колір фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень трьох беззнакових цілих чисел;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Вирівняйте по правому полю і встановіть ширину поля.

**Варіант 14**

а) Напишіть програму, яка б виводила на екран у стовпчик назви інститутів ІФНТУНГу, позначивши кожен певним кольором. Колір фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень чисел 123, 45.080, 2345566.1 типу float з різною кількістю цифр цілої та дробової частини для кожного;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Встановіть точність, ширину поля та зафіксуйте крапку.

**Варіант 15**

а) Напишіть програму, яка б виводила на екран у стовпчик прізвища та імена Ваших одногрупників, виділені різними кольорами. Колір фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень трьох чисел з 4 знаками цілої та 3 дробової частини;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Встановіть точність, ширину поля, зафіксуйте крапку.

**Варіант 16**

а) Напишіть програму, яка б виводила на екран у стовпчик назви знаків Зодіаку, кожна з яких виділена різними кольорами. Колір фону оберіть за власним бажанням;

б) Напишіть інструкцію вводу та виводу значень п’яти х чисел з 3 знаками цілої та 2 дробової частини;

в) Виведіть значення змінних з попереднього пункту у стовпчик. Вирівняйте по лівому полю, встановіть точність та ширину поля.

**Варіант 19**

а) Напишіть програму, яка б виводила на екран назви місяців, що мають 30 днів. Колір фону оберіть за власним бажанням.

б) Напишіть інструкцію вводу та виводу цін на три види фруктів у форматі Назва – ціна (змінна типу float);

в) Виведіть значення змінних з попереднього пункту у вигляді таблиці з заголовком і сіткою.

**Варіант 20**

а) Напишіть програму, яка б виводила на екран вигляд календаря на поточний тиждень з підписами днів тижня та датою, виділених іншим кольором. Колір фону оберіть за власним бажанням;

б) Напишіть програму виведення числа π=3.14159265, і використайте її для трьох значень у різних форматах виведення.

в) Виведіть значення з попереднього пункту у вигляді таблиці з заголовком і сіткою в стовпчик.

**Питання для самоперевірки**

1. Яку структуру має програма на мові С?
2. Як включати коментарі у програму?
3. За якими правилами складається ім’я змінної у програмі на мові С?
4. Які типи даних, що використовуються у мові С, ви знаєте?
5. Як вводяться дані різних типів?
6. Яке призначення керуючих послідовностей, що використовуються в якості аргументів функції форматованого виведення printf()?

**Лабораторна робота №2**

**ПРОСТІ ОБЧИСЛЕННЯ. ОСНОВНІ МАТЕМАТИЧНІ ФУНКЦІЇ**

**СТАНДАРТНОЇ БІБЛІОТЕКИ ІЗ ЗАГОЛОВКОГО ФАЙЛУ <MATH.H>**

**Мета роботи:** узагальнити навички побудови лінійних алгоритмів для розв’язання задач; навчитись використовувати базові типи даних мови Сі при розв’язанні задач; закріпити знання правил побудови виразів та застосування арифметичних операцій і математичних функцій; закріпити знання базових функцій форматного вводу і виводу даних; отримати навички програмної реалізації найпростіших лінійних алгоритмів.

**Теоретичні відомості**

Алгоритм називається лінійним, якщо всі його дії виконуються послідовно, одна за одною, від початку до кінця. У лінійному алгоритмі не може бути команди, яка б передбачала різну послідовність виконання алгоритму. Формально при побудові схем лінійних алгоритмів використовуються три символи: термінатор;  ввід/вивід;  процес.

*Базові типи даних* мови Сі. Перелік типів даних, які програміст може використовувати у своїх програмах залежить від використовуваного середовища розробки

В наступній таблиці приведені характеристики цих та інших числових типів даних залежно у різних середовищах розробки.

Таблиця 2.1 – Типи даних

|  |  |  |  |
| --- | --- | --- | --- |
| Назва | Характеристика | Розмір, біт | Діапазон значень (для найбільшого з розмірів) |
| char | Цілочисельний (символьний) | 8 | 0 ... 255 |
| short | Цілочисельний | 16 | 0 ... 65535 |
| int | Цілочисельний | 16 | – 2147483648 … + 2147483647 |
| bool | Логічне значення значення | 1 | true (істина) або false (неправда) |
| float | Дійсне число одинарної точності | 4 | від ±3.4е-38 до  ±3.4е+38 (7 значущих цифр) |
| double | Дійсне число подвійної точності | 8 | від ±1.7е-308 до  ±1.7е+308 (15 значущих цифр) |
| longdouble | Дійсне число збільшеної точності | 10 | від ±1.2е-4932 до ±1.2е+4932 |

Перед використанням в програмі кожна змінна повинна бути оголошена. Оголошенням називається спеціальна директива, що повідомляє компілятору  про необхідність виділення певного об’ємупам’яті для зберігання даних заданого типу. Крім того, оголошення наділяє змінну певним ідентифікатором або, простіше, – назвою. Ідентифікатором може служити послідовність латинських літер, цифр і знаків підкреслювання будь-якої довжини, яка починається з літери або символу підкреслювання і не співпадає з жодним зарезервованим словом мови програмування. Оголошення змінних здійснюється за наступною (спрощеною) схемою: ; Крапка з комою в кінці оголошення є обов‟язковою. Приклади:

*Intalfa, beta, c;*

*FloatSum;*

Змінна може бути ініціалізована в момент оголошення, для чого використовується операція привласнення – “ = ”, наприклад:

*double e = 2.71;  intEarthRadius = 6357;*

Для оголошення змінної, значення якої в процесі виконання програми повинно залишатись сталим, використовуються так звані константи. Оголошення константи відрізняється від оголошення змінної наявністю модифікатора *const* перед типом константи. Крім того, константа повинна обов’язково бути проініціалізована в момент оголошення. Приклади оголошення констант:

*ConstintEarthRadius = 6357;*

*ConstfloatPi = 3.1415926;*

*Побудова виразів.* Вирази – це конструкції мови програмування, які підлягають обчисленню. В загальному випадку до виразів можуть входити: константи, змінні, виклики функцій, знаки операцій, круглі дужки, що змінюють порядок виконання дій. Вирази слід записувати в один рядок. Обчислення виразів виконується зліва направо; при цьому спочатку виконуються дії, записані в круглих дужках. Порядок виконання операцій визначається їх пріоритетами. Перелік деяких арифметичних операцій мови С в порядку зменшення пріоритетів приведений в наступній таблиці.

Таблиця 2.2 *–* Перелік арифметичних операційй мови Сі

|  |  |  |  |
| --- | --- | --- | --- |
| Операція | Опис | Математичний вираз | Вираз мови Сі |
| ( ) | Виклик функції | sin a | sin(a) |
| ++ | Постфікснийінкремент |  | a++ |
| -- | Постфікснийдекремент |  | a-- |
| ++ | Префікснийінкремент |  | ++a |
| -- | Префікснийдекремент |  | --a |
| – | Унарний мінус | –– a | –a |
| \* | Множення | ab | a \* b |
| / | Ділення | а:в | a / b |
| % | Обчислення залишку | a mod b | a % b |
| + | Додавання | a + b | a + b |
| – | Віднімання | a – b | a – b |
| = | Привласнення | a = b | a = b |

При обчисленні виразів першими виконуються дії, розміщені в круглих дужках. Якщо круглі дужки – вкладені, то вираз, розташований всередині найбільш внутрішньої пари обчислюється першим. Якщо є декілька пар круглих дужок однакового рівня, тобто не вкладених, то їх вміст обчислюється зліва направо.

**Основні математичні функції**. Широкий спектр математичний функцій представлений в модулі math.h. Деякі, найбільш часто використовувані математичні функції цього модуля представлені у наступній таблиці.

Таблиця 2.3 – Математичні функції на Сі

|  |  |  |
| --- | --- | --- |
| Назва функції | Опис | Формат |
| abs | Модуль цілочисельного значення | intabs (int) |
| fabs | Модуль значення з плаваючою точкою | doublefabs (double) |
| pow | Зведення у степінь | doublepow (double, double) |
| sqrt | Квадратний корінь | doublesqrt (double) |
| exp | Експонента (e x ) | doubleexp (double) |
| log | Логарифм натуральний | doublelog (double) |
| log10 | Логарифм десятковий | double log10 (double) |
| asin | Арксинус | doubleasin (double) |
| acos | Арккосинус | doubleacos (double) |
| atan | Арктангенс | doubleatan (double) |
| sin | Синус | doublesin (double) |
| cos | Косинус | doublecos (double) |
| tan | Тангенс | doubletan (double) |

При використанні математичних функцій слід уважно перевіряти, чи входять значення аргументів до області визначення використовуваної функції. Наприклад, аргументом функції *sqrt* не може бути від’ємне число, а другий аргумент (показник степені) функції *pow* при від’ємному першому аргументові (значенні, яке зводиться) не може бути дробовим.

Структура лінійної програми складається з введення аргумента чи аргументів, присвоєння сталим їх числових значень, обчислення значень функції за заданими залежностями і виведення результатів обчислень.

При складанні лінійних програм для прикладів, які наведені в таблиці 1.4, необхідно виконати такий обсяг роботи:

* скласти графічний алгоритм;
* виконати ідентифікацію змінних;
* скласти програму однією або двома мовами, передбачивши введення аргумента і виведення проміжних результатів обчислень, значення аргумента та функції;
* налагодити програму;
* розв’язати задачу на ЕОМ в діалоговому режимі.

### **Завдання**

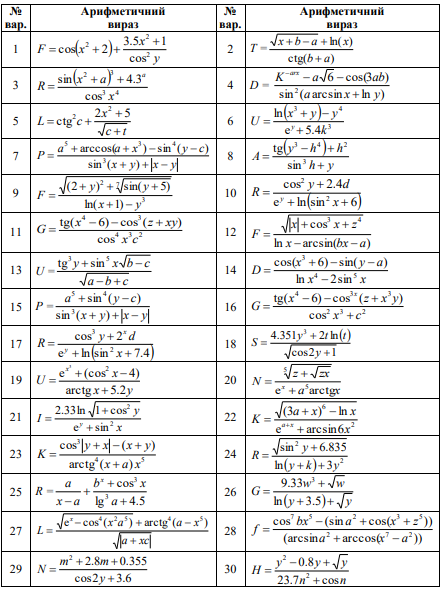
1. У звітіі лабораторної роботи дати відповіді на контрольні питання.
2. У звіті лабораторної роботи записати мовою С арифметичні вирази, наведені в табл. 2.4 … 2.6 відповідно до індивідуального варіанта.
3. У звіті лабораторної роботи скласти схеми алгоритмів і написати програми мовою Сi для розв’язання завдань, поданих в табл. 2.7 … 2.9 відпо- відно до індивідуального варіанта.
4. Створити на комп’ютері програмні проекти в середовищі Сi для реалізації написаних програм. Занести результати обчислень до звіту.

*Таблиця 2.4*

#### Варіанти арифметичних виразів

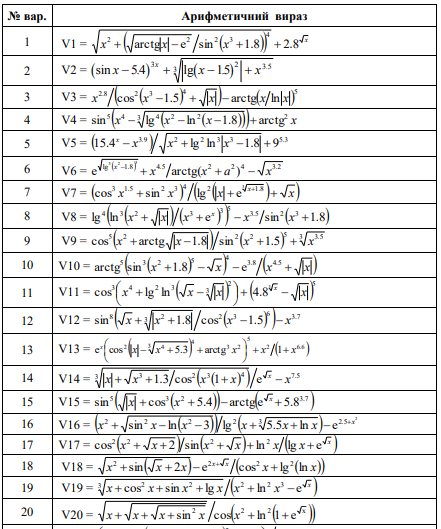
*Таблиця 2.5*

#### Варіанти арифметичних виразів



*Таблиця 2.6*

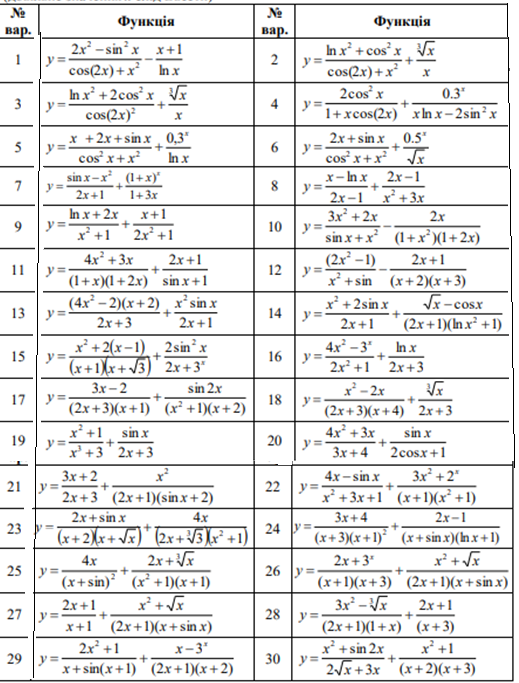
#### Варіанти арифметичних виразів



*Таблиця 2.7*

#### Варіанти завдань для програм з лінійною структурою

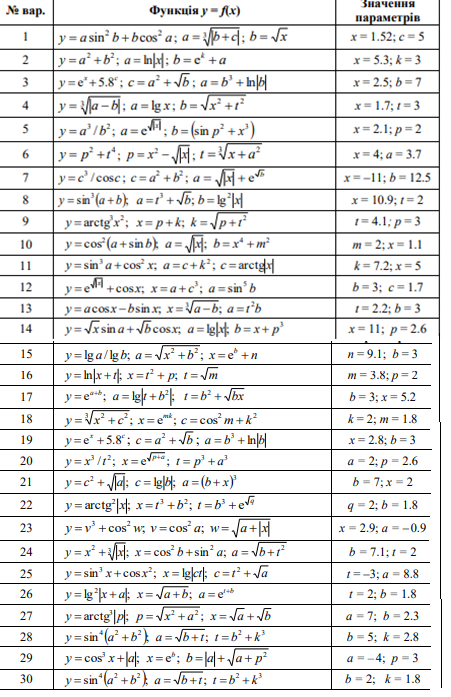
(довільне значення *х* слід Ввести)



*Таблиця 2.8*

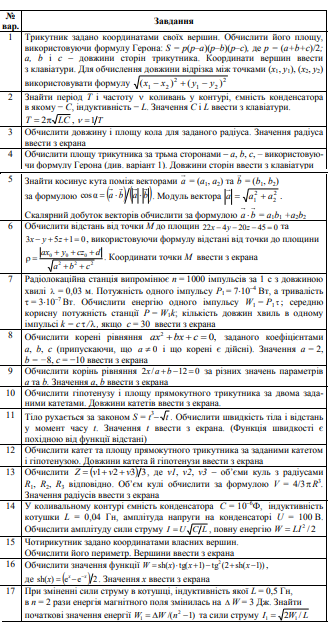
#### Варіанти завдань з лінійною структурою

(перше зі значень параметрів задати як константу, друге – ввести з екрана)



#### *Таблиця 2. 9*

#### Варіанти задач з лінійною структурою

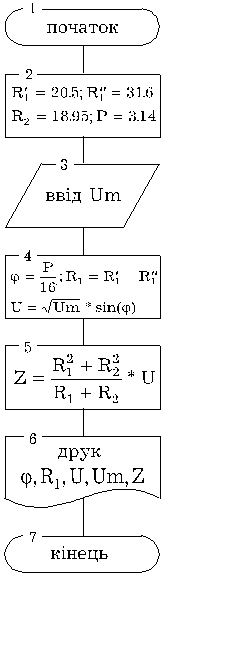


Приклад виконання. Скласти графічний алгоритм і програму для обчислення 

Значення сталих величин: 

Значення аргумента: Um=360.6 .

1. Графічний алгоритм наведений на рис. 2.1.

 Рисунок 2.1 – Графічний алгоритм

2. Ідетинфікація змінних

|  |  |
| --- | --- |
| Змінна | Ідентифікатор |
| Z  R1  R2        Um    U | Z  R1  R2  R3  R4  P  U1  F  U |

4. Програма мовою С

/\*лінійна програма\*/

#include <math.h>

#include <stdio.h>

#define Pi 3.14

main( )

{

float r1,u,f,z,

r2=18.95,

r3=20.5,

r4=31.6,

u1=360.6;

f=Pi/16;

r1=r3+r4;

u=sqrt(u1)\*sin(f);

z=(r1\*r1+r2\*r2)/(r1+r2)\*u;

printf(“\nF=%f R1=%f U=%f \ nZ=%f U1=%f” , f, r1, u, z, u1);

}

F= 0.196250 R1= 52.099998 U= 3.702808

Z= 160.278399 U1= 360.600006

**Питання для самоперевірки**

1. Що таке ідентифікатор? Які правила побудови ідентифікаторів?
2. Які основні оператори введення-виведення даних?
3. Як виконується введення-виведення даних мовою Сі?
4. Який порядок дій в арифметичному виразі?
5. Як задається аргумент стандартної тригонометричної функції?
6. Що таке програма лінійної структури?
7. Якою командою запускається на виконання?
8. Охарактеризуйте основні типи даних?

**Лабораторна робота №3**

**РОЗГАЛУЖЕННІ ПРОГРАМИ**

**Мета роботи**:Набути навики розробки і програмування обчислювальних процесів розгалуженої структури, складати алгоритм розв’язку для поставленої задачі.Оволодіти базовим синтаксисом та основними елементами мови С.  Розвинути практичні навички роботи в середовищі програмування, відлагодження та тестування програм.

**Теоретичні відомості**

**Операції відношення і логічні операції**.

Операції відношення використовуються для порівняння. Це такі операції:

<   менше

<=   менше або рівне

>   більше

>=   більше бо рівне

==   рівне

!=   не рівне

Логічні операції:

&&  i (AND)

||   або (OR)

!   не (NOT)

Ці операції використовуються в умовних виразах (умовах).

Приклади умовних виразів:

 а < 0, 101 >= 105, ‘a’==’A’, ‘a’!=’A’

Кожна умова перевіряється: істинна вона чи хибна (true або false). У мові С немає логічного (Boolean) типу. Тому результат –цілочисельний. True відповідає ненульовій величині, а false – нульовій величині.

Операції відношення і логічні операції мають пріоритет нижчий, ніж арифметичні операції. Це значить, що вираз 12 > 10 + 1 розглядається як вираз       12 > (10+1).

Пріоритет логічних операцій і операцій відношення такий:

|  |  |
| --- | --- |
| вищий | **!** |
|  | **>, <, >=, <=** |
|  | **==, !=** |
|  | && |
| нижчий | **||** |

Тут дужки теж мають найвищий пріоритет

**Умовний оператор if.**

Повна форма цього оператора така:

Повна форма цього оператора така:

**if (умова) оператор;**

**else оператор;**

Якщо значення умови “істинне”, то виконується оператор (блок), що іде за умовою. Якщо умова – “хибна”, то виконується оператор за ключовим словом **else.**Друга частина (**else**) може бути відсутня.

В якості умови може бути будь-який вираз. Перевіряється тільки, чи значення його є ненульовим, чи нульовим.

Часто зустрічається необхідність використовувати конструкцію **if-else-if:**

**if (умова) оператор;**

**else if (умова) оператор;**

**else if(умова) оператор;**

**. . .**

**else оператор;**

Умови перевіряються згори донизу. Як тільки якась умова = “істинна”, виконається “оператор”, а вся інша частина буде проігнорована.

 Щоб перевірити, чи рівне нулю число, треба записати

if (x==0) printf(“Число рівне нулю”);

else printf(“Число не рівне нулю”);

Той самий результат можна отримати так:

if (!x) printf(“Число рівне нулю”);

else printf(“Число не рівне нулю”);

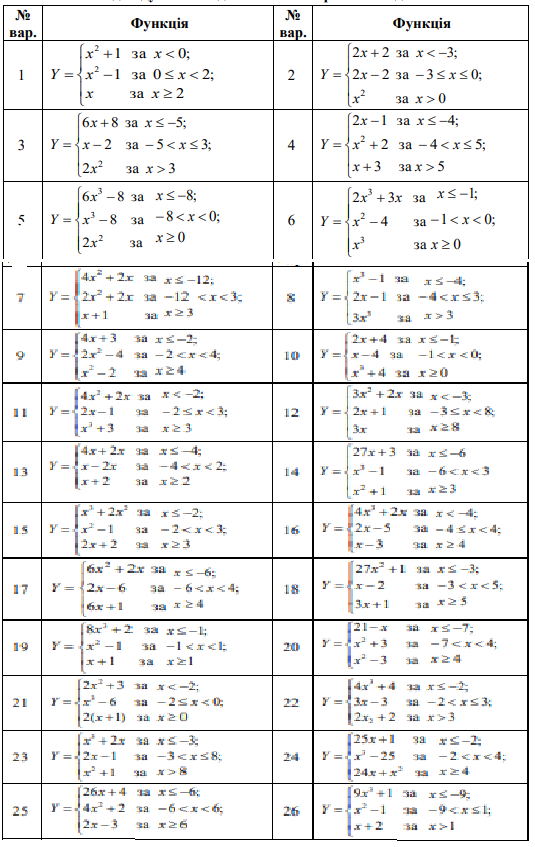
Структура розгалуженої програми складається з введення аргумента, формування умови розгалуження, розрахунку в кожній гілці значення функції відповідно до умови задачі, виведення значення аргумента і обчислення значення функції.

При складанні програм з розгалуженням (таблиця 3.1) необхідно:

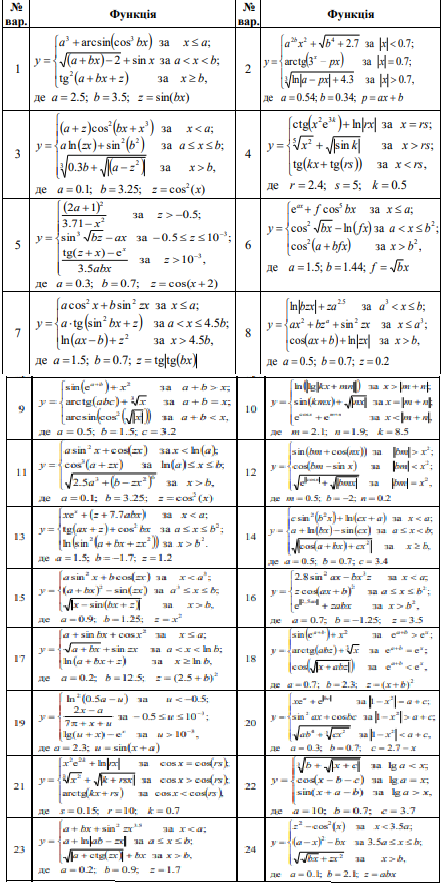
* скласти графічний алгоритм;
* ідентифікувати змінні;
* скласти програму мовою С, в якій передбачити введення аргумента і виведення значення обчисленої функції в кожній вітці алгоритму;
* розв’язати задачу на ЕОМ в діалоговому режимі.

Задачу слід розв’язати для двох значень аргумента, які вказані в умові задачі, що дає можливість тестувати алгоритм.

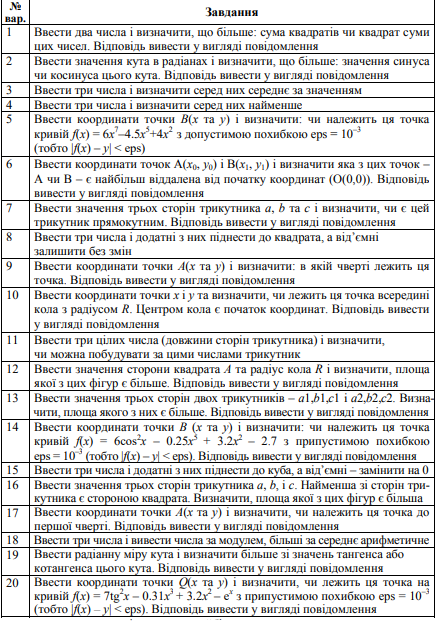
Таблиця 3.1- Індивідуальні завдання базового рівня складності



Таблиця 3.2 – Індивідуальні завдання середнього рівня складності



Таблиця 3.3 – Індивідуальні завдання високого рівня складності



**Завдання 3.4. Знайти значення .**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | Завдання | № | Завдання |
| 1 |  | 2 |  |
| 3 |  | 4 |  |
| 5 |  | 6 |  |
| 7 |  | 8 |  |
| 9 |  | 10 |  |
| 11 |  | 12 |  |
| 13 |  | 14 |  |

Приклад виконання. Скласти графічний алгоритм і програми для обчислення функції



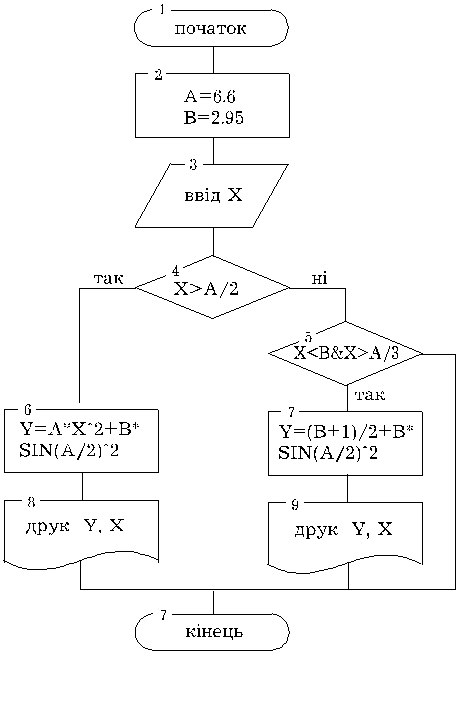
a=6.6; b=2.95.

Задачу розв’язати для двох значень аргумента х=2.46 і х=4.1.

1. Графічний алгоритм показаний на рис.3.1

2. Ідентифікація змінних

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Змінна | a | b | x | Y |
| Ідентифікатор | A | B | X | Y |

Рисунок 3.1– Графічний алгоритм

4. Програма мовою Сі

#include <math.h>

#include <stdio.h>

#define A 6.6

#define B 2.95

main( )

{

float x,y;

scanf(“f”,&x);

if(x>a/2.)

{

y=A\*pow(x,2.)+B\*pow(sin(a/2.).2.);

printf(“\ny=%f x=%f”,y,x);

}

else

if((x<B)&(x>a/3.))

{

y=(B+1)/2.+A\*cos(pow(A.3.));

printf(“\ny=%f x=%f”,y,x);

}

}

Y=2.240538 X=2.460000

Y=111.0194 X=4.100000

Пояснення до програм

Щоб позбутися зайвих операторів при програмуванні розгалужених процесів, необхідно слідкувати за тим, щоб обхід здійснювався по вітці ТАК. Кількість умов у графічному алгоритмі повинна бути на одиницю меншою за кількість гілок.

В приведених програмах обчислення у гілках організовано за допомогою логічних умовних операторів if … else …

#### **Питання для самоперевірки**

1. Як працює умовний оператор?
2. Як працює безумовний оператор за обчисленням?
3. Яка різниця між арифметичним і логічним умовними операторами?
4. Чим визначається кількість умовних операторів у програмі?
5. Які логічні фунуції можуть бути використані в умовних операторах?
6. Чим визначається кількість операторів у тілі умовного оператора?
7. Як уникнути надлишкового застосування безумовного оператора?
8. Як отримуються дані логічного типу?

**Лабораторна робота №4**

**ВИКОРИСТАННЯ КЕРУЮЧИХ КОНСТРУКЦІЙ МОВИ** С **ЦИКЛІЧНІ ПРОГРАМИ**

**Мета роботи:** Оволодіння практичними навиками розробки і програмування обчислювальних процесів циклічної структури і навиками відлагодження та тестування програм в середовищі C.

**Теоретичні відомості**

В мові С існує три види операторів циклів:

For, while, do – while.

**Цикл for** (для, на протязі)

For (<вираз1> ;<вираз2>;<вираз3>)<оператор>;

<вираз1> - використовується для надання початкового значення керуючій змінній, яка тут використовується.

<вираз2> - для перевірки виконання умови продовження циклу.

<вираз3> - для зміни керуючої змінної.

Будь-який з трьох виразів може бути відсутній, але крапка з комою залишається. Якщо відсутні вирази 1 або 3, то керуюча змінна не використовується. Якщо немає виразу 2, то вважається, що він істинний, цикл не закінчується.

**Цикли While і do-While**

Оператор while в загальній формі має вигляд

While (<умова>) <оператор>;

Де оператор може бути простим, складеним або порожнім.

Цикл виконується доти, доки умова виконується. Якщо умова не виконується, програма переходить до наступного оператора. Спочатку перевіряється умова, а потім виконується оператор.

В циклі **do – while** умова перевіряється в кінці. Він має вигляд:

Do {

< послідовність операторів >

}

while (<умова>);

Фігурні дужки не обов’язкові, якщо оператор тільки один.

**Цикл For**

Стандартний запис цикла for

for (int i = 0; i < 10; i++)

printf("%d\n", i);

В об’явлені циклу записується: змінна циклу, її початкове значення та діапазон. В прикладі будуть виводитись числа від 0 до 10.

Якщо в тілі циклу всього один рядок коду, тоді фігурні дужки можна пропустити і не записувати їх.

**Цикл While**

цикл while відрізняється від цикла for способм запису. В while необхідно прописати умову, а решта параметрів записуються за межами циклу:

int i = 1; // Создание переменной

while (i <= 10) { // Здесь только условие

printf("%d\n", i);

i++; // Увеличение переменной

}

**Цикл do** дуже схожий до while циклу за виключенням того, що цикл do виконує код, що знаходиться в тілі циклу принаймні раз. В циклі while, якщо умова хибна відразу, то тіло циклу ніколи не буде виконано.

Ви можете використати цикл do якщо код буде , наприклад, виконуватись у відповідь на введення даних користувачем.

do

{

      // Обробка даних.

      response = Сonsole.ReadLine();

} while (response != "Кінець");

**Оператори для работи в циклах**

Є 2 основных оператора для работы в циклах:

Оператор *break* - використовується для виходу із циклу повністю;

Оператор continue - пропускає лише одну ітерацію і не виходить із циклу.

Цикл типова структура, яка характерна для багатьох програм. Цикл складається із заголовка і тіла циклу. Заголовок циклу мовою Сі описується оператором циклу FOR. В цьому операторі задається початкове значення параметра циклу, крок його зміни і перевірка досягнення параметром циклу кінцевого значення. Тіло циклу являє собою послідовність операторів, які виконуються в циклі.

Цикли також можуть бути побудовані за допомогою операторів умовного і безумовного переходу, так звані цикли з передумовою і післяумовою.

Для завдань, які описані в таблиці 3, необхідно:

- скласти графічний алгоритм, використавши оператор циклу або умовний оператор;

- провести ідентифікацію змінних;

- скласти програми з використанням оператора циклу і умовного оператора;

- розв’язати задачу на ЕОМ в діалоговому режимі;

- провести аналіз результатів.

Якщо за умовою задачі не заданий один з параметрів циклу (початкове, кінцеве значення параметра або крок зміни), то його необхідно визначити, користуючись залежністю

 (4.1)

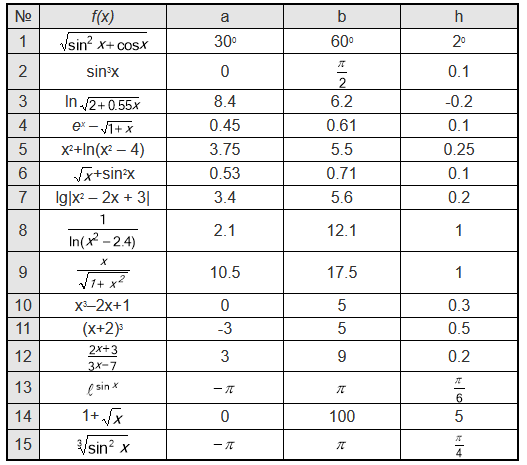
де N - кількість циклів;хп,хк - початкове і кінцеве значення параметра циклу;

- крок зміни параметра циклу.

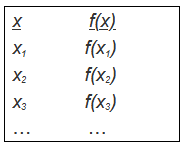
## **Індивідуальні завдання**

**Завдання 1.**

1. Протабулювати функцію f(x) на відрізку [a,b] з кроком h.



Результати вивести на екран у вигляді:



**Завдання 2.**

|  |  |
| --- | --- |
| 1 | Перевірити справедливість рівності при заданій точності : |
| 2 | Перевірити справедливість рівності при заданій точності : |
| 3 | Перевірити справедливість рівності при заданій точності : |
| 4 | Перевірити справедливість рівності при заданій точності : |
| 5 | Перевірити справедливість рівності при заданій точності : |
| 6 | Перевірити справедливість рівності при заданій точності : |
| 7 | Перевірити справедливість рівності при заданій точності : |
| 8 | Перевірити справедливість рівності при заданій точності : |
| 9 | Перевірити справедливість рівності при заданій точності : |
| 10 | Перевірити справедливість рівності при заданій точності : |
| 11 | Перевірити справедливість рівності при заданій точності : |
| 12 | Перевірити справедливість рівності при заданій точності : |
| 13 | Перевірити справедливість рівності при заданій точності : |
| 14 | Перевірити справедливість рівності при заданій точності : |

**Завдання 3.**

|  |  |
| --- | --- |
|  | Нехай , де . Визначити . |
|  | Нехай , де . Визначити . |
|  | Нехай , де . Визначити . |
|  | Нехай , де . Визначити . |
|  | Нехай , . Визначити . |
|  | Нехай , . Визначити . |
|  | Нехай , де . Визначити . |
|  | Нехай ,. Визначити . |
|  | Нехай ,. Визначити . |
|  | Нехай , . Визначити . |
|  | Нехай , . Визначити . |
|  | Нехай , . Визначити . |
|  | Нехай , де . Визначити . |
|  | Нехай , де . Визначити . |

**Завдання 4**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № ва- |  | Константи | | Аргу- | Почат- кове знач. | Кінце- ве зна-чення | Крок зміни Пара- | Кіль- кість інтер- |
| рі- ан-ту | Функція | a | b | мент | пара-метра циклу | пара- метра циклу | Метра Циклу | валів пара- метра |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  | 21.4 | 1.95 | x | 4.6 | - | 1.5 | 8 |
| 2 |  | 121.2 | 3.8 | q | 1.4 | 1.85 | - | 9 |
| 3 |  | 8.3 | 1.43 | p | - | -4.74 | 0.15 | 10 |
| 4 |  | 1.6 | 2.09 | m | - | 16 | 1.5 | 7 |
| 5 |  | 0.83 | - | x | 1.15 | - | 0.35 | 11 |
| 6 |  | 3.85 | 1.8 |  | - | 150 | 70 | 8 |
| 7 |  | 1.6 | 14.3 | t | 2.75 | 5.0 | - | 9 |
| 8 |  | - | 6.215 |  | 400 | 540 | - | 7 |
| 9 |  | 2.43 | - | y | 1.62 | - | 0.15 | 8 |
| 10 |  | 1.85 | 2.63 |  | -3.45 | - | -1.6 | 11 |
| 11 |  | - | 3.85 | d | -2.3 | 0.01 | - | 7 |
| 12 |  | 2.65 | 1.48 | x | 0.75 | 0.012 | - | 9 |
| 13 |  | 3.2 | - | y | -4.8 | 0.1 | - | 7 |
| 14 |  | - | 8.45 | z | - | 0.24 | 3.2 | 9 |
| 15 |  | -16.3 | - | x | -16.3 | - | -9.3 | 9 |
| 16 |  | -8.6 | 3.28 | d | 3.6 | - | 4.0 | 14 |
| 17 |  | 1.42 | 0.85 | z | 1.6 | 0.35 | - | 5 |
| 18 |  | 3.2 | - | f | -4.8 | 0.1 | - | 7 |
| 19 |  | 8.9 | - |  | 350 | - | 500 | 6 |
| 20 |  | 1.2 | - | z | 2.6 | - | 3.8 | 8 |
| 21 |  | - | 6.15 | x | - | 0.2 | 1.65 | 10 |
| 22 |  | 3.26 | - | x | - | 0.18 | 3.41 | 7 |
| 23 |  | 1.85 | 6.21 | g | 22 | 3.1 | - | 9 |
| 24 |  | 4.55 | 7.53 | x | 4.8 | 0.25 | - | 5 |
| 25 |  | 6.92 | - | f | 0.6 | - | 1.4 | 8 |

Приклад виконання. Скласти графічний алгоритм і прграми обчислень таблиці значень функції y=bx2+ex-1, b=11.83. Аргумент х змінюється від початкового значення х=2.6 з постійним кроком 0.15, кількість значень функції n=8.

Графічний алгоритм і програми скласти з використанням оператора циклу та умовного оператора.

Розв’язок задачі.

1. Визначимо кінцеве значення аргумента х:

, звідки 

1. Графічний алгоритм з використанням оператора циклу показаний на рис. 4.1, з використанням умовного оператора на рис. 4.2.

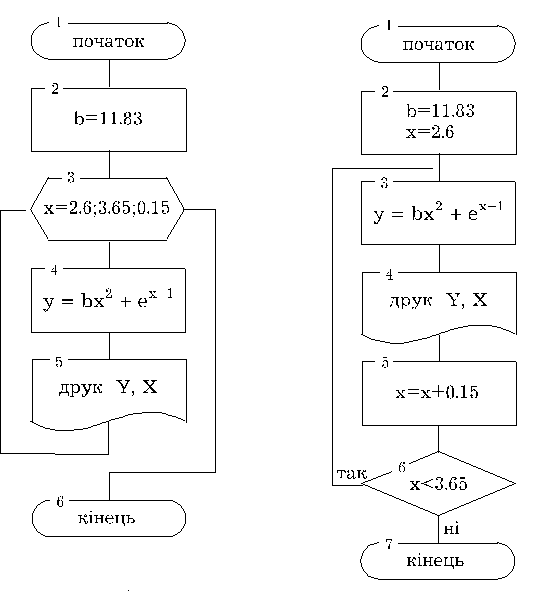


Рисунок 4.1

Рисунок 4.2

Програма

#include <stdio.h>

#include <math.h>

#define x1 2.6

#define x2 3.66

#define dx 1.5e-1

#define b 11.83

main( )

{

float x,y;

for(x=x1;x<x2;x+=dx)

{

y=b\*x\*x+exp(x-1);

printf(“\ny=%f x=%f,y,x);

}

}

Y=84.923828 X=2.600000

Y=95.218979 X=2.750000

Y=106.176201 X=2.900000

Y=117.816490 X=3.050000

Y=130.164230 X=3.200000

Y=143.247772 X=3.350000

Y=157.100037 X=3.500000

Y=171.759277 X=3.650001

Пояснення до програм

В графічному алгоритмі (рис.4.1) в символі 3 формується заголовок циклу, де вказано початкове, кінцеве значення параметра циклу і крок його зміни, блоки 4 і 5 являють собою тіло циклу. На рис.4.2 наведено алгоритм циклічної програми із застосуванням логічного оператора (блок 6), де значення параметра циклу зрівнюється з кінцевим. В блоці 3 задається початкове значення параметра циклу, а в блоці 5 нарощується параметр циклу на величину кроку.

В прикладі наведено дві блок-схеми, одна з них - із застосуванням оператора циклу for , що відповідає алгоритму, зображеному на рис.4.1, друга - з використанням умовного оператора для циклу з післяумовою, що відповідає алгоритму, показаному на рис.4.2.

**Питання для самоперевірки**

1. Що таке цикл? Складові частини циклу?
2. Чому в циклічних процесах з регулярною зміною аргумента мовою Сі деколи необхідно вводити додаткову керуючу змінну?
3. Які дані необхідні для організації циклу?
4. В яких межах змінюється параметр циклу в приведених програмах?
5. Яка різниця в алгоритмах циклів, побудованих за допомогою оператора циклу і за допомогою умовних операторів?
6. Який оператор в наведених програмах контролює умову виходу з циклу?
7. Дайте зрівняльну характеристику програм з використанням умовного оператора і оператора циклу?
8. Які відмінності в організації циклу з післяумовою і передумовою.

**Лабораторна робота №5**

**ВИКОРИСТАННЯ КЕРУЮЧИХ КОНСТРУКЦІЙ МОВИ** **С. ЦИКЛІЧНІ ПРОГРАМИ, ЩО МІСТЯТЬ РОЗГАЛУЖЕННЯ**

**Мета**: навчитися: складати програми мовою C з регулярною змінною аргумента, які містять розгалуженням;описувати та використовувати змінні для розгалужень і циклів; програмно змінювати значення властивостей об’єктів;перевіряти результат виконання програми на відповідність умові задачі.

Цикл, що містить розгалуження часто використовується в програмуванні. Він характеризується тим, що використання розгалуження всередині циклу досить поширене і може бути корисним для керування поведінкою циклу в залежності від певних умов.

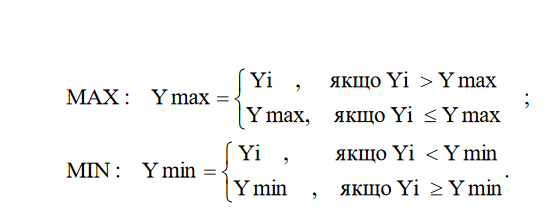
**Знаходження МІN та МАХ** значення функції припускає, що функція визначена на заданому інтервалі та необхідно визначити найбільше (МАХ) чи найменше (МІN) серед них. Оскільки аналітичний вираз функції незмінний на інтервалі визначення, то різні її значення утворюються за рахунок зміни параметрів

(аргументів) цієї функції. Тому процедура визначення МІN та МАХ значення функції виконується у циклі. Перед початком циклу з визначення МІN чи МАХ значення функції Y=f(x) обчислюється перше значення Y1 функції, яке приймається за МІN чи МАХ, тобто Ymin=Y1 чи Ymax=Y1. Далі для обчислення наступного значення функції змінюємо значення параметра циклу та, якщо нове значення параметра припустиме, обчислюємо відповідне значення функції. Отримане значення функції Y2, в залежності від умови задачі, порівнюємо при

знаходженні МІN з Ymin, при знаходженні МАХ з Ymax. Якщо Y2< Ymin, то Ymin набуває значення Y2 (Ymin=Y2), а якщо Y2>=Ymin, то Ymin залишає своє значення, після чого здійснюється перехід до обчислення наступного значення

параметра. Після зміни параметра циклу процес повторюється. При знаходженні MAX, Y2 порівнюється з Ymax. Якщо Y2>MAX, то Ymax набуває значення Y2 (Ymax=Y2), інакше Ymax зберігає своє значення і після зміни параметра циклу.

процес повторюється.Все це можна описати математичною залежністю



де Yi – наступне значення функції. Вихід з циклу здійснюється після досягнення параметром верхньої межі інтервалу.

**Для завдань, які описані в таблиці 3, необхідно**:

- скласти графічний алгоритм, використавши оператор циклу або умовний оператор;

- провести ідентифікацію змінних;

- скласти програми з використанням оператора циклу і умовного оператора;

- розв’язати задачу на ЕОМ в діалоговому режимі;

- провести аналіз результатів.

**Варіанти індивідуальних завдань**

**Завдання 1.**

1. Обчислити значення функції



у набуває значення від 0 до 8 з кроком 0.15.

2. Обчислити значення функції

,

де =2; 2.4; 2.8; 3.2; 3.6; 4.0; q=0.1; 0.3; 1.1; 1.25; 1.9; 2.8.

Величини  і q змінюються одночасно.

3. Знайти найменше значення функції і відповідне їй значення аргумента:

Аргумент змінюється від початкового значення 0.3 з кроком 0.25 до кінцевого значення 3.3 (радіан).

4. Вивести на друк тільки від’ємні значення і їх кількість:

Аргумент х змінюється від початкового значення 1.1 з кроком 0.2 до кінцевого 4.1.

5. Знайти найменше додатнє і найменше від’ємне значення функції і відплвідне їм значення аргумента:

Аргумент х змінюється від початкового значення 0.1 з кроком 0.15 до кінцевого 3.1 (радіан).

6. Обчислити різницю між максимальним і мінімальним значенням функції:

 = 0.28.

Аргумент х змінюється від початкового значення 0.1 зкроком 0.32 до кінцевого значення 6.5.

7. Знайти квадрат максимального значення функції:

 b = 4.85.

Аргумент х змінюється від початкового значення 1/16 з кроком 0.25 до кінцевого 4.2.

8. Обчислити вС значення функції і аргумента:

Аргумент b змінюється від початкового значення 9.5 з кроком 0.25 до кінцевого 12.75.

9. Обчислити значення функції і вивести на друк окремо відємні, а окремо додатні значення функції і відповідні значення їм аргументів:

Аргумент х змінюється від початкового значення 1.2 з кроком 0.15 до кінцевого 4.4.

10. Вивести на друк значення функції, які задовольняють умову у<2.5 i y>1.5:

Аргумент а змінюється від початкового значення 1.15 до кінцевого 6.3. Кількість значень аргумента n=9.

11. Вивести на друк значення, які задовольняють умову у>1.35 i y<2.4, і відповідні значення аргумента. Аргумент х змінюється від 0.42 до 0.86 з кроком 0.04:

12. Обчислити добуток мінімального і максимального значень функції. Кількість значень аргумента n=19. Аргумент y змінюється від початкового значення 0.1 з кроком 0.75 (радіан):

13. Вивести на друк значення функції, які лежать в межах

a>b>c, підрахувати їх кількість.

Аргумент х змінюється від початкового значення 3.4 з кроком 0.11 до кінцевого 6.6.

14. Обчислити 20 значень функції і відповідних значень аргумента, а також окремо вивести на друк максимальне зна-чення функції:

Аргумент с змінюється від початкового значення 1.35 до кінцево-го 4.35 з постійним кроком.

15. Вивести на друк ті значення функції у і їх кількість, які лежать в межах 2.5>y>-1.5:

Аргумент с змінюється від початкового значення 4.2 до 8.3 з кроком 0.15.

16. Обчислити члени послідовності



які задовольняють умову 1.5<a<3.8; m=1,2,3,...,16.

17. Обчислити таблицю значень функції

для значень х, які змінюються від 0.5 до 8.5 включно з кроком 0.05. При цьому, якщо знаменник менше 10-3 за абсолютним зна-ченням, то покласти у=106. Коефіцієнти а і b ввести з клавіатури.

18. Скласти блок-схему алгоритму і програму для розв’язу-вання квадратного рівняння



Якщо дискримінант додатній, то вивести на друк дійсні корені. Якщо дискримінант від’ємний, то вивести окремо дійсну і уявну частини коренів. Введення чисел a, b, q виконати з клавіатури.

19. За відомими декартовими координатами х і у довільної точки на площині визначити її полярні координати і . При цьому



а полярний кут

20. Обчислити пари функцій f1 i f2, якщо межі зміни аргументів однакові. На друк вивести значення функції і аргумента.

Якщо аргумент лежить за вказаними межами, то вивести на друк значення аргумента і повідомлення “ФУНКЦІЯ НЕ ЗНАЙДЕНА”.

Числові значення аргумента ввести з клавіатури.

21. Температуру з градуСв за шкалою ЦельСя (С) в градуси за шкалою Фаренгейта (F) переводять за залежністю .

Перевести температури від 12СдоС з кроком 1С.

22. Зміна потужності випромінювання ізотопів Q в чаС t описується залежністю



де - початкова потужність; - період піврозпаду.

Визначити, через скільки діб потужність випромінювання ізотопу, період піврозпаду якого дорівнює 8 діб, знизиться до безпечної величини 0.15 рентгена на годину,  рентгена на годину.

23. Тиск рідини з глибиною зростає. Надрукувати таблицю зміни тиску глинистого розчину в середині через кожні 150 м.

Глибина свердловини - 1800 м, густина розчину =1300 кг/м.

24. За перший рік розробки нафтового родовища видобуток нафти склав 200 тис. т на рік. В наступні роки видобуток нафти зростав на 8% за рік. Визначити, скільки буде видобуто нафти з родовища за 25 років.

25. Для умови задачі 4.24 обчислити, через який найменший строк видобуток нафти складатиме не менше 350 тис. т нафти за рік.

26. Концентрація хлорного вапна в басейні об’ємом V складає С0. В басейн вливають чисту воду при постійному в ньому рівні /приплив дорівнює витоку/. З умови ідеального перемішування концентрація хлорного вапна змінюється за законом



де t - час; С0 - початкова концентрація.

Визначити, через який час концентрація хлорного вапна досягне безпечної для людини велечини 0.1 г/л, Q=120 м/год,

V=25 м3, С0=12 г/л.

**Завдання 2.**

1. Скласти програму для обчислення значень функції на проміжку *[a, b]* з кроком *h*. У програмі передбачити обчислення величин, вказаних у варіантах завдань.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Функція** | **a** | **b** | **h** | **Величини, які необхідно обчислити** |
| 1. |  | -6 | 6 | 1.2 | середнє всіх у |
| 2. |  | 1.5 | 5 | 0.5 | кількість у, коли у<0 |
| 3. |  | 1 | 5 | 0.5 | min значення у, на проміжку у>4 |
| 4. |  | 2 | 7.5 | 0.5 | середнє арифметичне всіх значень у <0 |
| 5. |  | -5 | 5 | 1 | max значення у, якщо у>0 |
| 6. |  | -4 | 4 | 0.8 | кількість у, коли у<0 |
| 7. |  | -10 | -3 | 0.7 | max значення у, на проміжку 2<у<50 |
| 8. |  | -4 | 2 | 0.5 | кількість у, якщо у>0.5 |
| 9. |  | -3 | 3.5 | 0.5 | Середнє значення y, якщо у>4 |
| 10. |  | -6 | 2 | 0.8 | кількість у, якщо у<0 |
| 11. |  | -3.14 | 3.14 | 0.628 | Парні y і їх кількість, якщо у<0 |
| 12. |  | -5 | 5 | 1 | кількість у, якщо у>0 і у<0 |
| 13. |  | -4 | 4 | 0.8 | кількість, якщо у>4 |
| 14. |  | -3 | 3 | 0.6 | кількість у, якщо у>0 і у<0 |
| 15. |  | -2 | 4 | 0.6 | кількість у, якщо у>0.5 |
| 16.. |  | -10 | 0 | 1 | Min на проміжку у<4 і кількість у>4 |
| 17. |  | 1 | 7.6 | 0.6 | Min значення у, для у>10 |
| 18. |  | -4 | 4 | 0.8 | кількість у, якщо у>0 і у<0 |
| 19. |  | -5 | 1 | 0.6 | середнє, якщо у>0 і у<0 |
| 20. |  | 2 | 7 | 0.5 | Кількість парних у, якщо у>0 |
| 21. |  | 5 | 12 | 0.5 | кількість непарних у, якщо у<0 |
| 22. |  | -2 | 2 | 0.4 | Всі непарні у, якщо 0< у<10 |
| 23. |  | -6 | 4 | 1 | середнє арифметичне всіх у |
| 24. |  | 3 | 7 | 0.4 | Всі парні у, якщо у>4 |
| 25. |  | -2 | 2 | 0.4 | кількість у, 0<у<1 |
| 26. |  | -3.5 | 3.5 | 0.5 | кількість у, якщо 0< у<0.5 |
| 27. |  | 2 | 8 | 0.6 | Різницю між min та max |
| 28. |  | 1.2 | 3.5 | 0.2 | середнє арифметичне значень у |
| 29. |  | 1 | 4 | 0.3 | кількість парних у, для у<0 |
| 30. |  | -5 | 2 | 0.7 | кількість у, коли 0< у<10 |

**Завдання 3.**

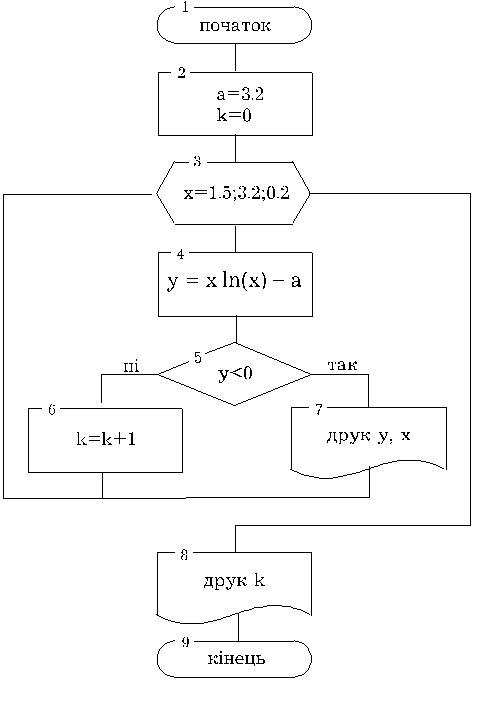
|  |  |
| --- | --- |
| 1 | За даним проміжком дійсних чисел x є [a .. b ] , y=2 обчислити  ,  де |
| 2 | За даним даним проміжком дійсних чисел x є [a .. b ], y=2.5 обчислити  )),  де |
| 3 | За даним проміжком дійсних чисел x є [a .. b ] обчислити  ,  де |
| 4 | За даним проміжком дійсних чисел x є [a .. b ] обчислити  ,  де |
| 5 | За даним проміжком дійсних чисел x є [a .. b ] обчислити  ,  де |
| 6 | За даними проміжком дійсних чисел x є [a .. b ] обчислити  ,  де |
| 7 | За даним проміжком дійсних чисел x є [a .. b ] , y=3.89, z=9.45 Обчислити  . |
| 8 | За даним проміжком дійсних чисел x є [a .. b ], z= 3.96, y=-1.2. Обчислити  . |
| 9 | За даним проміжком дійсних чисел x є [a .. b ] обчислити:  . |
| 10 | За даним проміжком дійсних чисел x є [a .. b ], y=1.2, z=0.25 Обчислити значення виразу , де |
| 11 | За даним проміжком дійсних чисел x є [a .. b ] Обчислити значення виразу  ,  де |
| 12 | За даним проміжком дійсних чисел x є [a .. b ] , y=2 обчислити  ,  де |
| 13 | За даним проміжком дійсних чисел x є [a .. b ] , y=3.89, z=9.45 Обчислити  . |
| 14 | За даним проміжком дійсних чисел x є [a .. b ] , y=2 обчислити  ,  де |

Приклад виконання. Вивести на друк тільки від’ємні значення функції і відповідні значення аргумента, а також кількість додатних зна-чень функції

Аргумент х змінюється від початкового значення 1.5 до кінцево-го 3.3 з кроком 0.2.

1. Графічний алгоритм наведений на рис.5.1

Рисунок 5.1 – Графічний алгоритм

2 Ідентифікація змінних

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Змінна | a | b | x | y |
| Ідентифікатор | A | B | X | Y |

Тут к - кількість додатних значень

4.Програма мовою С

#include <stdio.h>

#include <math.h>

#define A 3.2

#define X1 1.5

#define X2 3.31

#define DX 2e-1

main( )

{

float x,y;

int k=0;

for(x=X1;x<=X2;x+=DX)

{

y=x\*log(x)-A;

if (y<0)

printf(“\n Y=%f X=%2.if”,y,x);

else

k++;

}

printf(“\n КІЛЬКІСТЬ ДОДАТН. ЗНАЧ. ФУНКЦ.=%i”,k);

}

Y= -2.591802 X=1.5

Y= -2.297932 X=1.7

Y= -1.980477 X=1.9

Y= -1.641931 X=2.1

Y= -1.284309 X=2.3

Y= -0.909273 X=2.5

Y= -0.518220 X=2.7

Y= -0.112338 X=2.9

КІЛЬКІСТЬ ДОДАТН. ЗНАЧ. ФУНКЦ.= 2

Пояснення до програм

Графічний алгоритм (рис.5.1) складається з циклу, блок 3 - це заголовок циклу і розгалуження. В блоці 5 порівнюється значення функції у з нулем. Якщо значення у від’ємне, то друкується значення функції і відповідне значення аргумента, блок 7. Якщо значення у невід’ємне, то вмістиме лічильника циклу зростає на одиницю, блок 6.

Програма має циклічну структуру. В тілі циклу знаходиться розгалуження, яке реалізовано за допомогою умовного оператора. Цикл організований за допомогою оператора for. На друк виведено всі від’ємні значенння функції і вказано, що два значення функції є додатні.

**Питання для самоперевірки**

1. Напишіть програму без застосування оператора циклу.
2. Чи може розгалуження не входити до тіла циклу?
3. Як побудувати програму мовою С за допомогою неповного умовного оператора, використовуючи цикл з післяумовою чи передумовою?
4. Побудуйте програму з використання циклічних структур з передумовою та післяумовою.
5. Змініть програми таким чином, щоб перед виведенням результатів обчислень було надруковано повідомлення “РЕЗУЛЬТАТИ ОБЧИСЛЕННЯ”.

**Лаборторна робота №6**

**МАСИВ ЯК СТРУКТУРОВАНИЙ ТИП. ОДНОВИМІРНІ МАСИВИ**

**Мета роботи**: вивчити складений тип даних – масив. Написати і відлагодити програму з використанням одновимірних масивів

**Теоретична частина**

Масив - кінцева послідовність даних одного типу.

Масив - об'єкт складного типу, кожен елемент якого визначається ім'ям (*ID*) і цілочисельним значенням індексу (номери), по якому до елементу масиву проводиться доступ. Розглянемо одновимірні масиви.

***Увага***! ***Індекси масивів в мові С починаються з 0.***

У програмі одновимірний масив декларується таким чином:

*тип* *ID масиву* [*розмір*];

де *розмір* - вказує кількість елементів в масиві. Розмір масиву може задаватися константою або константним вираженням. Для використання масивів змінного розміру існує окремий механізм - динамічне виділення пам'яті.

Приклади декларації масивів :

int ***a***[5];

double ***b***[4] = 1.5, 2.5, 3.75;

у цілочисельному масиві ***а*** перший елемент ***а***[0], другий - ***а***[1], ., п'ятий - ***а***[4]. Для масиву ***b***, що складається з дійсних чисел, виконана ініціалізація, причому елементи масиву отримають наступні значення: ***b***[0]=1.5, ***b***[1]=2.5, ***b***[2]=3.75, ***b***[3]=0.

У мові C не перевіряється вихід індексу за межі масиву. Коректність використання індексів елементів масиву повинен контролювати програміст.

Приклади опису масивів :

const Nmax=10; - завдання максимального значення;

***typedef*** double mas1[Nmax\*2]; - опис типу одновимірного масиву;

mas1 *a*; - декларація масиву *а* типу mas1;

int *ss*[10]; - масив з десяти цілих чисел.

Елементи масивів можуть використовуватися у виразах так само, як і звичайні змінні, наприклад:

*f* = 2\**a*[3] + *a*[*Ss*[*i*] + 1]\*3;

*a*[*n*] = 1 + sqrt(fabs(*a*[*n* - 1]));

Приклад. Елементам масиву tod присвоюються значення від 0 до 55.

     #include <stdio.h>

     int main(void)

     {

      int tod[55];

      int i;

      for(i=0; i<55; i++)

           tod[i] = i;

      for(i=0; i<55; i++)

           printf("%d ", tod[i]);

      return 0;

     }

Для задач цього розділу необхідно скласти графічний алгоритм, провести ідентифікацію змінних, скласти прграму і розв’язати задачу на ЕОМ.

**Варіанти завдань.**

**Завдання 1.**

|  |  |
| --- | --- |
|  | Побудувати масив *А=*(*ai*), елементи якого задаються формулою:    де *x, y* – вводяться з клавіатури. Вивести кількість елементів масиву *А*, які більші за задане число *z*. |
|  | Задано масив , де  Знайти добуток елементів масиву *В* з непарними номерами. |
|  | Побудувати масив *А=*(*ai*), елементи якого задаються формулою:    Обчислити суму додатних елементів масиву *А.* |
|  | Елементи масиву задаються так:  .  Обчислити |
|  | Елементи масиву  задаються так: . Побудувати масив  так: |
|  | Елементи масиву  задаються так:  .  Знайти |
|  | Елементи масиву  задаються так:  .  Знайти середнє арифметичне всіх елементів масиву, які потрапляють у про­мі­жок . |
|  | Побудувати масив *А=*(*ai*), елементи якого задаються формулою:    де *х* вводиться з клавіатури. Знайти порядковий номер найменшого елемента ма­сиву *А.* |
|  | Побудувати масив *А=*(*ai*), елементи якого задаються формулою:    Обчислити суму додатних елементів масиву *А* і вивести її на форму. |
|  | Побудувати масив *А=*(*ai*), елементи якого задаються формулою:    Знайти найбільший елемент масиву *А.* |
|  | Побудувати масив *А=*(*ai*), елементи якого задаються формулою:    Вивести добуток елементів масиву *А* з парними індексами. |
|  | Побудувати масив *Х=*, елементи якого задаються формулою:    де *a*, *b* – вводяться з клавіатури. Знайти середнє арифметичне значення від’єм­них елемен­тів масиву *Х.* |
|  | Побудувати масив *V=*(*vi*), елементи якого задаються формулою:    де *a, b, х* – вводяться з клавіатури. Обчислити добуток елементів масиву *V* з парними індексами |
|  | Побудувати масив *А=*(*ai*), елементи якого задаються формулою:    де *b* – вводиться з клавіатури. Вивести суму елементів масиву *А* з непарними ін­дексами. |

**Завдання 2.**

|  |  |
| --- | --- |
|  | Дано два вектори . Знайти суму векторів. |
|  | Дано вектор  і число . Знайти добуток вектора на число. |
|  | Дано два вектори . Знайти скалярний добуток векторів. |
|  | Дано два вектори . Знайти векторний добуток векторів. |
|  | Дано два вектори . Знайти косинус кута між ними. |
|  | Дано два вектори .З’ясувати, чи паралельні вони. |
|  | Дано два вектори .З’ясувати, чи є вони перпендикулярними. |
|  | Обчислити значення виразу s=2<a,b>-3<a,c>, де a,b,c, <x,y> – скалярний добуток векторів. |
|  | Ззнайти вектор  c=a-3\*b+2\*c, де a,b,c. |
|  | Ззнайти вектор  c=2(a +c)- b, де a,b,c. |
|  | Ззнайти вектор  c=2<a,b>с-3\*b, де a,b,c, <x,y> – скалярний добуток векторів. |
|  | Ззнайти вектор  c=2(a +c)- b, де a,b,c. |
|  | Ззнайти вектор  c=2(a +c)-3(a-b), де a,b,c. |
|  | Обчислити значення виразу  s=2<a+b, a - b>, де a,b, <x,y> – скалярний добуток векторів.. |

**Завдання 3.**

1. Впорядкувати елементи масиву за зростанням.
2. Впорядкувати елементи масиву за спаданням.
3. Впорядкувати елементи масиву таким чином, щоб спочатку були розміщені всі додатні елементи, а потім всі від’ємні(0 вважаємо додатнім елементом).
4. Стиснути масив, вилучивши з нього всі елементи, модуль яких не перевищує 1, місце яке звільнилось в кінці масиву заповнити нулями.
5. Стиснути масив, вилучивши з нього всі елементи, модуль яких знаходиться в інтервалі , місце яке звільнилось в кінці масиву заповнити нулями.
6. Перетворити масив таким чином, щоб спочатку розміщувались всі елементи рівні 0, а потім всі інші.
7. Перетворити масив таким чином, щоб спочатку розміщувались всі елементи, які мають непарні індекси, а потім з парними індексами.
8. Перетворити масив таким чином, щоб спочатку розміщувались всі елементи модуль яких не перевищує 1, а потім всі інші.
9. Перетворити масив таким чином, щоб всі елементи рівні 0, розміщувались в кінці.
10. Перетворити масив таким чином, щоб спочатку розміщувались всі елементи, що мають парні індекси, а потім з непарними індексами.
11. Стиснути масив, вилучивши з нього всі елементи, значення яких знаходиться в інтервалі , місце яке звільнилось в кінці масиву заповнити нулями.
12. Перетворити масив таким чином, щоб спочатку розміщувались всі елементи, ціла частина яких лежить в інтервалі , а потім всі інші.
13. Впорядкувати елементи масиву за спаданню модулів елементів.
14. Впорядкувати елементи масиву за зростанню модулів елементів.
15. Перетворити масив таким чином, щоб спочатку розміщувались всі від’ємні елементи, а потім всі додатні (0 вважати додатнім).

**Завдання 4.**

1. Дано одновимірний масив Х, який складається з 12 еле-ментів, обчислити



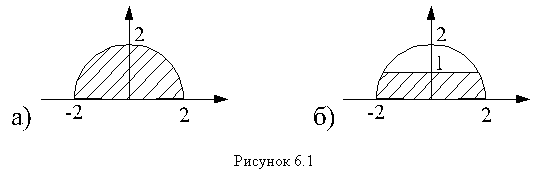
Результат вивести у вигляді масиву у.

2. Обчислити значення виразу



Аргумент х змінюється від 0.8 до 1.7 з кроком 0.05 (радіан). Результати обчислень вивести у вигляді масиву.

3. Дано координати точок х1, y1, x2, y2,...,x20,y20 та рівняння прямої y=ax+b; b= -2.1; a=1.34. Сформувати і вивести на друк масив С, який складається з ординат точок, що належать цій пря-мій, і визначити їх кількість.

 4. Дано координати точок х1, y1, x2, y2,...,x12,y12. Визначити, чи належать точки заштрихованій фігурі (рис.6.1).

5. Дано координати точок х1, y1, x2, y2,...,x15,y15. Сформувати і вивести на друк два масиви: масив А - ординати точок, які знахо-дяться в першій чверті, і їх кількість; масив В - абсциси точок, які знаходяться в третій чверті, і їх кількість.

6. Дано координати точок х1, y1, x2, y2,...,x15,y15. Якщо точка попадає в площину, обмежену колом х2+y2=R2 , то вивести на друк координати цієї точки і повідомлення “ТОЧКА В ПЛОЩИНІ”; якщо точка за межами кола, то вивести на друк координати точки і повідомлення “ТОЧКА ЗА МЕЖАМИ”. R=15.

7. Дано цілочисловий одновимірний масив, який складається з 16 елементів. Знайти парні елементи масиву, які більші числа А, і їх кількість. Число А ввести з клавіатури.

8. Дано одновимірний масив P, який складається з 16 елементів. Знайти номери тих елементів, модуль яких дорівнює заданому числу А. Число А ввести з клавіатури.

9. Дано цілочисловий одновимірний масив T, який складається з 25 елементів. Сформувати два масиви: перший, який складається з додатних елементів масиву T, другий - з від’ємних.

10. Дано одновимірний масив P, який складається з 30 елементів. Знайти мінімальний елемент і його порядковий номер. ВС числа, які знаходяться перед мінімальним елементом, поділити на нього. Вивести на друк заданий і новоутворений масиви.

11. Дано одновимірний масив P, який складається з 32 еле-ментів. Обчислити значення функції для від’ємних елементів ма-сиву і підрахувати їх кількість.



12. Обчислити елементи масиву T, який складається з 8 еле-ментів:



13. Дано одновимірний масив В, який складається з 15 еле-ментів. Сформувати вектор А, першим елементом якого є найменше число, а наступні елементи вектора А - числа, які знаходяться за найменшим елементом масиву В.

14. Дано одновимірний масив Х, який складається з 15 еле-ментів. Визначити півсуму найбільшого і найменшого елементів масиву.

15. Дано одновимірний масив С, який складається з 16 елементів. Поміняти місцями мінімальний і максимальний елементи.

16. Дано одновимірний масив В, який складається з 12 елементів. Сформувати вектор С, що повинен складатися з елементів вектора В, які знаходяться перед максимальним елементом. Вивести на друк вектори В і С.

17. Дано цілочисловий одновимірний масив С, який складається з 15 елементів. Знайти максимальний елемент і його порядковий номер. Всі непарні числа, які знаходяться за максимальним елементом, звести до квадрата. Вивести на друк заданий і новоутворений масиви.

18. Дано одновимірний масив Х, розміром 20 елементів. Обчислити суму кожної пари сусідніх елементів масиву.

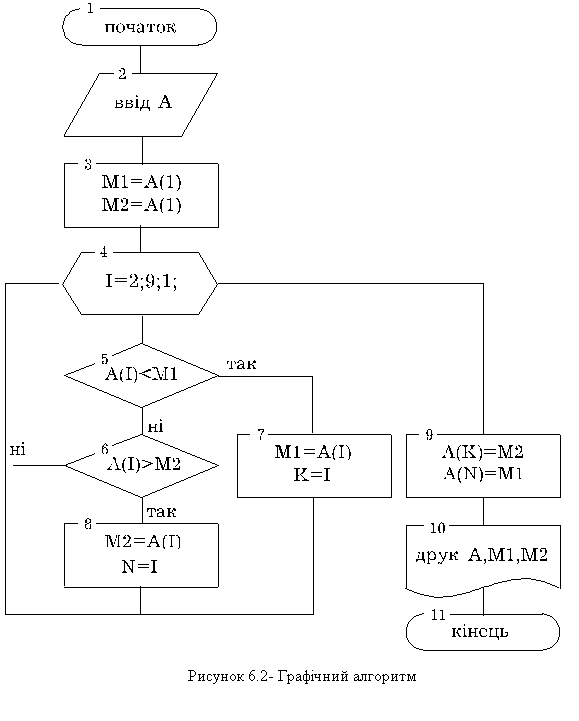
Додаткові завдання

1. Дано одновимірний масив Х, розміром 15 елементів. Провести циклічний зсув елементів у масиві вправо на 2 позиції.
2. Дано одновимірний масив С, який складається з 12 елементів. Вилучити з масиву k-й елемент масиву (k<12).

3. Дано одновимірний масив Х, який складається з 12 елементів. Замінити одиницями елементи масиву, які знаходяться після найменшого елементу.

Приклад виконання. Дано одновимірний масив А, який складається з 9 елементів (21; 32.6; 8.3; -4; 12.2; 4.5; 3.8; 7.5; 6.1). Поміняти місцями мінімальний і максимальний елементи, і вивести на друк перетворений масив, а також мінімальний і максимальний елементи.

1. Графічний алгоритм показаний на рис.6.2

Ідентифікація змінних

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Змінна | А | Міні- мальний елемент | Макси- мальний елемент | Номер міні-мального елемента | Номер макси- мального елемента |
| Ідентифікатор | А | M1 | М2 | K | N |

4. Програма мовою С

#include<stdio.h>

#include<math.h>

#define NUM 9

main()

{

float a[NUM]={21,32.6,8.3,-4,12.2,4.5,3.8,7.5,6.1},

m1,m2;

int i,k,n;

printf(“ВИХІДНІ ДАНІ \n”);

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

printf(“a[%i]=%f\n”,i,a[i]);

printf(“РЕЗУЛЬТАТИ РОЗРАХУНКУ \n”);

m1=m2=a[0];

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

{

if(m1<a[i])

{

m1=a[i];

k=i;

}

if(m2>a[i])

{

m2=a[i];

n=i;

}

}

a[k]=m2; a[n]=m1;

printf(“max=%f min=%f\n”,m1,m2);

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

printf(“a[%i]=%f\n,i,a[i]);

}

ВИХІДНІ ДАНІ

A(0)=21.000000

A(1)=32.600000

A(2)=8.300000

A(3)= -4.000000

A(4)=12.200000

A(5)=4.500000

A(6)=3.800000

A(7)=7.500000

A(8)=6.100000

РЕЗУЛЬТАТИ РОЗРАХУНКУ

max=32.600000 min= -4.000000

a[0]=21.000000

a[1]= -4.000000

a[2]=8.300000

a[3]=32.600000

a[4]=12.200000

a[5]=4.500000

a[6]=3.800000

a[7]=7.500000

a[8]=6.100000

Пояснення до програм

Після введення масиву А, символ 2 (рис.5.1) мінімальному М1 і максимальному М2 значенню присвоюється значення першого елемента масиву, блок 3. Блоки 5, 6, 7, 8 відображають пошук мінімального і максимального елементів масиву, який виконується в циклі. Блок 4 являє собою заголовок циклу, в ньому проводиться перебір індексів елементів масиву. В блоці 5 порівнюється елемент масиву з мінімальним значенням. Якщо значення цього елемента менше від М1, то змінній М1 присвоюється значення елемента. В противному випадку аналогічно відбувається перевірка для знаходження максимального елемента масиву, блок 6. Після закінчення циклу максимальний і мінімальний елементи міняються місцями, блок 9. На друк виводяться перетворений масив і значення мінімального і максимального елементів масиву.

**Питання для самоперевірки**

1. Як оголошуються масиви мовами?
2. Для чого оголошуються масиви?
3. Як записується елемент масиву?
4. Які способи введення і виведення одновимірних масивів ви знаєте?
5. Як визначається номер індексу, який заданий арифметичним виразом?
6. Охарактеризуйте основні атрибути масиву.
7. Який алгоритм пошуку найменшого елемента масиву?
8. Чому необхідно ввести додаткову змінну для перестановки місцями елементів масиву?

**Лабораторна робота №7**

**ЦИКЛІЧНІ ПРОГРАМИ З НАКОПИЧЕННЯМ**

**СУМИ АБО ДОБУТКУ**

**Мета роботи:** Навчитись створювати програми з накопиченням суми або добутку

**Теоретичні відомості**

Особливосі програм з накопиченням суми або добутку полягає в тому, формування заголовку циклу змінній, в якій накопичується сума, присвоюється нуль, а змінній дл добутку – одиниця. Структура програми являє собою цикл з параметром, який є індексом масиву, і містить розгалуження. Очевидно, що цикл можна організувати за допомогою оператора циклу чи оператора умовного переходу. Також можуть бути використані циклічні структури з передумовою або післяумовою

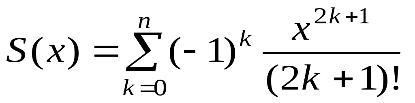
Для задач цього розділу необхідно скласти графічний алгоритм, провести ідентифікацію змінних, скласти прграму і розв’язати задачу на ЕОМ.

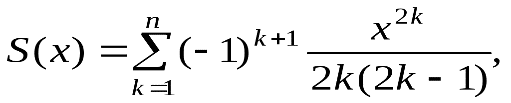
**Варіанти завдань.**

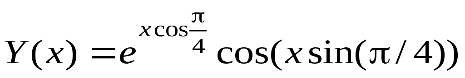
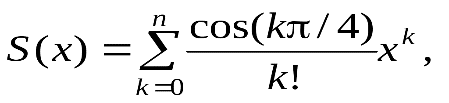
**Завдання 1**

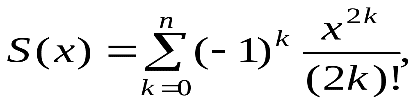
1. Для кожного ***x***, що змінюється від ***a*** до ***b*** з кроком ***h***, знайти значення функції *Y(x)*, суми *S(x)* і |*Y(x)* - *S(x)|* і вивести у вигляді таблиці. Значення *a*, *b*, *h* і *n* вводяться з клавіатури.

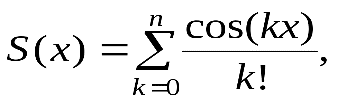
Роботу програми перевірити для *a* = 0,1; *b* = 1,0; *h* = 0,1; значення параметра *n* вибрати залежно від завдання.

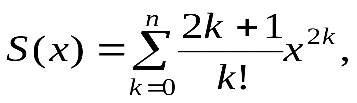
1. , .

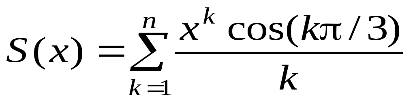
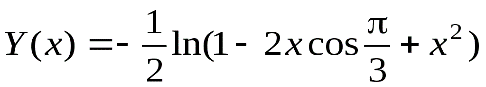
2. ****.

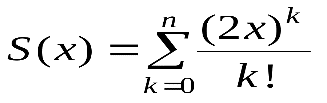
3. .

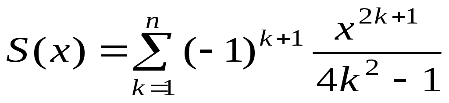
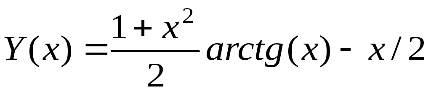
4. .

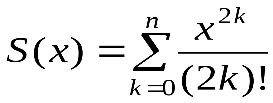
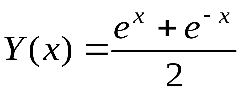
5. .

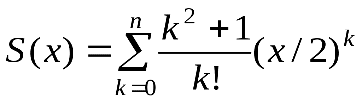
6. .

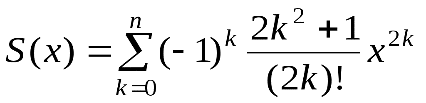
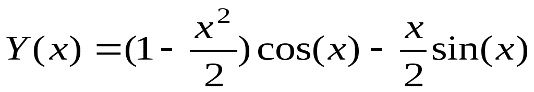
7. , .

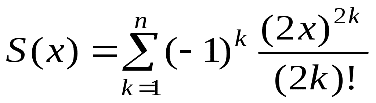
8. , .

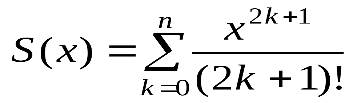
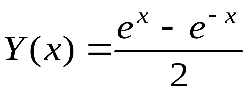
9. , .

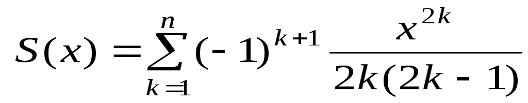
10. , .

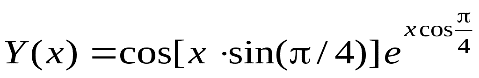
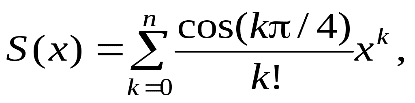
11. , .

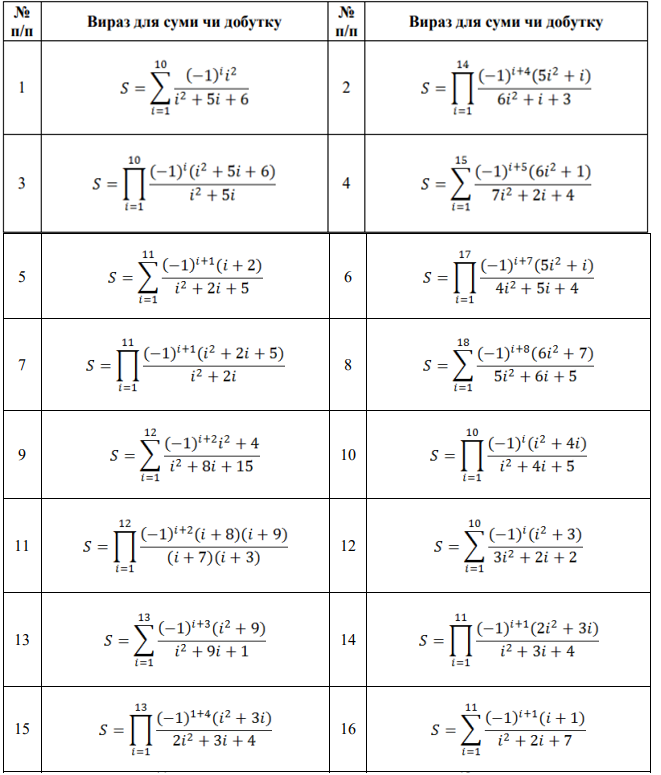
12. , .

13. , .

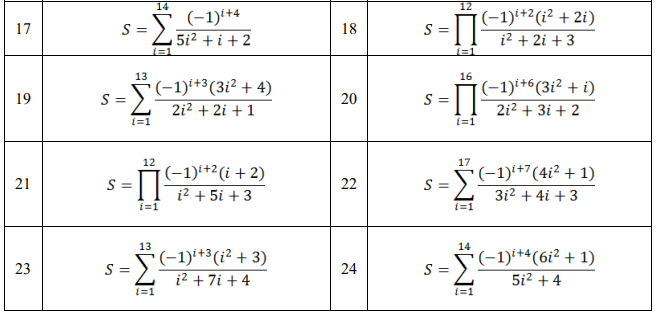
14. , .

15. ****, ****.

16. .

**Завдання2**





**Завдання 3**

1. Обчислити значення функції



Аргумент х змінюється від початкового значення 1.3 до кінцевого значення -14.3. Кількість значень аргумента n=9; a=1.32.

2. Обчислити значення функції



Аргумент z змінюється від початкового значення 1.2 з кроком 0.25 до кінцевого 4.2; х=8.35; а=3.214.

3. Дано одновимірний масив з 25 чисел. Знайти суму перших непарних 11 чисел.

4. Дано одновимірний масив Y, який складається з 15 елементів. Обчислити функцію



5. Дано одновимірний масив С, який складається з 15 елементів. Обчислити і надрукувати добуток додатних і суму від’ємних елементів.

6. Дано одновимірний масив С, який складається з 12 елементів. Обчислити вС значення функції, а також суму значень функцій для від’ємних елементів масиву:



7. Дано одновимірний масив А, який складається з 18 елементів. Обчислити і надрукувати суму від’ємних парних елементів.

8. Дано одновимірний масив Т, який складається з 18 елементів. Обчислити і надрукувати добуток перших 7 елементів і суми наступних елементів, які знаходяться на парних позиціях.

9. Дано одновимірний масив С, який складається з 15 елементів. Обчислити і надрукувати добуток непарних елементів і їх кількість.

10. Дано одновимірний масив Y, який складається з 14 елементів. Обчислити і надрукувати суму перших 8 елементів і добуток наступних елементів, які діляться на 3 з остачею 2.

11. Дано два одновимірні масиви чисел Х і Т, кожний з яких складається з 12 елементів. Обчислити:



12. Знайти середнє арифметичне з модулів обчислених значень функції



13. Дано два одновимірні вектори Х і Т, кожний з яких скла-дається з 10 елементів. Обчислити їх скалярний добуток.

14. Обчислити значення



де аі, bі визначаються за формулами





15. Дано одновимірний масив М, який складається з 18 елементів. Обчислити кількість від’ємних елементів і їх суму.

16. Обчислити



17. Дано одновимірний масив С, який складається з 16 елементів. Обчислити середнє арифметичне значення парних еле-ментів масиву, які діляться на 3 з остачею 1.

18. Дано одновимірний масив Р, який складається з 12 елементів. Обчислити суму модулів вСх від’ємних елементів більших від числа А(А=1.25). Число А ввести з клавіатури.

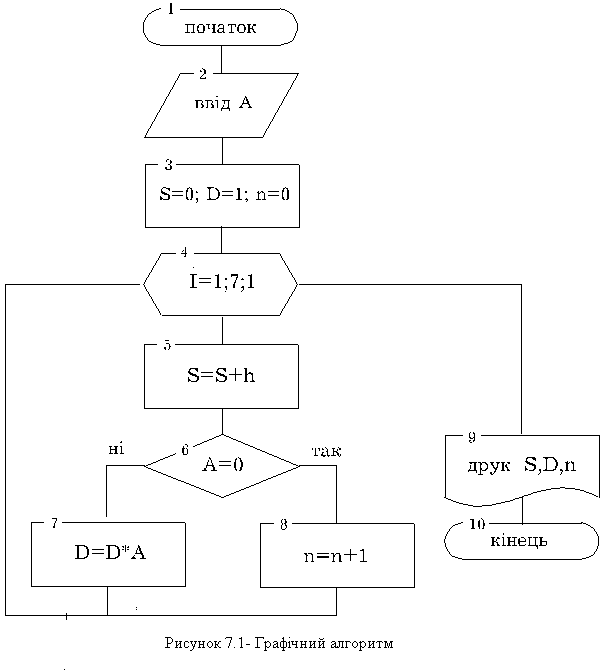
19. Дано одновимірний масив В, який складається з 12 елементів. Обчислити добуток елементів масиву, які не дорівнюють нулю, значення яких знаходяться в межах А<B(I)<C, і їх кількість. Значення А і С ввести з клавіатури.

20. Дано цілочисловий одновимірний масив А, який складається з 14 елементів. Обчислити і надрукувати суму парних елементів, які знаходяться на непарних позиціях, і їх кількість.

Приклад виконання. Дано вектор, який складається з 7 елементів. Обчислити суму вcіх елементів і добуток ненульових елементів, а також підрахувати кількість нульових елементів. A={8.9; -3.4; 2.6; 0; 4.2; 3.1; 0}.

1. Графічний алгоритм показаний на рис.6.1
2. Ідентифікація змінних

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Змінна | А | S | D | n |
| Ідентифікатор | A(i) | S | D | N |



Де S – сума елементів,

D – добуток ненульових елементів,

n – кількість нульових елементів.

4. Програма мовою С

#include <stdio.h>

main( )

{

float a[7]={8.9, -3.4, 2.6, 0, 4.2, 3.1, 0},

s=0,d=1,m2;

int i,n=0;

printf(“DATA:\n”);

for(i=0;i<7;i++)

{

printf(“a[%i]=%f\n”,i,a[i]);

s+=a[i];

if (a[i]= =0)

n++;

else

d\*=a[i];

}

printf(“СУМА= %f\n”ДОБУТОК НЕНУЛ. ЕЛЕМ.=%f\n”,s,d);

printf(“КІЛЬКІСТЬ НУЛ. ЕЛЕМ.=%i\n”,n);

}

СУМА=15.400000

ДОБУТОК НЕНУЛ. ЕЛЕМ.= -1024.362450

КІЛЬКІСТЬ НУЛ. ЕЛЕМ.= 2

Пояснення до програм

Особливість програм з накопиченням суми або добутку полягає в тому, що формування суми чи добутку проводиться в циклі. Перед заголовком циклу, блок 3, рис.7.1, змінній, в якій накопичується сума, присвоюється нуль (S=0), а змінній для добутку – одиниця (D=1). Обчислення добутку проводиться при хибності умови A(I)=0, блок 6. Структура програми являє собою цикл з параметром, який є індексом масиву, і містить розгалуження. Очевидно, що цикл можна організувати за допомогою оператора циклу чи оператора умовного переходу. Також можуть бути використані циклічні структури з передумовою чи післяумовою.

**Питання для самоперевірки.**

1.Яка структура алгоритму накопичення суми?

2.Яка структура алгоритму накопичення добутку?

3.Чому змінній, в якій накопичується добуток, перед його обчисленням присвоюють одиницю?

4.Яка структура алгоритму знаходження парних чисел; чисел кратних трьом, п’яти?

5.Побудуйте розгалуження за допомогою логічного умовного оператора.

6.Як побудувати програму накопичення суми і добутку, користуючись циклічними структурами з передумовою, з післяумовою?

**Лабораторна робота №8**

**МАСИВ ЯК СТРУКТУРОВАНИЙ ТИП. ДВОВИМІРНІ МАСИВИ.**

**Мета роботи:** Засвоїти техніку використання двовимірних масивів при програмуванні мовою С. Ознайомитись зі способами оголошення та ініціалізації двовимірних масивів.Засвоїти способи виділення пам’яті для динамічних масивів. Навчитись будувати математичну модель задачі.Засвоїти методи перевірки правильності вхідних даних.

**Теоретичні відомості**

**Багатовимірні масиви**

Двовимірний масив (матриця) можна представити як одновимірний масив, кожний елемент якого – масив. Тривимірний масив – це масив, кожний елемент якого являє собою двовимірну матрицю.

char Matrix2D[6][9];              // Двовимірний масив 6x9 елементів

unsigned long Arr3D[4][2][8];     // Тривимірний

Багатовимірні масиви ініціалізуються в порядку якнайшвидшої зміни само­го правого індексу: спочатку відбувається присвоєння початкових значень всіх елементів останнього індексу, потімпопереднього і т. д.:

int Mass[3][2][4] = {1,2,3,4,5,6,7,8,9,10,11,12,

                 13,14,15,16,17,18,19,20,21,22,23,24};

Рекомендується для наочності групувати дані за допомогою проміжних фігурних дужок:

int Mass[3][2][4]={{{l,2,3,4},{5,6,7,8}},

                  {{9,10,ll,12},{13,14,15,16}},

                  {{17,18,19,20},{21,22,23,24}};

Для багатовимірних масивів при ініціалізації дозволяється опускати лише величину першої розмірності:

int main()

{

  char x[][3]={{9,8,7},{6,5,4},{3,2,1}};

  for(int i=0; i<3; i++)

  {

       for(int j=0; j<3; j++)

            printf("%d ", (int)x[i][j]);

       printf("\n");

  }

  return 0;

}

**Динамічне виділення пам’яті під  масиви**

Змінні у програмах повинні розміщатися в одному з трьох місць: в області даних програми, в області стеку, в області вільної пам’яті (купи).

Кожній змінній у програмі може відводитися пам’ять або статично (в момент завантаження), або динамічно (у процесі виконання програми).

До цих пір всі використовувані масиви оголошувались статично, а отже, зберігали  значення своїх елементів в області даних. Якщо кількість елементів невелика, таке розміщення виправдано. Але досить часто виникають випадки, коли необхідно мати великі масиви даних або розмір масиву заздалегідь не може бути визначений. Тут на допомогу приходить можливість використання динамічної пам’яті.

Для того, щоб у пам’яті можна було розмістити будь-який динамічний об’єкт, для нього необхідно попередньо виділити відповідне місце. По завершенні роботи з об’єктом виділену пам’ять необхідно звільнити.

Виділення пам’яті можна здійснювати двома способами.

**І спосіб**: функції **malloc(),** **calloc()**, **free()**. Ці функції описані у заголо­вочному файлі <stdlib.h> або <malloc.h>

Функція malloc(size) виділяє size байтів з купи. У випадку успішного виділення пам’яті покажчик встановлюється на виділений блок пам’яті. При невдалому виділенні пам’яті функція повертає NULL.

Функція calloc(num, size), окрім виділення області пам’яті під масив об’єктів, ще здійснює ініціалізацію елементів масиву нульовими значеннями. Тут num вказує, скільки елементів буде зберігатися у масиві, а size – розмір кожного елемента у байтах.

Наприклад, якщо необхідно виділити пам’ять для масиву з n цілих чисел типу long, це можна зробити за допомогою оператора:

long \* fptr = (long \*)malloc(n\*sizeof(long));

Якщо необхідно виділити пам’ять під двовимірний масив розмірності m×n, це можна зробити за допомогою оператора

float\* fptr = (float\*)malloc(n\*m\*sizeof(float));

Функція free(\*block) звільняє пам’ять, на яку вказує покажчик block.

**ІІ спосіб**: функції **new(),  delete().**Ці функції з’явилися в С++.

malloc(), calloc(), free() працюють і в С, і в С++. Нові опе­ра­то­ри гнучкого розподілення пам’яті new() i delete() мають додаткові можли­вості.

Якщо оператори malloc(), calloc() повертають пустий покажик, який далі пере­творюється до потрібного типу, то оператор new() повертає покаж­чик на той тип, для якого виділяється пам’ять, і додаткових перетворень не потребує.

Синтаксис операторів:

тип \*ім’я\_масиву = new тип [тип число];

. . .

delete[] ім’я;

Для завдань, наведених в таблиці 8, необхідно:

* скласти графічний алгоритм з використанням операторів циклу або умовних операторів;
* скласти програму для опрацювання масиву;
* вивести на друк необхідні дані;
* розв’язати задачу на ЕОМ в діалоговому режимі;
* проаналізувати одержані результати.

**Варіанти завдань**

**Завдання 1.** Дана цілочислова прямокутна матриця.

1. Визначити суму від’ємних елементів матриці з обома парними індексами.
2. Визначити суму додатних елементів матриці з першим парним і другим непарним індексами.
3. Визначити добуток від’ємних елементів матриці з обома непарними індексами.
4. Визначити добуток додатних елементів матриці вище головної діагоналі.
5. Визначити суму від’ємних елементів матриці з парною сумою індексів.
6. Визначити суму додатних елементів матриці з непарною сумою індексів.
7. Визначити добуток додатних парних елементів матриці.
8. Визначити добуток додатних елементів матриці нижче головної діагоналі.
9. Визначити кількість від’ємних елементів матриці з обома парними індексами.
10. Визначити кількість додатних елементів матриці з першим парним і другим
11. Визначити кількість від’ємних елементів матриці з обома непарними індексами.
12. Визначити кількість додатних елементів матриці вище головної діагоналі.
13. Визначити суму від’ємних елементів матриці на побічній діагоналі.
14. Визначити суму додатних елементів матриці вище побічної діагоналі.

**Завдання 2.**

|  |  |
| --- | --- |
|  | Побудувати матрицю *А*, елементи якої задаються формулою:  , , .  Побудувати одновимірний масив (переписати матрицю в одновимірний масив). |
|  | Дано матрицю , та матриця *В* такої самої розмірності. За­мі­ни­ти всі нульові елементи матриці *А* відповідними елементами матриці *В*. |
|  | Побудувати квадратну матрицю *А*, елементи якої задаються формулою:  , .  Знайти максимальний елемент головної діагоналі і добуток елементів того ряд­ка, у якому він знаходиться. |
|  | Дано дійсну матрицю розмірності , всі елементи якої різні. Знайти ска­ляр­ний добуток *i*-го рядка і *j*-го стовпчика (*i*, *j* задаються користувачем). |
|  | Побудувати квадратну матрицю *А*, елементи якої задаються формулою:  .  Сформувати одновимірний масив *b*, *i*-ий елемент якого є добутком усіх еле­ментів *i*-го стовпчика матриці *А*. Знайти найбільший парний елемент масиву *b*. |
|  | Побудувати прямокутну матрицю *А*, елементи якої задаються формулою:    Обчислити добуток елементів, добуток ін­дек­сів яких менший за задане число *x***.** |
|  | Побудувати прямокутну матрицю *А*, елементи якої задаються формулою:    Обчислити суму елементів матриці *А*, сума індексів яких непарна. |
|  | Побудувати квадратну матрицю *А*, елементи якої задаються формулою:    Знайти найбільший за модулем елемент матриці *А* та його індекси. |
|  | Побудувати прямокутну матрицю *А*, елементи якої задаються формулою:    Обчислити суму елементів матриці *А*, сума індексів яких непарна. |
|  | Побудувати прямокутну матрицю *А*, елементи якої задаються формулою:    Обчислити суму елементів матриці *А*, сума індексів яких непарна. |
|  | Побудувати прямокутну матрицю *А*, елементи якої задаються формулою:    Обчислити суму додатних елементів елементів матриці *А*. |
|  | Побудувати прямокутну матрицю *А*, елементи якої задаються формулою:    Обчислити суму елементів матриці *А*, сума індексів яких непарна. |
|  | Побудувати квадратну матрицю *А*, елементи якої задаються формулою:  .  Сформувати одновимірний масив *b*, *i*-ий елемент якого є добутком усіх еле­ментів *i*-го стовпчика матриці *А*. Знайти найбільший парний елемент масиву *b*. |
|  | Побудувати квадратну матрицю *А*, елементи якої задаються формулою:  .  Сформувати одновимірний масив *b*, *i*-ий елемент якого є добутком усіх еле­ментів *i*-го стовпчика матриці *А*. Знайти найбільший непарний елемент масиву *b*. |

**Завдання 3.**

|  |  |
| --- | --- |
|  | Створити програму, для знаходження детермінанта квадратної матриці . |
|  | Дано матрицю  та вектор . Перевірити, чи виконується рівність . |
|  | Дано матриці  і  розмірності , знайти . |
|  | Дано матриці  і . Знайти матрицю . |
|  | Дано матрицю  і вектор . Знайти вектор . |
|  | Дано матрицю  і вектор . Знайти вектор . |
|  | Дано матрицю  і число . Знайти добуток числа на матрицю. |
|  | Дано матрицю . Знайти матрицю транспоновану до даної. |
|  | Дано матрицю . Звести дану матрицю до верхньої трикутної матриці. |
|  | Дано матрицю . Звести дану матрицю до нижньої трикутної матриці. |
|  | Дано матрицю .Перевірити, чи є дана матриця діагональною матрицею. |
|  | Дано матрицю .Перевірити, чи є дана матриця верхньою трикутною матрицею. |
|  | Дано матрицю .Перевірити, чи є дана матриця нижньою трикутною матрицею. |
|  | Дано матрицю .Перевірити, чи є дана матриця симетричною відносно головної діагоналі. |

**Завдання 4.** Дана цілочислова квадратна матриця.

1. Розмістити елементи парних рядків у порядку зростання.
2. Розмістити елементи непарних рядків у порядку зростання.
3. Розмістити елементи парних стовпців у порядку зростання.
4. Розмістити елементи непарних стовпців у порядку зростання.
5. Розмістити елементи парних рядків у порядку спадання.
6. Розмістити елементи непарних рядків у порядку спадання.
7. Розмістити елементи парних стовпців у порядку спадання.
8. Розмістити елементи непарних стовпців у порядку спадання.
9. Розмістити елементи діагоналі матриці у порядку зростання.
10. Розмістити елементи діагоналі матриці у порядку спадання.
11. Циклічно зсунути парні рядки матриці зліва направо на k позицій.
12. Циклічно зсунути парні стовпці матриці зліва направо на k позицій. слідування.
13. Перемістити від’ємні елементи парних рядків наліво не міняючи їх порядок
14. Перемістити від’ємні елементи парних стовпців вверх не міняючи їх порядок слідування.

**Завдання 5.**

**1.** Дана цілочислова прямокутна матриця. Визначити кількість рядків, які не містять жодного нульового елемента.

**2**. Дана цілочислова прямокутна матриця. Визначити кількість стовпців, які не містять жодного нульового елемента.

**3**. Дана цілочислова прямокутна матриця. Визначити кількість стовпців, які містять хоча б один нульовий елемент.

**4**. Дана цілочислова квадратна матриця. Визначити добуток елементів в тих рядках, які не містять від’ємних елементів.

**5**. Дана цілочислова квадратна матриця. Визначити суму елементів в тих стовпцях, які не містять від’ємних елементів.

**6**. Дана цілочислова прямокутна матриця. Визначити суму елементів в тих стовпцях, які містять хоча б один від’ємний елемент.

**7**. Дана цілочислова матриця 8×8. Знайти такі k, що k-й рядок матриці співпадає з k-м стовбцем.

**8.**Переставляючи стовпці даної цілочислової матриці, розташувати їх у відповідності з ростом характеристик. Характеристикою стовпця цілочислової матриці назвемо суму модулів його від’ємних непарних елементів.

**9**. Сусідами елемента матриці називаються елементи, хоча б один індекс яких відрізняється на 1 від заданого. Операція згладжування матриці дає нову матрицю того ж розміру, кожен елемент якої знаходиться як середнє арифметичне сусідніх елементів. Побудувати згладжену матрицю до заданої.

**10**. Сусідами елемента матриці називаються елементи, хоча б один індекс яких відрізняється на 1 від заданого. Елемент матриці називається локальним мінімумом, якщо він строго менший від своїх сусідів. Знайти кількість локальних мінімумів у заданій матриці.

**11**. Коефіцієнти системи лінійних рівнянь задані у виді прямокутної матриці. За допомогою допустимих перетворень звести матрицю до трикутного виду.

**12.** Ущільнити задану матрицю, вилучаючи із неї рядки і стовпці, заповнені нулями.

**13**. Ущільнити задану матрицю, вилучаючи із неї рядки і стовпці, заповнені максимальним елементом.

**14**. Ущільнити задану матрицю, вилучаючи із неї рядки і стовпці, заповнені елементом, розміщеним на головній діагоналі.

**Завдання 6.**

**1.** Дана цілочислова прямокутна матриця. Визначити максимальне із чисел, яке зустрічається в даній матриці більше одного разу.

**2**. Дана цілочислова прямокутна матриця. Переставляючи рядки даної матриці, розташувати їх у відповідності з ростом характеристик. Характеристикою рядка цілочислової матриці назвемо суму її додатних парних елементів.

**3**. Дана цілочислова прямокутна матриця. Визначити номер рядка, в якому знаходиться сама довша серія однакових елементів.

**4**. Дана цілочислова квадратна матриця. Визначити максимум серед сум елементів діагоналей, паралельних головній діагоналі матриці.

**5**. Дана цілочислова квадратна матриця. Визначити мінімум серед сум модулів елементів діагоналей, паралельних побічній діагоналі матриці.

**6**. Дана цілочислова прямокутна матриця. Визначити номера рядків і стовпців всіх сідлових точок матриці. Матриця А має сідлову точку А0, якщо Aij є мінімальним елементом в і-у рядку і максимальним в j-у стовпці.

**7**. Дана цілочислова матриця 8×8. Знайти суму елементів в тих рядках, які містять хоча б один від’ємний елемент.

**8.**Знайти суму елементів в тих стовпцях, які містять хоча б один від’ємний елемент такі k, що k-й рядок матриці співпадає з k-м стовбцем;

**9**. Сусідами елемента матриці називаються елементи, хоча б один індекс яких відрізняється на 1 від заданого. Операція згладжування матриці дає нову матрицю того ж розміру, кожен елемент якої знаходиться як середнє арифметичне сусідніх елементів. У згладженій матриці знайти суму модулів елементів, розташованих нижче головної діагоналі.

**10**. Сусідами елемента матриці називаються елементи, хоча б один індекс яких відрізняється на 1 від заданого. Елемент матриці називається локальним мінімумом, якщо він строго менший від своїх сусідів. Знайти суму модулів елементів, розташованих вище головної діагоналі.

**11**. Коефіцієнти системи лінійних рівнянь задані у виді прямокутної матриці. Знайти кількість рядків матриці, середнє арифметичне елементів яких менше за задану величину.

**12.** Знайти номер першого із рядків, який містить хоча б один додатний елемент.

**13.** Дана цілочислова квадратна матриця. Визначити кількість рядків, у яких елементи упорядковані за зростанням.

**14.** Дана цілочислова квадратна матриця. Визначити кількість рядків, елементи яких утворюють арифметичну прогресію

**Завдання 7**

1. Дано матрицю А розміром 4х5. Визначити кількість елементів в кожному рядку матриці А, модуль яких дорівнює порядковому номеру елемента в рядку. На друк вивести кількість таких елементів для кожного рядка.

2. Дано матриці А і В розміром 6х6 кожна. Знайти елементи матриці С як півсуму відповідних елементів матриць А і В.

3. Знайти елементи в кожному стовпці матриці G розміром 7х4, які більші числа А і менші числа С. Визначити кількість таких елементів. Числа А= -2 і С=3.5 ввести з клавіатури.

4. Обчислити



Аргумент х змінюється від початкового значення 5.2 до кінце-вого значення 8.3 з кроком 0.75. Вектор В складається з 8 елементів (додатні дійсні числа).

5. Дано матрицю Е розміром 4х6. Сформувати матрицю Q, значення елементів кожного стовпця якої обчислюється як різниця відповідних елементів двох суміжних стовпців матриці Е.

6. Дано матрицю Т розміром 6х5. Поділити елементи кожного стовпця на останній елемент стовпця. Перетворену матрицю надрукувати.

7. Дано матрицю А розміром 6х7. Для кожного рядка матриці обчислити суму елементів, значення яких перевищує задане число С. С=18.6.

8. Дано матрицю А розміром 6х6. Замінити в матриці елементи головної діагоналі нулями.

9. Дано матрицю С розміром nxn. Знайти і надрукувати індекси тих елементів матриці С, для яких cij=cji, а також підрахувати кількість таких елементів.

10. Дано матрицю А розміром 8х7. Поділити кожний елемент стовпця на елемент цього стовпця, який знаходиться на головній діагоналі, якщо цей елемент не дорівнює нулю.

11. Масив С дійсних чисел має 5 рядків і 11 стовпців. Присвоїти значення найменшого елемента масиву змінній М1, номер рядка, де знаходиться цей елемент,- змінній Т, номер

стовпця - змінній С.

12. Одновимірний масив А складається з 30 елементів. Знайти та надрукувати числа, які зустрічаються в масиві більше одного разу.

13. Для кожного рядка заданої матриці А розміром 8х5 знайти та надрукувати номери стовпців, які містять нульові елементи, і їх кількість.

14. Дано матрицю В розміром 5х6. Поділити елементи кожного рядка на елемент, який знаходиться в третьому стовпці цього рядка.

15. Дано квадратну матрицю А 6-го порядку. Знайти суму елементів матриці, які розміщені в рядках з від’ємним елементом на головній діагоналі. Обчислити кількість таких рядків.

16. Обчислити елементи вектора В, кожний з яких дорівнює скалярному добутку суми елементів рядків матриці Q розміром 5х6 на векторі Х, який складається з 6 елементів. Вказівка:



17. Дано двовимірний масив В розміром 15х15. Обчислити суму абсолютних значень елементів масиву, виключивши з неї діагональні елементи:

 i, j=1,2,3,...,15.

18. Дано матрицю Т розміром 6х7. Знайти максимальний і мінімальний елементи для кожного стовпця матриці Т.

19. Дано квадратну матрицю А n-го порядку (n<10). Утворити матрицю n-1 порядку шляхом вилучення з матриці А рядка і стовпця, які розміщені на перетині місцезнаходження мінімального елемента матриці А. Вивести на друк дві матриці і значення мінімального елемента матриці.

20. Дано дві цілочислові квадратні матриці 4-го порядку. Одержати нову матрицю шляхом віднімання від елементів кожного стовпця першої матриці суми елементів відповідних рядків другої матриці.

21. Дано матрицю розміром МхN. Сформувати з цієї матриці вектор за правилом: перші N елементів вектора - це елементи матриці першого рядка, другі N елементів - елементи другого рядка і т.д. (М=4; N=6).

22. Дано матрицю С розміром МхN. Поміняти перший елемент кожного стовпця матриці С з максимальним елементом цього стовпця, другий елемент цього стовпця з мінімальним елементом цього стовпця. Вивести на друк задану і новоутворену матриці.

23. Дано матрицю А( 3,4). Вивести на друк матрицю і номери стовпців і рядків, де знаходяться третій і п`ятий парні елементи цієї матриці.

24. Дано прямокутну матрицю МхN. Одержати нову матрицю шляхом ділення вСх елементів заданої матриці на елемент, найбільший за абсолютною величиною. На друк вивести нову матрицю і максимальний елемент за абсолютною величиною.

Приклад виконання. Дано матрицю В розміром 3х4. Обчислити добуток ненульових елементів кожного стовпця матриці. Результат обчислення вивести у вигляді одновимірного масиву.

Вихідна матриця



Розв’зок задачі.

Добуток обчислюємо за формулою

 і=1,2,3; j=1,2,3,4.

1. Графічний алгоритм наведений на рис.9.1

3. Програма мовою С

#include <stdio.h>

main( )

{

float b[3][4]={

{2, 6, 7.5, 4.1},

{5, 0, 2.3, -6.2},

{3, 1.5, 0, 4.9}

},

d[4];

int i,j;

for(j=0;j<4;j++)

{

d[j]=1; for(i=0;i<3;i++)

if(b[i][j]!=0)

d[j]\*=b[i][j];

printf(“d[%i]=%f\n”,j,d[j]);

}

}

D(0)=30.000000

D(1)=9.000000

D(2)=17.350000

D(3)= -124.557991

Пояснення до програм

Введення елементів матриці в пам’ять ЕОМ проводиться по рядках, тобто у внутрішньому циклі змінюється індекс стовпця, а в зовнішньому - індекс рядка.

При визначенні добутку ненульових елементів кожного стовпця матриці також організовано два вкладені цикли: зовнішній за номером стовпця - параметр циклу J і внутрішній за номером рядка за змінною І. У внутрішньому циклі проводиться перебір всіх елементів J-го стовпця, кожен з яких порівнюється з нулем. Якщо B(I,J)=0, то проводиться обчислення добутку елементів матриці в даному стовпці.

**Питання для самоперевірки**

1. Як оголошується двовимірний масив?
2. Яким способом можна записати значення індексів елементів?
3. Який порядок зміни індексів елементів двовимірного масиву?
4. Які способи введення елементів двовимірного масиву ви знаєте?
5. В чому різниця між перебором елементів двовимірного масиву по рядках і стовпцях?
6. Як вивести елементи масиву в текстовому вікні користуючись мовою Сі?

**Лабораторна робота №9**

**ВКЛАДЕНІ ЦИКЛИ**

**Мета роботи**: набути практичних навиків створення програм із вкладеними циклами.

**Теоретичні відомості**

Поряд із простими циклічними процесами при побудові алгоритмів складних перетворень використовуються вкладені циклічні обчислювальні процеси. Вкладені цикли застосовуються у випадку розрахунків функцій, що залежать від декількох аргументів.Вкладеним називається цикл, що містить в собі один чидекілька інших циклів. Цикл, що охоплює інші цикли, називається зовнішнім, а інші – внутрішніми. Основне правило побудови вкладених циклів – це охоплення зовнішнім циклом внутрішнього чи декількох внутрішніх. Кожен цикл окремо організується точно так, як і простий цикл.

Якщо один цикл повністю міститься в тілі іншого циклу, то таку конструкцію називають вкладеними циклами. На глибину вкладень обмежень не накладають. Вкладені цикли зазвичай використовуються для обробки багатовимірних масивів або ж інших алгоритмів зі складною структурою. При використанні вкладенихциклів треба складати програму в такий спосіб, щоб внутрішній цикл повністювкладався в тіло зовнішнього циклу, тобто цикли не повинні перетинатися. Своєю чергою, внутрішній цикл може містити власні вкладені цикли. Імена параметрів зовнішнього та внутрішнього циклів мають бути різними

Для завдань, наведених в таблиці 8.1, необхідно:

* скласти графічний алгоритм з використанням операторів циклу або умовних операторів;
* скласти не менше двох програм: одна пограма з використанням тільки операторів циклу, друга програма – один з циклів (зовнішній або внутрішній) реалізований за допомогою умовного оператора;
* Вивести на друк значення функції і відповідних їм значень аргументів. Виведення аргументів виконати так, щоб один з них друкувався перед початком внутрішнього циклу;
* розв’язати задачу на ЕОМ в діалоговому режимі;
* проаналізувати одержані результати.

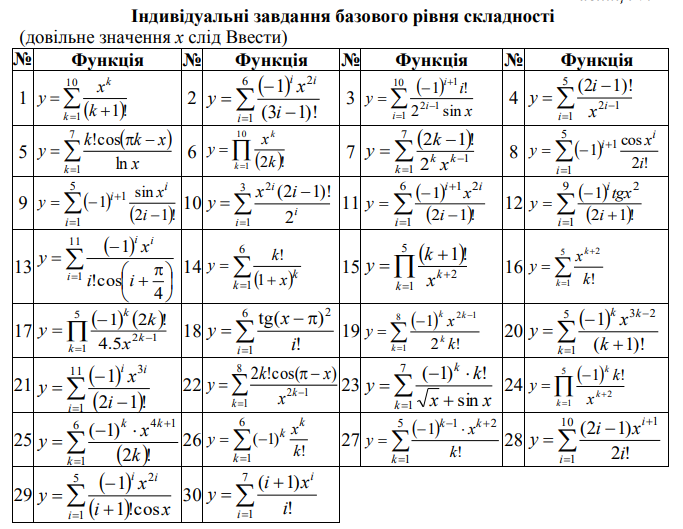
Задачі на вкладені цикли (таблиця 8.1) поділяються на три типи:

Перший тип задач. Варіанти: 2, 5, 8, 11, 14, 17, 21, 24. Необхідно обчислити значення y=f(x,z), де обидва параметри циклу х та у змінюються від початкового до кінцевого значення з постійним кроком.

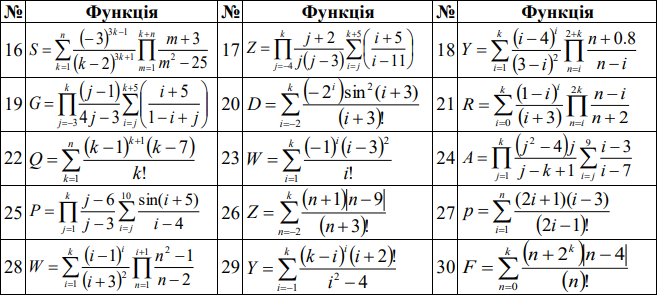
Другий тип задач. Варіанти: 1, 4, 7, 10, 13, 16, 20, 23, 25. Необхідно обчислити значення yi=f(z,xi), де параметр циклу z змінюється від початкового до кінцевого значення з постійним кроком z, а змінна xi(і=1,2,…,n) є елементом одновимірного масиву (дійсні числа).

Третій тип задач. Варіанти: 3, 6, 9, 12, 15, 18, 19, 22. Необхідно обчислити yij=f(xi,zj). В цих задачах х(і=1,2,…,n) і z(j=1,2,…,m) являють собою елементи одновимірних масивів дійсних чисел, які містять відповідно n i m елементів.

Перед складанням алгоритму слід визначити кількість зна-чень функції, які будуть одержані в результаті розв’язання задачі, т.б. кількість значень функції від кожного аргумента. Загальна кількість значень функції дорівнюватиме добутку кількостей значень всіх аргументів.







Таблиця 9.1- Варіанти завдань

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № |  | Параметр циклу з регулярною зміною аргумента | | | | Параметр циклу з індексною змінною | | |
| Варіан ту | Функція | Параметр циклу | Початкове значення | Кінцеве значення | Крок | Позначення індексу | Номер індексу | Значення індексної змінної |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  | с | 0,26 | 1,26 | 0,2 | і | 1 2 3 | 1,95 0.87 1.29 |
| 2 |  |  | 6.35 10.4 | 7.40 17.9 | 0.15 2.5 |  |  |  |
| 3 |  |  |  |  |  | i j | 1 2 3 4 1 2 3 | 1.35 2.6 1.86 0.95 0.60 1.28 3.15 |
| 4 |  | f | -6 | -2.4 | 1.2 | j | 1 2 3 | 28.3 41.7 13.4 |
| 5 |  |  | 8.63 0.75 | 9.83 1.45 | 0.4 0.35 |  |  |  |
| 6 |  |  |  |  |  | i k | 1 2 3 4 1 2 3 | 0.68 2.05 1.81 3.86 1.12 2.03 1.56 |
| 7 |  | x | 6.2 | 8 | 0.3 | i | 1 2 3 4 | 3.28 0.56 2.91 3.26 |
| 8 |  |  | 8.35 –0.9 | 12.8 –0.4 | 1.6 0.25 |  |  |  |
| 9 |  |  |  |  |  | k | 1 2 3 1 2 3 4 | 1.86 2.45 3.7 0.96 4.21 2.87 1.78 |
| 10 |  | a | 1.7 | 2.4 | 0.35 | i | 1 2 3 4 | 0.236 2.851 3.270 7.882 |
| 11 |  |  | 83 -4.3 | 105 –3.8 | 11 0.15 |  |  |  |
| 12 |  |  |  |  |  | k | 1 2 3 1 2 3 4 | 1.3 –1.6 –2.3 4.51 –4.16 2.61 7.28 |
| 13 |  | x | 12.4 | 19.4 | 3.5 | i | 1 2 3 4 | -3.46 1.852 3.03 –2.57 |
| 14 |  |  | 10.6 0.85 | 6.6 1.60 | -2 0.25 |  |  |  |
| 15 |  |  |  |  |  | i j | 1 2 3 1 2 3 4 | 62.6 –34.8 41.71 2.75 8.34 6.95 11.7 |
| 16 |  | h | 81.6 | 105 | 12 | n | 1 2 3 4 | 0.18 –1.86 0.65 –2.46 |
| 17 |  |  | 2.45 6.82 | 3.16 10.3 | 0.35 1.15 |  |  |  |
| 18 |  |  |  |  |  | j m | 1 2 3 1 2 3 4 | 2.46 –7.38 11.6 1.6 0.35 6.71 9.35 |
| 19 |  |  |  |  |  | i j | 1 2 3 4 1 2 3 4 | 21.87 32.45 13.01 48.43 842.3 326.8 485.0 206.3 |
| 20 |  | x | -3.1 | 6 | -1.4 | k | 1 2 3 4 | 4.285 2.06 5.781 1.984 |
| 21 |  |  | 0.8 –3.4 | 1.32 –4.2 | 0.26 –0.4 |  |  |  |
| 22 |  |  |  |  |  | i k | 1 2 3 1 2 3 | -2.4 1.36 –3.35 128.6 85.9 324.3 |
| 23 |  | q | 2.8 | 5.4 | 1.3 | m | 1 2 3 | -1.85 2.39 3.65 |
| 24 |  |  | 1.6 7.35 | 2.5 9.75 | 0.3 1.2 |  |  |  |
| 25 |  | b | -18 | -22 | -2 | j | 1 2 3 | 4.41 1.85 16.35 |

Приклад виконання. Обчислити значення функції



Аргумент змінюється від початкового значення b=21.5 до кінце-вого b=43.5 з кроком b=5.5. Змінна t являє собою одновимірний масив (t1=1.32; t2=2.73; t3=0.86).

1. Графічний алгоритм розв’язання задачі наведений на рис.8.

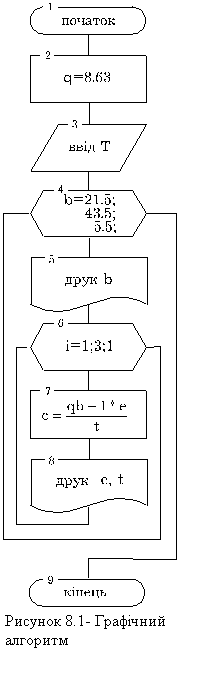


Рисунок 9.1 – графічний алгоритм

2. Ідентифікація змінних

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Змінна | q | b | ti | ci |
| Ідентифікатор | G | B | T(I) | C(I) |

3. Програма мовою С

#include <stdio.h>

#include <math.h>

#define G 8.63

main( )

{

float t[3]={1.32, 2.73, 0.86},

b,c;

int i;

for(b=21.5;b<=43.5;b+=5.5)

{

printf(“b=%f\n”,b);

for(i=0;i<3;i++)

{

c=G\*sqrt(b+1)\*exp(t[i])/t[i];

printf(“C=%f T(%i)=%f\n”,c,i,t[i]);

}

}

}

B=21.500000

C=116.090546 T(0)=1.320000

C=229.912903 T(1)=2.730000

C=112.485580 T(2)=0.860000

B=27.000000

C=129.504425 T(0)=1.320000

C=256.478577 T(1)=2.730000

C=125.482918 T(2)= 0.860000

B=32.500000

C=141.653717 T(0)=1.320000

C=280.539795 T(1)=2.730000

C=137.254944 T(2)=0.860000

B=38.000000

C=152.840302 T(0)=1.320000

C=302.694397 T(1)=2.730000

C=148.094147 T(2)=0.860000

B=43.500000

C=163.262177 T(0)=1.320000

C=323.334534 T(1)=2.730000

C=158.192398 T(2)=0.860000

Пояснення до програм

Алгоритм програми складається з двох циклів, рис.8.1 Зовнішній цикл з параметром b, символ 4, і внутрішній - з параметром І, символ 6. Тіло внутрішнього циклу містить обчислення значень функції с, символ 7, і друкування значень функції с і t, символ 8.1 Аргумент b друкується після заголовка зовнішнього циклу, символ 5.

**Питання для самоперевірки**

1. Чому в програмі мовою GW Basic і С зовнішній цикл - цикл з передумовою, а мовою Pascal - постумовою?
2. Чому не допускається перетин вкладених циклів?
3. Як визначити кількість значень функції при зміні двох аргументів?
4. В наведених програмах побудуйте зовнішній або внутрішній цикли за допомогою оператора умовного переходу.
5. Побудуйте графічний алгоритм для обчислення значень функції трьох змінних y=f(a,b,c); i=1,2,...,N; j=1,2,...,M; k=1,2,...,L.

**Лабораторна робота № 10**

**ФУНКЦІЇ. МЕХАНІЗМ ПЕРЕДАЧІ ПАРАМЕТРІВ. ГЛОБАЛЬНІ ТА ЛОКАЛЬНІ ЗМІННІ**

## **Мета**: отримання навичок з алгоритмізації і програмування задач з використанням підпрограм та звертання до них; навичок з вибору параметрів підпрограм.

**Теоретичні відомості**

Умові C передбачено засоби, завдяки яким можна оформляти послідовність операторів як підпрограму. Підпрограма – це названа група операторів, яку можна виконати в будь-якому місці програми довільну кількість разів.

Усі підпрограми поділяються на два класи: стандартні (зарезервовані) і визначені користувачем. Стандартні підпрограми є частиною мови, вони заздалегідь не описуються. Підпрограми, визначені користувачем, обов'язково описуються. Сам опис не передбачає виконання жодних дій.

У мові C існують лише підпрограми функції, але якщо правильно описати підпрограму, то вона може працювати і як функція, і як процедура.

Функції. Функції в С повинні мати унікальні імена. Суттєве значення має тип значення функції, що повертається. Функція може мати параметри, що необхідні для її виконання. Список параметрів задається в круглих дужках після імені функції. Кількість параметрів може бути довільним. Найпростішим прикладом використання функції є функція main (головна), яка повинна бути присутньою в будь-якій програмі, що розроблюється на С. З неї починається виконання програми. Частіше всього функція main не має параметрів і не повертає значення. В цьому випадку замість відсутніх типів вказується ключове слово void (якщо тип параметру, що повертається, не вказаний, то по замовчуванню приймається int). Приклад: void main(void).

Якщо функція повертає значення, то це значення передається фрагменту, що її викликав, за допомогою виразу, що записується в операторі return. Return викликає завершення роботи даної функції, передачу значення у функцію, що викликала дану і повернення керування на оператор, що йде після виклику функції.

Під час виклику функції вказуються фактичні параметри виклику, їх кількість повинна відповідати числу і типу формальних параметрів у заголовку функції, що викликається. Якщо функція не повертає значення (тобто повертає void, аналог процедури у паскалі), то оператор return може використовуватися у варіанті «return;» (без виразу, що слідує за ним).

Приклад:

#include <stdio.h> #include <conio.h> float pi(void)

|  |  |  |
| --- | --- | --- |
|  |  |  |
| { | // функція тільки повертає значення константи Пі | |
| return 3.14159265359; | | // значення що повертається |

30

|  |  |  |
| --- | --- | --- |
| } |  |  |
| int SummaAandB(int A,int B) | | // функція повертає суму змінних A і B |
| { |  |
| return A+B; |  | // значення що повертається |
| } |  |  |
| void PrintName(void) | |  |
| { | //функція виводить на екран ім’я Serg 10 раз по одному | |
|  | //в рядку, не повертає і не приймає ніяких значень | |
| for(int t=0;t<10;t++) | |  |
| printf(“Serg\n”); } | |  |
| void PrintA(int A) | // функція виводить на екран значення змінної А | |
| { |
| printf(“%d”,A); } |  |  |
| void main(void) |  | // виклик функції pi |
| { float f=pi(); |  |
| printf(“pi=%f\n”,f); | | // виклик функції SummaAandB |
| int i=SummaAandB(1,3); | |
| printf(“summa(A+B)=%d\n”,i); | | // виклик функції PrintName |
| PrintName(); |  |
| PrintA(20); | // виклик функції PrintA | |
| } |  |  |

Функції на мові Сі необхідно оголошувати (прототип функції). У відповідному оголошенню буде дана інформація про параметри. Вона представляється у наступному вигляді:

<тип> <ім’я функції> (<опис пар.\_1>,<опис пар.\_2>,...,<опис пар.\_n>); Для кожного параметру можна вказати тільки його тип (наприклад, <тип>

<ім’я функції> (int, float);), а можна дати і ім’я параметру (наприклад, <тип> <ім’я функції> (int a,float b); ). Дозволено створювати функції зі змінною кількістю параметрів. Тоді під час задавання прототипу замість останнього із параметрів вказується багатокрапка (наприклда void f1(int a, float b,…).

Можна визначити і функцію void f1 (...) {тіло функції} і звертання: f1 (a1, a2), де а1 и а2 достаються якимось особливим чином із стеку чи відомі як глобальні).

Локальні і глобальні змінні. У мові Сі змінні діляться на глобальні і локальні.

Глобальні змінні об’являються у файлі початкового тексту програмного модулю поза будь-якою функцією (локальні об’являються всередині функції). Глобальні змінні створюються в точці оголошення і доступні (видимі) у початковому тексті від точки оголошення до кінця файлу, в якому вони задані (вони видимі і всередині функцій). Глобальні змінні видимі також і для зовнішніх модулів.

Локальні змінні по відношенню до функцій є внутрішніми. Вони починають існувати в точці оголошення всередині функції і знищуються після виходу з неї. Якщо вони записані в списку параметрів функції (у круглих дужках), то треба розглядати таке оголошення як введення до першої відкриваючої фігурної дужки. Для тих локальних змінних, яких нема в списку параметрів, оголошення робиться після першої відкриваючої фігурної дужки.

Програма в Сі, зв’язок між функціями і передача параметрів до функції. Програми на мові Сі зазвичай складаються з деякої кількості окремих функцій (підпрограм), серед яких повинна бути одна з ім’ям main,. з якої починається виконання програми. Як правило, функції мають невеликі розміри, і можуть знаходитись як в одному, так і в декількох файлах. Якщо функції розташовані в різних фізичних файлах, то для виконання їх як одної програми, необхідно зібрати їх у файлі проекту. У Сі заборонено визначати одну функцію всередині іншої, тому всі імена функцій є глобальними. Зв’язок між функціями здійснюється через аргументи, значення, що повертаються і зовнішні (глобальні) змінні. Передача значення (повернення значення) із викликаної функції у ту, що викликала, реалізується за допомогою оператора повернення, який має вигляд: return <вираз>;

Таких операторів у підпрограмі може бути декілька, і тоді вони фіксують відповідні точки виходу. Функція, що викликає може за необхідністю ігнорувати значення, що повертається. Після слова return можна нічого не записувати. В цьому випадку ніякого значення не повертається. Керування передається функції, що викликала і у випадку виходу "по кінцю" без використання return (після закриваючої фігурної дужки).

Приклади:

|  |  |
| --- | --- |
| #include <stdio.h> |  |
| int f1(void) |  |
| { printf("працює f1()"); | //функція повертає значення 1 |
| return 1; } |
| void main (void) | //значення, що повертається використовується |
| { int k=f1(); |
| } | //у зовнішній програмі |
| #include <stdio.h> |  |
| int f1 (int a,int b) | //функція приймає параметри і повертає значення |
| {return a+b;} |
| void main(void) |  |
| {int a=17; int b=16; | //значення, що повертається, виводиться у main |
| printf("%d",f1(a,b)); |
| } |  |
| #include <stdio.h> | //функція приймає параметри |
| f1 (int a,int b) |
| {a+b;} | //і повертає значення за замовчуванням |
| void main(void) |  |

32

{int a=17; int b=16;printf("%d",f1(a,b));}

//результат роботи такий же, як і в попередньому випадку //хоча програма некоректна, про що видається //попередження, як і у варіанті рядка:

{int a=17,b=16, с=f1(a,b); printf("%d",c);}

Приклад використання глобальних змінних:

#include <stdio.h> #include <conio.h>

|  |  |
| --- | --- |
| int a,b; | //глобальні змінні |
| int f1(int x) | //х – локальний формальний параметр, приймає значення |

{return (a+b)\*x;} //фактичного параметра k під час виклику функції f1(k);

|  |  |
| --- | --- |
| void main (void) |  |
| {clrscr(); |  |
| int k=35; a=10;b=12; int c= f1(k); | //с і k – локальні у main |
| printf("%d",c); } |

Якщо деякі змінні, константи, масиви, структури оголошені як глобальні, то їх не треба включати в описок параметрів викликаної функції. Вона все рівно отримає до них доступ.

Програмі main також можуть бути передані параметри під час запуску. В цьому випадку текст програми може виглядати наступним чино:

#include <stdio.h>

void main(int argc, char \*argv[])

//тут в argc – кількість елементів у рядку запуску програми; {char \*s; s=argv[0]; // argv[] – масив посилань на елементи цього

рядка

printf("name programm=\"%s\"\n",s); //перший елемент рядка запуску

//конструкція \” використовується для виводу символу лапки «“» if(argc > 1) {s=argv[1];

printf("parametr programm=\"%s\"\n",s);} //другий елемент рядка запуску

}

Якщо створити програму, наприклад з ім’ям prog.exe, в каталозі D:\BC\WORK и запустити його із цього каталогу командою, наприклад: prog 77, то в результаті отримаємо виведення на екран двох рядків:

name programm=” D:\BC\WORK\PROG.EXE” parametr programm=”77”.

Відповідність між формальними і фактичними параметрами. У мові Сі аргументи функції передаються за значенням, тобто функція отримує в іменах формальних параметрів тимчасову копію кожного аргументу, а не його адресу (параметри значення). Це означає, що в самій функції не може змінюватися значення самого оригінального аргументу у програмі, що викликала функцію. Якщо в якості аргументу функції використовується ім’я масиву, то передається початок масиву (адрес початку масиву), а самі елементи не копіюються. Функція може змінювати елементи масиву, зсовуючись (індексуванням) від його початку.

Відповідність між параметрами. Якщо виклик функції реалізований наступним чином:

int a,b=73, c=40; а=fun(b,с);

Тут b і с – параметри, значення яких передаються до підпрограми. Якщо опис функції починається так: void fun(int b,int c); то імена аргументів, що передаються у виклику і в програмі fun будуть однаковими. Якщо ж опис функції починається, наприклад, рядком void fun(int i,int j); , то замість ім’я b у функції, що викликала, для того ж аргументу у функції fun буде використано ім’я i, а замість с – j .

Якщо до функції звернутись так с=fun(&b,&c);, то підпрограмі передаються адреси змінних b і о. Тому прототип (заголовок функції) повинен бути, наприклад, таким: void fun(int \*k, int \*c); В цьому випадку k отримує адресу змінної b, а с – адресу змінної с. В результаті у програмі, що викликає, с

– це змінна цілого типу, а в програмі, що викликана, с – це вказівник на змінну цілого типу. Якщо у виклику записані ті ж імена, що і в списку параметрів, але вони записані в іншому порядку, то все рівно встановлюється відповідність між i-м іменем у списку і i-м іменем у виклику.

Вище вже вказувалось, що змінні передаються функції по значенню, тому немає прямого способу у функції, що викликається, змінити деяку змінну у функції, що викликає. Однак це легко зробити, якщо передавати у функцію не змінні, а їх адреси.

Приклад:

#include <stdio.h>

void f1(int \*a,int b, int \*c1)// аргументи: a – за адресою, b – за значенням,

|  |  |
| --- | --- |
| {\*c1=\*a+b; return;} | //c1 – за адресою. Вказівник с1 посилається на с |
| void main (void) | // і значення с=\*с1; |
| {int c,a=17, b=16; | //параметри, що передаються; b – за значенням |
| f1(&a,b,&c); |
| printf("%d",c); } | // а і с – за адресою. Результат отримуємо у с. |

Передавання масиву у вигляді параметру. Тут можливі три варіанти.

Причому в прототипі і тілі записи можуть комбінуватися із цих варіантів. 1.Параметр задається як масив (наприклад: int mn[100]).

2.Параметр задається як масив без вказування його розміру (наприклад: int m[ ]).

3.Параметр задається як вказівник (наприклад: int \*m, прототип int f1(int

\*mn).

Незалежно від вибраного варіанту, функції, що викликана, передається вказівник на початок масиву. Самі ж елементи масиву не копіюються.

#include <stdio.h> void f1 (int x[20],int k)

{int i; for (i=0; i<k; i++) x[i]=i;} #define k 5

void main (void)

При складанні програм з використанням підпрограм необхідно:

- скласти графічний алгоритм розв’язання задачі з використанням функції, передбачити введення вихідних даних, звертання до функції, виведення вихідних даних задачі і результатів обчислень;

- скласти окремо графічний алгоритм процедури і навести опис формальних параметрів;

- скласти головну програму, в якій передбачити звернення до процедури. Процедуру-програму можна використати як внутрішню, так і зовнішню (за рекомендацією викладача);

- скласти програму самої процедури;

- розв’язати задачу в діалоговому режимі.

В умовах більшості задач не задані цифрові значення вихідних даних, студентові надається можливість самому задатися необхідними числовими значеннями для тестування складеної програми.

**Завдання 1**

**Скласти програму для обчислення величин, заданих у варіантах завдань.**

 1.      Задано три матриці А(3,3), В(4,3), С(3,2). Обчислити кількість додатних, від’ємних та нульових елементів для кожної матриці. Знаходження даних, ввід елементів матриці оформити через процедуру.

2.      Задано три матриці А(4,4), В(2,2), С(3,3). Обчислити суму елементів головної діагоналі для кожної матриці. Знаходження сум, ввід елементів матриці оформити через процедуру.

3.      Задано три масиви А(6), В(5), С(7). Знайти суму найбільшого і найменшого елементів масивів відповідно. Знаходження сум, ввід елементів масиву оформити через процедуру.

4.      Задано три матриці А(4,4), В(2,2), С(3,3). Знайти максимальне з трьох чисел, які є сумами квадратів елементів, що знаходяться над головними діагоналями відповідних матриць. Знаходження сум, ввід елементів матриці оформити через процедуру.

5.      Задано три масиви А(5), В(8), С(6). Знайти суму від’ємних елементів кожного масиву відповідно. Знаходження сум, ввід елементів масиву оформити через процедуру.

6.      Задано три матриці А(5,3), В(3,3), С(3,2). Обчислити суму від’ємних кожного стовпчика матриць А, В, С відповідно. Знаходження даних, ввід елементів матриці оформити через процедуру.

7.      Задано три масиви А(5), В(6), С(4). Знайти середнє арифметичне від’ємних елементів кожного масиву відповідно. Знаходження середнього арифметичного, ввід елементів масиву оформити через процедуру.

8.      Задано три масиви А(6), В(4), С(8). Знайти добуток ненульових елементів кожного масиву відповідно. Знаходження добутків, ввід елементів масиву оформити через процедуру

9.      Задано три матриці А(3,4), В(2,4), С(3,3). Знайти середнє арифметичне додатних елементів кожного стовпчика матриць А,В,С відповідно. Знаходження середнього арифметичного, ввід елементів матриці оформити через процедуру.

10.  Задано три масиви А(10), В(6), С(8). Обчислити кількість додатних, від’ємних та нульових елементів для кожного масиву. Знаходження даних, ввід елементів масивів оформити через процедуру.

11.  Задано три матриці А(4,3), В(4,2), С(3,4). Знайти середнє арифметичне від’ємних елементів кожної стрічки матриць А, В, С відповідно. Знаходження середнього арифметичного, ввід елементів матриці оформити через процедуру.

12.  Задано три масиви А(7), В(6), С(5). Знайти значення мінімального елемента кожного масиву та його номер. Знаходження значень, ввід елементів масиву оформити через процедуру

13.  Задано три матриці А(3,1), В(2,2), С(3,4). Знайти добуток ненульових елементів для кожної матриці відповідно. Знаходження добутку, ввід елементів матриці оформити через процедуру.

14.  Задано три матриці А(4,2), В(2,3), С(3,4). ). Знайти добуток ненульових елементів кожної непарної стрічки матриць А, В, С. Знаходження добутку, ввід елементів матриці оформити через процедуру.

15.  Задано три масиви А(5), В(4), С(8). Обчислити кількість додатних елементів для кожного масиву та їх загальну кількість. Знаходження даних, ввід елементів масивів оформити через процедуру.

16.  Задано три матриці А(3,3), В(4,3), С(3,2). Знайти значення максимального елемента кожної матриці та його індекси. Знаходження значень, ввід елементів матриці оформити через процедуру.

17.  Задано три масиви А(6), В(7), С(5). Знайти середнє арифметичне додатних елементів кожного масиву відповідно. Знаходження середнього арифметичного, ввід елементів масиву оформити через процедуру.

18.  Задано три матриці А(3,3), В(4,4), С(3,2). Знайти суми елементів кожного парного стовпчика матриць А,В,С відповідно. Знаходження сум, ввід елементів матриці оформити через процедуру.

19.  Задано три матриці А(3,4), В(2,3), С(3,3). Знайти кількість додатних елементів в кожній стрічці матриці і їх загальну кількість для кожної матриці окремо. Знаходження даних, ввід елементів матриці оформити через процедуру.

20.  Задано три матриці А(3,3), В(4,3), С(3,2). Знайти значення мінімального елемента кожної матриці та його індекси. Знаходження значень, ввід елементів матриці оформити через процедуру.

21.  Задано три масиви А(3), В(5), С(6). Знайти суму двох найбільших елементів кожного масиву відповідно. Знаходження значень, ввід елементів масиву оформити через процедуру

22.  Задано три матриці А(4,4), В(3,4), С(4,2). Знайти кількість від’ємних елементів в кожному стовпці матриці і їх загальну кількість для кожної матриці окремо. Знаходження даних, ввід елементів матриці оформити через процедуру.

23.  Задано три матриці А(2,3), В(3,3), С(4,2). Знайти суму найбільшого і найменшого елементів матриць відповідно. Знаходження суми, ввід елементів матриці оформити через процедуру

24.  Задано три масиви А(7), В(6), С(5). Знайти значення максимального по модулю елемента кожного масиву та його номер. Знаходження значень, ввід елементів масиву оформити через процедуру.

25.  Задано три матриці А(4,3), В(2,3), С(3,3). Знайти значення максимального по модулю елемента кожної матриці та його індекси. Знаходження даних, ввід елементів матриці оформити через процедуру.

26.  Задано три масиви А(5), В(8), С(6). Знайти кількість від’ємних елементів кожного масиву відповідно. Знаходження значень, ввід елементів масиву оформити через процедуру.

27.  Задано три масиви А(5), В(8), С(6). Знайти суму двох найбільших по модулю елементів кожного масиву відповідно. Знаходження значень, ввід елементів масиву оформити через процедуру.

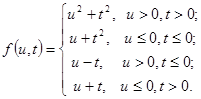
28.  Задано три матриці А(4,4), В(2,3), С(4,3). Знайти суму двох найбільших по модулю елементів матриць відповідно. Знаходження значень, ввід елементів матриці оформити через функції.

29.  Задано три масиви А(3), В(6), С(5). Замінити всі від’ємні елементи їхніми квадратами. Перетворені масиви вивести на екран. Заміну, ввід та вивід елементів масиву оформити через процедуру.

30.  Задано три матриці А(4,4), В(2,3), С(4,3). Замінити всі додатні елементи протилежними. Перетворені матриці вивести на екран. Заміну, ввід та вивід елементів матриці оформити через процедуру

**Завдання 2. Скласти програму обчислення виразу, використовуючи в програмі функцію. За складеною програмою провести обчислення при заданих початкових даних.**

 1. ,

де  a=2.5; b=-7.3.

2.

де  ; a=-2.1.

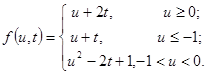
3.  ,

де  x=0.51.

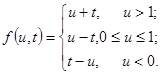
4. ,

де  s=1.783; t=5.71.

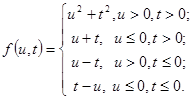
5. ,

де  x=2.31; y=4.2; a=3.1; b=0.02.

6. ,

де  x=; y=0.41; a=0.1; b=-2.1.

7. ,

де  a=-0.5; b=4.2.

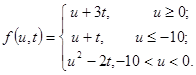
8. ,

де  x=0.31.

9.,

де  x=0.2.

10.,

де  x=-4.21; y=-31.2.

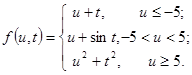
11.,

де  ; b=21.3.

12. ,

де  x=0.3; y=6.1.

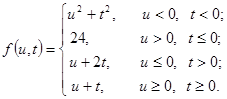
13. ,

де  a=-0.2; b=-0.42.

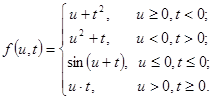
14.,

де  x=0.51; y=0.17.

15. ,

де  x=23.7; y=41.2.

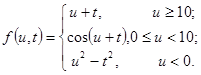
16. ,

де  x=-14.2; y=0.32.

17. ,

де  x=-3.18; y=13.131.

18. ,

де  a=19.2; b=0.48.

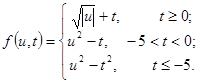
19. ,

де  x=-0.53

20. ,

де  x=0.61.

21. ,

де  a=1.34; b=1.41.

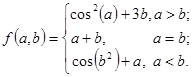
22. ,

де  x=-3.17; y=5.94.

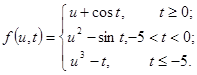
23. ,

де  x=0.47; y=2.13.

24. ,

де  a=1.23; b=4.36.

25. ,

де  a=15.78; b=3.41.

26.,

де  x=11.91; y=3.76.

27.,

де  x=0.53, у=1.72

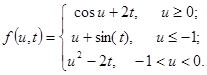
28.,

де  x=1.53; y=4.57.

29.,

де  s=1.8; t=0.57.

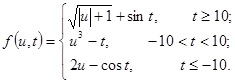
30.,

де  x=1.67; y=14.22; a=0.1; b=1.

31.,

де  x=0.2

32.,

де  a=3.73; b=2.34.

**Завдання 3**

1. Дано квадратну матрицю А розміром nxn. Скласти програму нормування матриці, поділивши вС елементи кожного рядка на максимальний елемент за модулем цього рядка. Пошук максимального елемента рядка оформити у вигляді прцедури.

2. Дано три одновимірні масиви різної розмірності. Скласти прграму формування одновимірного вектора D, елементами якого є відмінні від нуля мінімальні елементи за модулем заданих масивів. Пошук мінімального елемента за абсолютним значенням виконати у вигляді процедури.

3. Оформити у вигляді прграми обчислення функції

ex=1+x+x2/2!+x3/3!+...

і порівняти одержаний результат зі значенням, отриманим при використанні стандартної функції EXP(X), для х=2.3; 8.5.

4. Дано двохвимірний масив. Відсортувати масив по стовпцях у спаданні. Перестановку чисел виконати у вигляді підпрограми.

5. Дано два одновимірні масиви, які складаються не більш як з 30 елементів кожний. Використавши підпрограму пошуку максимального елемента, знайти півсуму максимальних елементів заданих масивів.

7. Дано два трикутники зі сторонами a, b, c i d, e, f. Використавши підпрограму, визначити, чи ці трикутники є прямокутні чи ні?

8. Дано сторони двох трикутників: a, b, c та d, e, f. Знайти, на скільки площа одного трикутника більша від другого. Площа трикутника ABC зі сторонами A, B, C обчислюється за формулою Герона



де R - півпериметр трикутника ABC. Обчислення площі трикутників оформити у вигляді прцедури.

9. Дано три квадратні матриці. Сформувати вектор, який складається з мінімальних елементів кожної матриці, відмінних від нуля. Пошук мінімального елемента виконати у вигляді процедури.

10. Дана прямокутна матриця. Сформувати вектор, який складається з максимальних елементів кожного стовпця (рядка) матриці. Пошук максимального елемента в стовпці (рядку) виконати у вигляді підпрограми.

11. Використовуючи підпрограму обчислення значення однієї функції, обчислити і надрукувати таблицю значень таких функцій:



Аргумент х змінюється від початкового значення 2.8 з кроком 0.15 до кінцевого 3.7.

12. Використовуючи підпрограму, обчислити значення функції 

Для значень х, які містяться в проміжку [-1.5;1.5] з кроком х=0.5, =0, і для значень х, які містяться в проміжку [-2.5;3] з кроком 0.6; =0.1. Передбачити недопустимість ділення на 0.

13. Скласти прграму знаходження суми стовпців матриці розміром mxn рузельтат записати у вигляді вектора.

14. Обчислити значення перших n (n<30) елементів гео-метричної прогресії з допомогою підпрограми, якщо відомий перший елемент прогресії а та знаменник пргресії q. Кожний наступний елемент геометричної прогресії утворюється множенням попереднього на знаменник прогресії. Для розміщення в пам’яті обчислених елементів їх слід оголосити як одновимірний масив.

15. Дано декартові координати 8 точок. Підрахувати, скільки точок належить площині, обмеженій колом з радіусом R і центром в точці (0,0). Провірку належності оформити у вигляді підпрограми.

16. Дано декартові координати 6 точок і координати базової точки. Визначити сумарну віддаль між базовою і рештою точок, а також максимальну віддаль. Обчислення віддалі між двома точками оформити у вигляді процедури.

17. Дано дві квадратні матриці одного порядку А і В. Показати, що для них вірний асоціативний закон, тобто А\*В=В\*А. У випадку стверджувальної відповіді вивести на друк повідомлення “АСОЦІАТИВНИЙ ЗАКОН ВИКОНУЄТЬСЯ”,в противному випадку - “АСОЦІАТИВНИЙ ЗАКОН НЕ ВИКОНУЄТЬСЯ”. Обчислення добутку двох матриць виконати у вигляді підпрограми.

18. Дано три прямокутні матриці. Знайти слід кожної матриці і вивести на друк значення найменшого за модулем сліду. Обчислення сліду матриці оформити у вигляді підпрограми. Слідом матриці називається сума елементів головної діагоналі.

19. Дано три вектори. Визначити максимальний парний елемент кожного вектора і упорядкувати їх за спаданням. Пошук максимального елемента вектора виконати у вигляді підпрограми.

Приклад виконання. Дано два вектори А і В розміром від 1 до 10. Скласти програму обчислення максимального елемента векторів. Знаходження максимального елемента виконати у вигляді процедури.

Розв’язок задачі.

1. Графічний алгоритм розв’язання задачі показаний на рис.10.1, а - головна програма, рис.10.2 - підпрограма.

початок

A(i)

I=1;NA

B(i)

I=1;NB

Maximum (An,A,maxA)

Друк MaxA

Maximum (An,A,maxB)

Друк MaxB

кінець

Рисунок 10.1 – Графічний алгоритм

Max=D(1)

J=1;k

D(i)>max

Max=D(i)

Так

Ні

Рисунок 10.2 – Графічний алгоритм процедури

3. Програма мовою С

# include<conio.h>

# include <stdio.h>

# include <math.h>

float a[20],b[20],maxa,maxb;

int i,j,na,nb;

void max(int nd,float d[],float \*maxd)

{

\*maxd=d[1];

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

if(d[i]>\*maxd) \*maxd=d[i];

}

main()

{

clrscr();

printf("Введіть кількість елементів векторів А і В");

scanf("%d,%d",&na,&nb);

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

{ printf("A(%d)=",i);

scanf("%f",&a[i]);

}

printf("\n");

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

{ printf("B(%d)=",i);

scanf("%f",&b[i]);

}

max(na,a,&maxa);

max(nb,b,&maxb);

printf("Максимальний елемент вектора А=%f\n",maxa);

printf("Максимальний елемент вектора B=%f",maxb);

}

Максимальний елемент вектора А=895

Максимальний елемент вектора B=189

Пояснення до програм

Графічний алгоритм показаний на рис.9.1 На рис.9.2 показаний алгоритм головної програми, який складається з введення елементів векторів А та В, звертання до підпрограми обчислення максимального елемента (MAXD), яке здійснює обчислення максимального елемента вектора А при другому звертанні – максимального елемента вектора В і вивоедення результатів обчислення. На рис.9.2 наведений алгоритм підпрограми обчислення максимального елемента (MAXD). Формальними параметрами підпрограми є вектор D, кількість елементів цього вектора, ND та MAXD.

**Питання для самоперевірки**

1. З якою метою використовуються підпрограми?
2. Що таке формальні і фактичні параметри в процедурах - під-програмах?
3. Як оформляється процедура мовами програмування?
4. Яка різниця між зовнішньою і внутрішньою процедурами?
5. Як формується звернення до підпрограми мовами програмування?

6.Охарактеризуйте відмінності при звертанні до підпрограми мовами програму

7. Де оголошується функції користувача?

8. З чого складається функції користувача?

9. Як передаються дані до функції користувача?

10. Яким чином здіснюється опис двовимірного масиву у функції користувача?

**Лабораторна робота № 11**

**ВИКОРИСТАННЯ СТРУКТУР У МОВІ С**

**Мета роботи:** Набуття навичок у використанні структур при розв’язанні прикладних задач.

**Теоретичні відомості**

Структура — це сукупність різнотипних елементів, яким присвоюється одне ім'я (воно може бути відсутнім), що займає одну ділянку пам'яті. Елементи, що складають структуру, називаються полями.

Змінна типу структура, як і будь-яка змінна, повинна бути описана. Цей опис складається з двох кроків: опису шаблону (тобто складу) або типу структури та опису змінних структурного типу.

Синтаксис опису структури має вигляд:

**struct** [<ім'я структури>]

{ <тип 1> ім'я поля 1;

<тип 2> ім'я поля 2 . . } р1, р2 .. .;

де struct — службове слово;

<ім'я структури> — ім'я типу структура (може бути відсутнім);

<тип 1>, <тип 2> — імена стандартних або визначених типів;

ім'я поля 1, ім'я поля 2,... — імена полів структури;

р1, р2 . . .; — імена змінних типу структура.

**Варіанти завдань**

**Завдання 1.**

1. База даних співробітників фірми містить наступні дані: паспортні дані, освіта, спеціальність, посада, оклад. Описати структуру запису, який би містив вказані дані та організувати збереження записів у файлі. Передбачити процедури додавання, вилучення записів та пошуку записів за довільним запитом.
2. Склад. База товарів, які зберігаються на складі: назва товару, одиниця виміру, кількість. Організувати реєстрацію надходження товарів (формування накладної прийому) і відвантаження (формування накладної видачі). Дані зберігаються в масиві.
3. Довідник меломана. База груп і виконавців; база пісень; база дисків з переліком пісень. Організувати вибір пісень заданої групи та всіх дисків, де зустрічається задана пісня. Дані зберігаються в масиві.
4. Особиста бібліотека. Картотека домашньої бібліотеки: дані книги (автори, назва, видавництво і т.д.), розділ бібліотеки (спеціальна література, хобі, домашнє господарство і т.д.), походження книги і наявність на даний час. Організувати вибір книги за довільним запитом та проведення інвентаризації. Дані зберігаються в масиві.
5. Довідник покупця. База торгівельних підприємств міста: назва, адреса та телефон, спеціалізація, час роботи. Організувати вибір магазину за довільним запитом. Дані зберігаються в масиві.
6. Відділ кадрів. База даних про співробітників фірми: паспортні дані, освіта, спеціальність, посада, оклад. Організувати вибір за довільним запитом. Дані зберігаються в масиві.
7. Довідник студента. База даних вузів: адреса, кількість факультетів, рівень акредитації, рейтинг. Організувати вибір за довільним запитом. Дані зберігаються в масиві, який створюється динамічно.
8. Довідник студента. База даних предметів: назва предмету, кількість годин, викладач, рейтинг. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
9. Довідник студента. База даних – розклад занять, містить п’ять записів з наступними полями: 1-ша пара, 2-га пара, 3-тя пара, 4-та пара, секція. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
10. Довідник студента. База даних – розклад руху маршрутних таксі: номер маршруту, кінцева зупинка, марка автобуса, час поїздки. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
11. Довідник старости. База даних студентів групи: прізвище ім’я по-батькові, рік народження, адреса, рейтинг. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
12. Довідник старости. База даних студентів групи: прізвище ім’я по-батькові, дата останнього дня чергування, стать, адреса. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
13. Телефонний довідник друзів. База даних телефонних номерів друзів: прізвище ім’я по-батькові, рік народження, номер мобільного телефону. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
14. Телефонний довідник кафедри. База даних телефонних номерів співробітників: прізвище ім’я по-батькові, посада, номер стільникового телефону, номер мобільного телефону. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.

**Завдання 2. У кожному із завдань додати поля, які містить дату створення та останнього редагування запису. Бажано кожну із складних характеристик (які складаються з декількох полів, наприклад, дата) описувати за допомогю окремих структур даних і використовувати поляти типу структури.**

1. Довідник автомеханіка. База даних запчастин: назва запчастини, марка автомобіля, рік випуску. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
2. Довідник кухарки. База даних продуктів: назва продукту, одиниці виміру, наявна кількість продукту, вартість одиниці. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
3. Довідник кухарки. База даних страв: назва страви, кількість складових, перелік складових, час приготування. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
4. База даних автовокзалу. База даних про рейси автобусів: номер рейсу, водій, вартість квитка, час відправлення, час прибуття. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
5. База даних поштарки. База даних абонементів: номер будинку, кількість газет, що виписуються, перелік назв газет, що виписуються. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
6. Довідник гравця. База даних ігор: назва гри, вартість гри, жанр, кількість рівнів. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
7. Довідник кондитера. База даних кондитерських підприємств: назва підприємства, адреса, рейтинг, спеціалізація. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
8. Довідник наглядача. База даних злочинців: прізвище, ім’я, по-батькові, рік народження, вид злочину, строк покарання. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
9. Довідник інспектора ДАІ. База даних автомобілів: номер автомобіля, власник, рік випуску, марка автомобіля. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
10. Довідник податківця. База даних підприємств: назва підприємства, рівень оподаткування, рік заснування, власник. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
11. Довідник прибиральниці. База даних містить відомості про аудиторії: номер аудиторії, час останнього прибирання, інтенсивність використання аудиторії, площа аудиторії. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
12. Довідник вихователя. База даних містить відомості про дітей дитячого садка: прізвище, ім’я, по-батькові, стать, адреса, номер телефону мами, номер телефону батька, група. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
13. Довідник покупця ЕОМ. База даних містить відомості про готові комплектації ЕОМ: марка процесора, тактова частота, об’єм оперативної пам’яті, об’єм вінчестера, марка відео карти, ціна. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.
14. Довідник покупця автомобіля. База даних містить відомості про автомобілі: марка автомобіля, колір, рік випуску, новий чи бувший у використанні, ціна. Організувати вибір за довільним запитом. Дані зберігаються в масиві записів, який створюється динамічно.

Приклад виконання

**Завдання**

Задано масив дійсних чисел. Знайти середнє арифметичне додатних чисел масиву.Написати програму для знаходження середнього бала, отриманого студентами в період сесії з дисциплін «Математика», «Фізика» та «Програмування», використовуючи структури.

**Програма**

#include <iostream.h>

#include <string.h>

#include <stdio.h

#include <conio.h>

void main ( )

{

struct stud

{ char fam [20]; int mat, fiz, prg; float sb; } st1, st2;

strcpy (st1.fam, "Кравченко O. С.");

st1.mat = 4;

st1.fiz = 5;

st1.prg = 5;

st1.sb = float (st1.fiz + st1.mat + st1.prg)/3;

st2 = st1;

puts (st2.fam);

cout << st2.mat << st2.fiz << st2.pig << st2.sb << endl;

getch();

}

**Питання для самоперевірки**

1. Що таке структура?

2. Чим відрізняється структура від масиву?

3.Що входить до опису структури?

4. Загальний вигляд опису структур.

5. У яких випадках застосовують структури?

6. До якого типу даних відносять структури (типу значень чи типу посилань)?

7. Як скопіювати поля однієї структури у іншу?

**Лабораторна робота №12**

**РОБОТА З ФАЙЛАМИ МОВОЮ С**

**Мета роботи**: отримання навичок з алгоритмізації та програмування задач з використанням файлових даних; освоїти проектування структури файлу, виведення даних до файлу та читання даних з файлу.

**Файли.**Файл – це обмежений набір даних, що розташовані на зовнішньому носію.

Уфайлі розміщуються дані, що призначені для тривалого зберігання. Кожному файлу дається унікальне ім’я.

Умові С відсутні інструкції для роботи з файлами. Всі необхідні дії виконуються через функції, що включені в стандартну бібліотеку. Вони дозволяють працювати з різними пристроями, такими, як диски, принтер, комунікаційні порти т.д. Ці пристрої сильно відрізняються один від іншого. Однак файлова система дозволяє перетворювати їх у один абстрактний логічний пристрій, що називається потоком. Існує два типа потоків: **текстові і двійкові**.

Перш ніж читати чи записувати інформацію у файл, він повинен бути відкритий. Це можливо зробити за допомогою бібліотечної функції *fopen*. Вона бере зовнішнє ім’я файлу (наприклад C:\MY\_FILE.TXT) і зв’язує його з внутрішнім логічним ім’ям, яке використовується надалі в програмі. Логічне ім’я – це вказівник на потрібний файл. Його необхідно оголосити, і робиться це, наприклад, так:

***FILE****\*lst*; Тут *FILE*– ім’я типу, що описане в стандартному визначенні *stdlo.h*, *1st*– вказівник на файл. Звертання до функції *foреn*у програмі відбувається так: *lst=fopen(ім’я файлу, вид файлу)*;

Ім’я файлу може бути, наприклад: C:\MY\_FILE.TXT – для файлу

MY\_FILE.TXT на диску C:; A:\MY\_DIR\EX2\_3.CPP – для файлу ЕХ2\_З.СРР у піддиректорії A:\MY\_DIR і т.д. Вид файлу може бути:

*r*– відкрити існуючий файл для читання; *w*– створити новий файл для запису (якщо файл з вказаним ім’ям існує, то він буде переписаний) *а*– доповнити файл (відкрити існуючий файл для запису інформації, починаючи з кінця файлу, або створити файл, якщо він не існує); *pb*– відкрити двійковий файл для читання; *wb*– створити двійковий файл для запису; *ab*– доповнити двійковий файл; *rt*– відкрити текстовий файл для читання; *wt*– створити текстовий файл для запису; *at*– доповнити текстовий файл; *r+*– відкрити існуючий файл для запису і читання; *w+*– створити новий файл для запису і читання; *а+*– доповнити або створити файл з можливістю запису і читання; *r+b*

– відкрити двійковий файл для запису і читання; *w+b*– створити двійковий файл для запису і читання; а+b – доповнити двійковий файл з наданням можливості запису і читання.

Якщо режим *t*або *b*не заданий (наприклад, *r*, *w*або *а*), то він визначається значенням глобальної змінної *\_fmode*. Якщо *\_fmode = 0\_BINARY*, то файли відкриваються у двійковому режимі, а якщо *\_fmode =О\_TEXT*– у текстовому режимі. Константи *0\_BINARY*і *O\_TEXT*визначені у файлі *fсntl.h*. Рядки виду *r+b*можна записувати і в іншій формі: *rb+*. Якщо в результаті звертання до функції *foреn*виникає помилка, то вона повертає вказівник на константу NULL. Після закінчення роботи з файлом, він повинен бути закритий. Це робиться за допомогою бібліотечної функції *fclose*. Вона має наступний прототип: *int fclose(FILE \*lst);*

При успішному завершенню функція *fclose*повертає значення нуль. Будьяке інше значення говорить про помилку.

**Виведення інформації до файлу**

*#include <stdio.h>****void****main (****void****) {****char****str[50];*

***FILE****\*rstr, \*wstr, \*pstr, \*astr; rstr =fopen("c:\\my\_file.txt","rt");*

*wstr =fopen("c:\\out\_file.txt","wt"); pstr =fopen("prn","wt");*

*astr =fopen("c:\\out\_plus.txt","at");****while****(fscanf(rstr, "%s", str)!=EOF)*

*{*

*printf("Виведення на дисплей: %S\n", str);*

*fprintf(wstr, "%s\n", str) ;//запис файлу (попередній вміст витирається) fprintf (pstr, "%s\n", str); // виведення на друк*

*fprintf (astr, "%s\n", str); //доповнення файлу*

***}***

*fclose(rstr); fclose(wstr); fclose(pstr); fclose (astr);*

*}*

В даному прикладі вказівники не ініціюються адресами відповідних файлів, що відкриті для вказаного типу операцій. Ім’я “prn”, що використовується для виведення на друк, є стандартним ім’ям друкуючого пристрою.

**Читання рядків із файлу і виведення їх на екран**

*#include <stdio.h>****void****main (****void****)****char****str [50];****FILE****\* fr, \* fw;*

*if ((fr=fopen ("A:\\fail.ttt","r+" ))==NULL)*

//відкривання файлу з дискети

**{**

*printf("Файл не відкрився. \nВведіть інформацію з клавіатури");*

|  |  |
| --- | --- |
| *fgets(str,49,stdin);* | *// можна gets (str ,49);* |
| *}* |  |
| ***else*** | *//або введіть рядок до 49* |
| *fgets (str,49,fr);* |
| *printf ("Виведення рядку: %s", str);* | *//символів без прогалини* |
| ***if****((fw=fopen("a:\\1.txt","w+"))==NULL)* |  |
| *{* |  |
| *printf("Файл не відкрився");* |  |
| *}* |  |
| *else* |  |
| *{* | *//запис до файлу* |
| *printf("\n у файл 1.txt ");* |
| *fputs(str,fw);* | *//функція записує* |
| *return;* | *//вихід із програми* |

*}*// якщо файл не відкрився, то виведення із *str*до файлу помилок *fprintf(stderr,"Виведення в стандартний файл помилок\n%s",str); fclose(fr);*

*fclose(fw);*

*}*

Програма зчитує із файлу *fail.ttt*дискети, в дисководі А: 49 символів або доки не зустрінеться символ кінця рядку. Якщо файл не відкрився, то пропонує ввести інформацію з клавіатури (вводиться 49 символів або до натискання клавіші ENTER). Потім інформація виводиться у файл 1.txt на дискеті або, якщо не вдалось його відкрити, у файл помилок на екрані.

**Завдання 1**

**Варіанти завдань**

1. Створити текстовий файл LR11.txt, що містить цілі числа. Зчитати дані з файлу і занести парні числа в файл LR11\_1.txt, а непарні в файл LR11\_2.txt. Визначити яких чисел більше.

2. Створити файл цілих чисел з N компонент. Вивести на екран суму компонентів файлу, які мають непарні індекси і кратні трьом. Вивести на екран вміст файлу.

3. Створити файл цілих чисел з N компонент. Вивести на екран середнє арифметичне компонент файлу, які більші за 10, але менші за 100. Вивести на екран вміст файлу.

4. Створити файл цілих чисел з N компонент. Вивести на екран мінімальний компонент файлу. Вивести на екран вміст файлу.

5. Створити файл F цілих чисел, що складається з N компонент. Після цього створити файли G та H. У файл G записати всі компоненти файлу F, піднесені до квадрату, а у файл H елементи файлу G, поділені на 2. Вивести на екран вміст файлів F, G, H.

6. Створити файл цілих чисел, що складається з N компонент. Визначити і вивести на екран максимальне з від’ємних чисел у файлі.

7. Створити файл F цілих чисел, що складається з N компонент. Після цього створити файл G. У файл G записати всі парні і нульові числа з файлу F. Вивести на екран вміст файлів F та G.

8. З клавіатури ввести цілі a та b і занести їх в файл F. Дописати в файл значення їх суми, різниці, добутку та цілочисельного залишку від ділення a на b.

9. Створити файл F цілих чисел, що складається з N компонент. Переглянути вміст файлу і числа, кратні 2 записати в файл F2, кратні 3 – в файл F3, кратні 5 – в файл F5, кратні 10 – в файл F10 (використайте оператор case, нічого якщо одне й те ж число буде записане в кілька файлів). Виведіть на екран вміст всіх файлів.

10. Створити текстовий файл L, що містить цілі числа. Зчитати дані з файлу і занести парні числа в файл L1, а непарні в файл L2. Визначити яких чисел більше.

11. Створити файл цілих чисел з N компонент. Вивести на екран суму компонентів файлу, які мають непарні індекси і кратні трьом. Вивести на екран вміст файлу. 225

12. Створити файл цілих чисел з N компонент. Вивести на екран середнє арифметичне компонент файлу, які більші за 2, але менші за 50. Вивести на екран вміст файлу.

13. Створити файл цілих чисел з N компонент. Вивести на екран мінімальний компонент файлу. Вивести на екран вміст файлу.

14. Створити файл F цілих чисел, що складається з N компонент. Після цього створити файли G та H. У файл G записати всі компоненти файлу F, піднесені до квадрату, а у файл H елементи файлу G, поділені на 8. Вивести на екран вміст файлів F, G, H.

15. Створити файл цілих чисел, що складається з N компонент. Визначити і вивести на екран середнє арифметичне від’ємних чисел у файлі.

**Завдання 2.** Передбачити до запис даних у файл так, щоб внаслідок виконання програми файл містив 10-15 екземплярів структурованих даних. Після завершення вводу даних вивести увесь вміст файла на консоль у вигляді таблиці. **Варіанти завдань**

1. Створити файл, який містить дані про авіарейси. Запис містить поля: номер рейсу (рядкова змінна), пункт призначення, час відправлення (рядкова змінна), ціна білета, кількість вільних місць.

2. Створити файл, який містить дані про кількість виробів категорій А, В, С, які зібрані робітниками заводу за місяць. Запис містить поля: прізвище робітника, номер цеху, кількість зібраних за місяць виробів категорії А, кількість зібраних за місяць виробів категорії В, кількість зібраних за місяць виробів категорії С.

3. Створити файл, який містить дані про здачу сесії. Запис містить поля: шифр групи, прізвище студента, назва предмета, оцінка, назва предмета...(всього 5 предметів).

4. Створити файл, який містить дані про продаж комп’ютерної техніки. Запис містить поля: код покупця, назва товару, ціна за одиницю товару, продана кількість.

5. Створити файл, який містить дані про рух пасажирських поїздів від ст. Івано-Франківськ. Запис містить поля: назва кінцевої станції, номер поїзда, час відправлення , наявність місць по категоріях вагонів- СВ, купейний, плацкартний, загальний.

6. Створити файл, який містить дані про книжковий фонд бібліотеки ІФНТУНГу. Запис містить поля: шифр книги, назва, прізвище автора, рік видання.

7. Створити файл, який містить дані про курси обміну валют в банках м. Івано-Франківська. Запис містить поля: назва банка, назва валюти, курс купівлі, курс продажу, назва валюти,...Назви валют: американський долар, марка, фунт, злотий.

8. Створити файл, який містить дані про міжміські автобусні рейси. Запис містить поля: назва міста призначення, час відправлення, ціна білета.

9. Створити файл, який містить дані про автомобілі та їх власників. Запис містить поля: номер автомобіля, марка, колір, рік випуску, прізвище власника.

10. Створити файл, який містить дані про студентів, які мешкають в гуртожитку №. 4. Запис містить поля: прізвище студента, номер кімнати, назва факультету, шифр групи.

11. Створити файл, який містить дані про сплату за електроенергію мешканцями будинку №. Запис містить поля: прізвище господаря, номер квартири, останнє сплачене значення по лічильнику, поточне значення лічильника.

12. Створити файл, який містить дані про наукові конференції. Запис містить поля: назва конференції, місце проведення, дата, вартість публікації, контактний телефон, прізвище контактної особи.

13. Створити файл, який містить дані про туристичні маршрути. Кожен запис містить дані: назва тура, ціна путівки, назва міста, кількість днів перебування в місті, назва міста, кількість днів перебування в місті,...(5 пунктів на маршруті).

14. Створити файл - телефонний довідник організацій міста. Кожен запис містить поля: назва організації, номер телефону, назва вулиці, номер будинку.

15. Створити файл - довідник про дні народження рідних та друзів. Кожен запис містить поля: ім’я, число, місяць, рік народження.

16. Створити файл, який містить дані про книги в книжковому магазині. Запис містить поля: прізвище автора, назва, рік видання, кількість сторінок, ціна.

**Завдання 2**. Скласти програму, яка опрацьовує файл, утворений програмою з завдання 1. Програма повинна виводити у вигляді таблиці увесь вміст файла, далі дати користувачеві змогу вибрати критерій пошуку даних за одним з полів структури у вигляді текстового меню з 3-4 пунктів.

Наприклад, для структури “Книга” можна створити меню: 1 – пошук книг за автором 2. – пошук книг за частиною назви 3 – пошук книг за роком видання 4. – завершити роботу з файлом. Після вибору користувачем відповідного пункту меню, програма повинна запитувати і зчитувати критерій для пошуку, та виводити у вигляді таблиці лише ті дані з файла, які відповідають заданому критерію До меню обов’язково включити пункт пошуку, що сформульовано у варіантах завдання нижче.

**Варіанти завдань**

1. По введених з клавіатури назві пункту призначення вивести у виді таблиці інформацію про всі рейси літаків в заданий пункт, а саме: номер рейсу, час відправлення, ціну білета.

2. Ввести з клавіатури вартість зборки виробів по категоріям (окремо для виробів А, виробів В, виробів С). Вивести інформацію про заробіток робітника , прізвище якого ввести з клавіатури.

3. Вивести прізвища студентів, які отримали “5” з усіх предметів.

4. По введеному з клавіатури коду покупця вивести список куплених ним товарів і загальну суму, на яку покупець купив товар.

5. По введеній з клавіатури назві кінцевої станції та поточний час вивести інформацію про поїзди, які відправляються в заданому напрямку та наявність місць по категоріям вагонів (дані вивести в табличному виді).

6. По введеній з клавіатури назві книги та прізвищу автора вивести повну інформацію про книгу (шифр книги, назва, прізвище автора , рік видання).

7. По назві валюти, введеній з клавіатури, вивести назву банка, в якому курс купівлі валюти найбільший.

8. По введеній з клавіатури назві міста призначення вивести в табличному виді інформацію про рейси в даному напрямку: час відправлення, ціна білета.

9. По введеному з клавіатури року випуску автомобілів вивести повну інформацію про них: номер автомобіля, марка, колір, прізвище власника.

10.По введеному з клавіатури шифру групи вивести в табличному виді повну інформацію про студентів групи, яки проживають в гуртожитку : прізвище студента, номер кімнати, назва факультету.

11.По введеному з клавіатури номери квартири вивести прізвище власника та суму, яку потрібно сплатити за електроенергію.

12.По введеному з клавіатури номеру телефону вивести назву міста, дату проведення конференції та назву.

13.По введеній з клавіатури назві тура роздрукувати ціну путівки, порахувати та роздрукувати вартість одного дня відпочинку на даному маршруті.

14.По введеному з клавіатури номеру (перші дві цифри телефонного номера) вивести в табличному виді дані про абонентів (назва організації, номер телефону, назва вулиці, номер будинку).

15.По введеній з клавіатури назві поточного місяця роздрукувати дані про дні народження в поточному місяці. Якщо дата «кругла» (кількість років ділиться на 5) вивести примітку - текстове повідомлення «Ювілей».

**Завдання 2**

1. Сформувати файл А, що містить інформацію про кінострічки:

- назва фільму;

- рік випуску;

- кіностудія;

- режисер.

З файлу А вибрати і переписати у файл В інформацію про кінострічки, зняті кіностудією ім. Довженка у 1985 - 92 рр.

2. Сформувати два файли - А і В. У файл А записати назви факультетів. У файл В - назви навчальних груп, приналежність їх до факультету, кількість студентів у групі. Використавши сформовані файли, вивести у вигляді таблиці дані про кількість студентів по факультетах.

|  |  |
| --- | --- |
| Факультет | Кількість студентів |
|  |  |

3. Сформувати файл А, який містить бібліографічну ін-формацію:

- автор книги; - назва видавництва;

- назва книги; -обсяг книги (кількість сторінок);

- рік видання.

Переписати у файл В інформацію про книги, видані у видавництві “Дніпро” після 1990 року.

4. Сформувати файл А, що містить інформацію про зарплату співробітників:

- прізвище, ім’я та по-батькові;

- посада;

- посадовий оклад;

- рік народження.

Використовуючи ці дані, визначити, у кого з інженерів найбільша, а у кого найменша зарплата.

5. Сформувати файл, що складається з відомостей про студентів вашої групи: день, місяць, рік народження, місце проживання, стать (Ч або Ж). Використовуючи сформований файл, вивести на друк прізвища студентів чоловічої статі, які проживають в Івано-Франківській області і яким на даний момент виповнилось 19 років.

6. Сформувати файл “К”, що містить інформацію про кубики: розмір кожного кубика (довжина ребра в см), колір (червоний, жовтий, зелений або синій), матеріал (дерево, метал, картон).

Переписати у файл “КІ” дані:

- кількість кубиків кожного з кольорів і їх сумарний об’єм;

- кількість дерев’яних кубіків з ребром 3 см.

7. Сформувати файл “ZAPAS”. Компоненти файлу є відомості про запчастини, їх назва, кількість, вартість, термін зберігання (місяці). Сформувати файл, де помістити відомості про ті запчастини, термін зберігання яких більше року, а також вивести на друк кількість запчастин, вартість яких понад 1000 гр., і їх загальну вартість.

8. Сформувати файл “В”, записами якого є: прізвища письменників, ініціали, назви книги, рік видання. Використовуючи ці данні, сформувати файл “Е”, де розмістити прізвища письменників, які видають більше двох книг за рік.

9. Сформувати файл за результатами сеСї студентів групи: академічна група, прізвище та ініціали, оцінки з 4-х дисциплін. Використавши сформований файл, вивести у вигляді таблиці дані про тих студентів, які навчаються без “незадовільних” оцінок, обчислити середній бал групи.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| П.І.П-б. | Математика | Хімія | Механіка | Історія |
|  |  |  |  |  |

10. Сформувати файл “А” із прізвищ і дат народження ваших друзів. Використовуючи сформований файл, надрукувати прізвища тих, хто народився в літні місяці, вік яких перевищує 20 років.

11. Сформувати файл “С”, в якому записаний текст, слова в тексті розділені крапкою з комою. Знайти кількість слів в тексті. Вивести на друк слова тексту; якщо слово містить букву “у”, то вказати, на яких позиціях позиціях вона знаходиться в слові, при відсутності букви “у” в слові вивести це слово і повідомлення “Букви у в слові немає”.

12. Сформувати файл, який містить інформацію про 10 автолюбителів: прізвище, марка автомобіля, колір, номер. Використовуючи ці дані, надати інформацію про автолюбителів, які мають “Жигулі” білого кольору і їх номер починається з 45.

13. Сформувати файл “В”, що містить інформацію про книги по програмуванню в бібліотеці: прізвище автора, назва видавництва, рік видання, алгоритмічна мова. Записати у файл “ВІ” список книг, виданих видавництвом “Просвіта” після 1990 року, а також кількість книг по мові GW BASIC. Ці дані також вивести на друк.

14. Відомості про власників автомобілів складаються з їхніх прізвищ, марки автомобіля і державного номера автомобіля. Сформувати файл, що містить ці відомості про 12 автомобілів. Вивести на друк номери автомобілів, які починаються з 3 і закінчуються на 5, і кількість автомобілів кожної марки.

15. Сформувати файл, що містить інформацію про книги за таким зразком: прізвище автора, назва, рік видання. Використовуючи сформований файл, знайти назви книг даного автора, рік видання. Прізвище автора ввести з клавіатури.

16. Сформувати файл, що містить інформацію про номери телефонів такого зразка: прізвище власника, його ініціали, номер телефону, адреса. Надрукувати прізвища та ініціали тих, які проживають по вул. Шевченка і їхній номер телефону містить цифру “3”.

17. Сформувати файл “С”, який містить інформацію про телефонні розмови (місто, з яким велась розмова, номер телефону, вартість розмови, час розмови). Сформувати файл абонементів, з якими велась розмова протягом 10 хвилин і більше і витрати на одну розмову складають більше 30 гр. Знайти вартість вСх переговорів.

18. Сформувати файл “ЕХ”, що містить відомості про експорт товарів такого зразка: найменування, країна, обсяг поставок і вартість поставок. Переписати у файл “EXP” назви країн, експорт до яких перевищує 10 найменувань, і загальну вартість таких поставок.

19. Створити файл, який містить відомості про іграшки: назва іграшки, вартість, вікові обмеження дітей, для яких призначена іграшка. Використовуючи ці дані, сформувати файл, що містить назви іграшок, ціна яких не перевищує 20 гр. і підходить дітям до 5 років.

20. В шаховому турнірі беруть участь 10 шахістів. Сформувати файл прізвищ учасників і файл результатів (у вигляді матриці): перемога - 1, нічия - 0.5, поразка - 0. Використовуючи сформовані файли, надрукувати прізвища учасників і кількість набраних ними очків в порядку зайнятих місць.

21. Сформувати файл “NAFTA”, де записати назву родовища, рік відкриття, собівартість тонни нафти, кількість видобутої нафти. Кількість родовищ N. Записати у файл “В1” назву тих родовищ, де собівартість тонни нафти менша С, і це родовище знаходиться в експлуатації більше Т років. Окремо вивести на друк назву родовища, з якого видобуто найбільше нафти.

Приклад виконання. Сформувати файл, що містить інформацію про розклад руху потягів, які відправляються з Івано-Франківського вокзалу: номер потягу, час відправлення, станція призначення, час в дорозі. Використовуючи ці данні, надати інформацію про розклад руху поїздів до станції Київ, які відправляються з 20 до 24 год. і перебувають в дорозі менше 15 год.

До станції Київ

------------------------------------------------------------------------

Номер | Час | Станція призначення | Час

поїзда |відправлення| | у дорозі

------------------------------------------------------------------------

203 21 Київ 12

------------------------------------------------------------------------

1. Графічний алгоритм показаний на рис.13.1, 13.2.

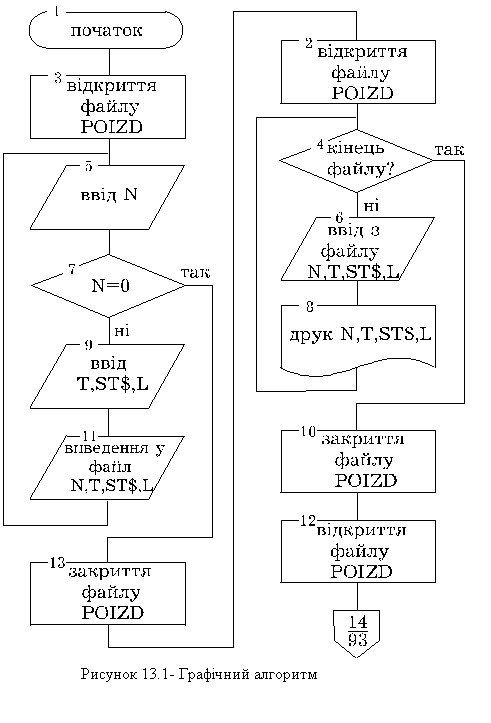
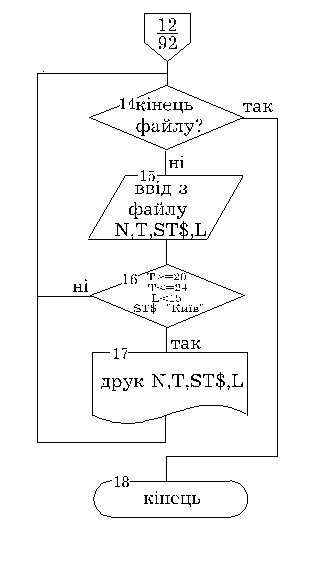


Рисунок 12.1– Графічний алгоритм

 Програма з використанням файлу послідовного доступу

3. Програма мовою С

#include <stdio.h>

#include <string.h>

#define SIZE 256

main( )

{

int n, t, l;

char st[SIZE];

FILE \*fp;

if((fp=fopen(“poizd”,”w”))= =NULL)

{

printf(“FILE NO OPEN\n”);

exit(1);

}

while(1)

{

\*st=‘\0’;

printf(“n>“);

scanf(“%i”,&n);

if (n= =0) break;

printf(“t>“);

skanf(“%i”,&t);

printf(“st>“);

skanf(“%i”,&st);

printf(“l>“);

skanf(“%i”,&l);

fprintf(fp,”%i\n%i\n%s\n%i\n”,n,t,st,l);

}

fclose(fp);

fp=fopen(“poizd”,”r”);

while(!feof(fp))

{

\*st=‘\0’;

fscanf(fp,”%i\n%i\n%s\n%i\n”,n,t,st,l);

printf(“ %i %i %s %i\n”,n,t,st,l);

}

rewind(fp);

printf(“До станції Київ:\n”);

while(!feof(fp))

{

\*st=‘\0’;

fscanf(fp,”%i\n%i\n%s\n%i\n”,n,t,st,l);

if(t>=20 && t<=24 && l<15 && (!strcmp(st,”Київ”)))

printf(“ %i %i %s %i\n”,n,t,st,l);

}

}

123 16 Львів 5

212 15 Москва 19

83 22 Перемишль 8

203 21 Київ 12

129 12 Чернівці 4

До станції Київ:

203 21 Київ 12

Пояснення до програм

Графічний алгоритм розв’язку задачі показаний на рис.13.1. Блок 3 відкриває файл “POIZD”. Після введення номера поїзда (N), блок 5, вводимо з клавіатури параметри:час відправлення T, станція призначення ST$, час в дорозі L. Ці дані записуються у файл “POIZD”, блок 11. Для закінчення введення даних необхідно ввести номер поїзда N=0, що забезпечує закриття файлу.

Для того, щоб одержати роздрук вмісту створеного файлу, організований цикл, де читаються і віддруковуються всі записи, блоки 2, 4, 6, 8, 10. Цикл організований по ідентифікації кінця файлу з використанням оператора WHILE.. і функції EOF.

Для обробки даних необхідно відкрити створений файл і організувати перегляд даних в циклі. При цьому необхідно слідкувати за виконанням умови, що поїзд рухається до станції Київ, відправляється з 20 до 24 години і перебуває в дорозі менше 15 годин, символ 16.

**Питання для самоперевірки**

1. Якими функціями відкривається і закривається файл?
2. Які існують модифікації функції OPEN?
3. Як визначити кількість записів у файлі?
4. Яка відмінність операторів fprintf() і fscanf() ?

**Перелік використаних джерел**

1. Програмування на С(С++). Парадигма процедурного програмування. Навчальний посібник. /Гнатів Б.В., Гнатів Л.Б. . – Львів: Видавництво «Растр-7», 2017. – 264 с.

2. Мова програмування C. Друге видання. Браян В. Кернiган, Денiс М. Рiчi, 2012 р. – 232 с.

3. Програмування та алгоритмічні мови 1. Алгоритмізація та основи програмування: Конспект лекцій / КПІ ім. Ігоря Сікорського; уклад.: І.В. Назарчук. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 140 с.

4. Гаркуша І.М. Конспект лекцій з дисципліни “Програмування”. Частина 1.– Д.: НТУ «ДП», 2020. – 103 с.

5. Гросфельд Ю.О., Морщавка С.В., Мороз Г.В. Методичні вказівки з дисципліни «Прикладне програмування» до лабораторних робіт/ Укл. Г.В. Мороз, Запоріжжя НУ «Запорізька політехніка», 2021 р. – 62 с.

6. Сі підручник. <https://w3schoolsua.github.io/c/index.html#gsc.tab=0>.

7.Програмування С/С++. Основні поняття. <https://www.eolymp.com/uk/blogs/posts/26>

8. Алгоритмічні мови та основи програмування: мова С. В. Ю. Вінник

## <https://programming.in.ua/programming/c-language.html>

9. Михайлюк І.Р. Основи програмування: конспект лекцій. / І. Р. Михайлюк. – Івано-Франківськ : ІФНТУНГ, 2023. - 81 с.