

Ім'я користувача:
приховано налаштуваннями конфіденційності

ID перевірки:
1016359285

Дата перевірки:
15.06.2024 01:47:52 EEST

Тип перевірки:
Doc vs Library + My Database

Дата звіту:
17.06.2024 09:35:51 EEST

ID користувача:
100011372

Назва документа: Курій_Максим, група ТК-22, курсова робота

Кількість сторінок: 18 Кількість слів: 2020 Кількість символів: 13709 Розмір файлу: 909.32 KB ID файлу: 1016163961

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

46.8% Схожість

Найбільша схожість: 40.8% з джерелом з Бібліотеки (ID файлу: 1016154850)

Пошук збігів з Інтернетом не проводився

46.8% Джерела з Бібліотеки

41

Сторінка 20

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

12

Підозріле форматування

6
сторінок

1. Теоретична частина

1.1. Алфавіт мови. Структура програми на мові C++.

Типи даних.

Мови програмування, як і розмовні, мають власний алфавіт – набір допустимих символів. Алфавіт мови складається з:

- великих і малих літер латинського алфавіту A, B, C...Z, a, b, c...z;
- цифр 0, 1, 2... 9;
- спеціальних символів: <<, >>, &, |, ~, ^, <, <=, ==, !=, >, >=, &, +, -, *, /, %, +, --, (,), [,], {, }, =, --, ?, \, \$, #, ', " та символу підкреслення _.

Програми складаються із синтаксичних конструкцій, які називаються командами. Команди бувають з лексем – неподільних елементів мови: слів, чисел, символів операцій.

Слова поділяються на ключові й ідентифікатори. Ідентифікатор – це назва, яку користувач надає об'єктам, наприклад, змінним, сталим, функціям. Зарезервовані ідентифікатори називаються ключовими.

Суттєвою особливістю мови C++ порівняно з іншими мовами є те, що програми складаються з функцій, які відіграють роль підпрограм в інших мовах.

Головна функція, яка має бути у кожній програмі, - це функція вигляду main()

```
{  
опис змінних;  
виконавчі оператори;  
}
```

де main () – заголовок функції

Отже, основними частинами типової структури програми на C++ є такі:

- директиви препроцесорної обробки;
- опис зовнішніх змінних (вихідних даних і результатів) та функцій;
- функції програми;
- головна функція — програми main()

Приклад запису фрагмента простої програми:

```
#include <iostream>
int main() {
    std::cout << "Привіт, студенте! Я C++" << std::endl;
    return 0
}
```

Усі дані, які беруть участь у розв'язуванні задачі, ретельно класифікують за типами. Тип визначає допустимі значення даного, операції, які можна над ними виконувати, й обсяг оперативної пам'яті.

Типи даних:

1. Прості (базові) типи:

- int - цілі числа;
- float - числа з плаваючою точкою;
- double - числа з подвійною точністю;
- char - символи;
- bool - логічний тип (істина/хибність).

2. Модифіковані типи:

- signed та unsigned - змінні знакові та беззнакові типи;
- short та long - змінні коротшого та довшого розміру.

3. Структурні типи:

- struct - структури;
- union - об'єднання;
- enum - перерахування;
- class - класи.

4. Вказівники та посилання:

- вказівники (*);
- посилання (&).

5. Масиви та рядки:

- одновимірні та багатовимірні масиви;
- символьні рядки (масиви символів);

6. Стандартні типи:

- string - рядок;
- vector - динамічний масив;
- map, set та інші контейнери STL.

1.2. Умовний оператор.

Під час написання програми може виникнути потреба трактувати декілька команд як одну. Така команда називається складеною. Складена команда – це конструкція такого вигляду:

```
{  
<команда 1>;  
...  
<команда N>;  
}
```

Кому як команду використовують тоді, коли необхідно інтерпретувати декілька виразів або команд як одне ціле. Вона має вигляд:

```
вираз 1, вираз 2  
або  
команда 1, команда 2
```

Алгоритми, в яких пропонується виконання ті чи інші послідовності дій в залежності від деяких умов, називаються алгоритмами з розгалуженням.

Для реалізації таких алгоритмів в мовах програмування існують спеціальні засоби, а саме умовний оператор і оператор вибору. Кожен умовний оператор передбачає виконання однієї з двох дій в залежності від істинності деякої умови.

Умовний оператор може використовуватися в повній або скороченій формі.

Схема умовного оператора.

```
if (<логічний вираз>)  
{  
    команди, коли_умова_істинна  
}
```

```
else{  
    команди_коли_умова_хибна  
}
```

Коротка команда розгалуження:

```
if (<логічний вираз>  
{  
    команди, коли_умова_істинна  
}
```

Умовою в умовному операторі може бути будь-який вираз логічного типу, тобто вираз, який може приймати одне з двох значень: True, False.

Найпростішими логічними виразами є вирази відношення, в C++ використовуються наступні позначення відношень:

- = – дорівнює;
- <> – не дорівнює
- > – більше;
- < – менше;
- >= – більше або дорівнює;
- <= – менше або дорівнює.

Команда вибору (switch):

```
switch(ЗМІННА){  
    case 1:команди_коли_значення=1;  
    break;  
    case 2:команди_коли_значення=2;  
    break;  
    ...  
    case n:команди_коли_значення=n;  
    break;  
    default:команди_коли_значення_не_відповідає_умовам;  
}
```

Вираз та ознаки мають бути цілочисельні. Оскільки транслятор мови C++ може розглядати значення змінних типу `char` і `enum` як цілі дані, то на місці виразу та ознак можуть бути сталі цілого, символьного або перерахованого типів. На місці команд 1-N може бути одна команда, декілька або не бути жодної команди. Команда `break` не є обов'язковою і записується у разі потреби. Вона слугує для виходу з команди `switch`.

Складова частина `default:команди_коли_значення_не_відповідає_умовам` також може бути відсутня – тоді матимемо коротку форму команди вибору `switch`.

1.3 Організація циклів.

Оператори циклу з передумовою, післяопераційна умова із параметром.

Цикл (повторення) – це процес виконання певного набору команд деяку кількість разів. У мові C++ є три оператори циклу: `for`, `while`, та `do while`.

Команда `for` має вигляд:

```
for ([вир1; вир2; вир3]) <команда 1>;
```

де `вир1` — вираз ініціювання, який призначений для встановлення початкового значення параметра, це вираз присвоювання;

`вир2` — вираз умови, що визначає умову повторення циклу;

`вир3` — вираз ітерації, який визначає крок зміни параметра, що керує циклом, після кожного виконання.

Вирази `вир1`, `вир2` та `вир3` — необов'язкові параметри, які розділяються символом «;».

Дія команди:

- обчислюється значення виразів 1 і 2;
- якщо значення виразу 2 істинне — виконуються оператори циклу. Якщо хибне – виконавець програми переходить до наступної після `for` команди.
- обчислюються значення виразів 3 і 2 і перевіряється попередній пункт.

Оскільки в операторі `for` перевірка виразу-умови відбувається перед циклом, то у випадку помилкової умови цикл може жодного разу не виконуватися.

Оператор `for` може використовувати декілька змінних, що керують циклом, а будь-які вирази можуть бути відсутніми.

Операторам циклів `for` слід віддати перевагу при організації циклів з лічильниками.

Оператор циклу з передумовою `while` виконується, якщо умова перевіряється до початку циклу, і має вигляд:

`while (вираз-умова) оператор;`

де оператор — тіло циклу, що може бути представлено простим або складеним оператором.

Дія команди `while`: якщо значення виразу-умови не дорівнює нулю («істина»), то виконується тіло циклу, а в протилежному випадку, тобто коли значення виразу дорівнює нулю («хибно»), — цикл не працює і керування передається наступному за циклом `while` оператору. Цикл з передумовою може не виконуватися жодного разу.

Оператор `while` зручно застосовувати у випадках, коли кількість ітерацій заздалегідь не відома.

Оператор циклу з післяумовою `do while` звичайно застосовується у випадках, коли тіло циклу виконується хоча б один раз, і має таку форму запису:

`do оператор`

`while (вираз-умова);`

У процесі виконання оператора `do while` спочатку здійснюється вхід до тіла циклу і виконується оператор, що являє собою тіло циклу (цей оператор може бути простим або складеним); далі перевіряється вираз `i`, якщо він правдивий («істина»), — цикл повторюється, а коли вираз помилковий («неправда») — здійснюється вихід з циклу.

1.4 Масиви.

Правила опису та роботи з одновимірними масивами.

Масив — це впорядкований скінченний набір даних одного типу. Якби зберігаються в послідовно розташованих комітках оперативної пам'яті і мають спільну назву. Назву масиву надає користувач.

Масив складається з елементів. Кожен елемент має індекси, за якими його можна знайти у масиві. Кількість індексів визначає розмірність масиву. Розрізняють одно- і багатовимірні масиви. Наприклад, двовимірний масив — це прямокутна таблиця, яка складається з декількох рядків і стовпців. Елемент, розташований на перетині i -го рядка і j -го стовпця масиву A , позначають $a[i, j]$. Головну діагональ утворюють елементи двовимірного масиву, для яких $i = j$.

Одновимірний масив — це пронумерована послідовність значень одного типу. Загальний вигляд конструкції опису одномірного масиву такий:

```
<тип><ім'я масиву>[<розмір>]
```

Розмір — це кількість елементів масиву. Розмір масиву необхідно знати і задавати заздалегідь, оскільки компілятор має зарезервувати для нього необхідний обсяг пам'яті. Розміром може бути лише стала величина (не змінна).

Ім'я масиву у програмі змінювати неможна — це стала величина, яка містить адресу першого елемента. Отже, назва масиву з вказівником на перший елемент.

Звернутися до елементів масиву можна двома способами: за допомогою імені масиву або використовувати вказівники. Нумерація елементів масиву завжди починається з нуля. Щоб звернутися до деякого елемента, необхідно зазначити ім'я масиву, а у квадратних дужках — його номер.

Сталі масиви треба ініціалізувати під час оголошення, інакше елементам масиву автоматично будуть присвоєні нульові значення.

Для опрацювання елементів масиву найчастіше використовують команду циклу `for`, хоча можна застосовувати і `while` або `do-while`.

1.5. Символьні величини

На відміну від інших мов програмування у C++ не визначено спеціального типу для опрацювання рядків. Рядок символів розглядається як масив елементів типу `char`, який закінчується символом `'\'`, що є ознакою кінця рядка. Такі рядки називають ASCII-рядками. Сталі типу рядок записують у лапках. У сталих рядках нуль-символ дописується автоматично.

Масиви символів оголошують так:

```
char<назва рядка>[довжина рядка];
```

Під час оголошення символьного масиву необхідно до фактичної довжини рядка додати одиницю до нульового символу (але не у всіх компіляторах). Якщо масив символів оголошують й ініціалізують одночасно, то довжину можна не зазначати, компілятор визначить її. Оскільки рядки є масивами символів, то назва рядка є вказівником на його перший елемент.

Увести весь масив символів можна за допомогою команди

```
cin<<<назва масиву>;
```

вивести значення рядка на екран можна за допомогою команди

```
cout<<<назва рядка>;
```

Для символьних змінних існує чимало функцій, наприклад, `sizeof()` для визначення розміру, `size()` для отримання кількості символів, `strlen()` для визначення довжини рядка та інші.

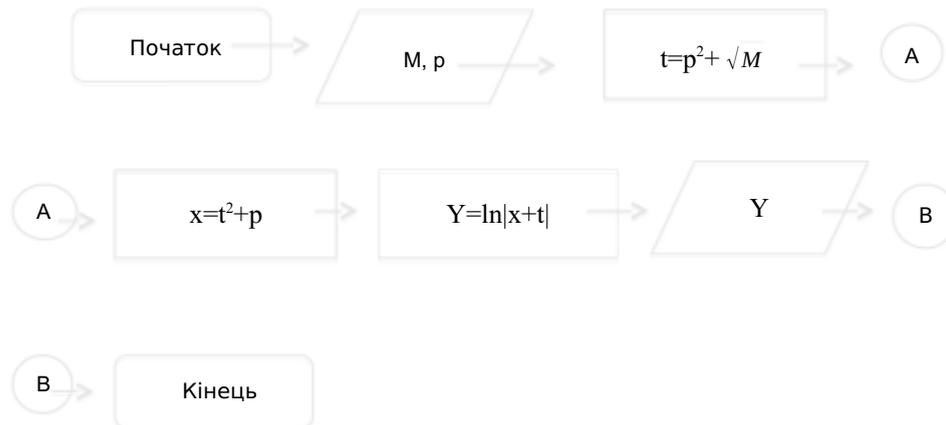
2. Індивідуальні завдання

Завдання 1

Скласти блок-схему лінійного алгоритму обчислення функції, програму на мові C++ та реалізувати її:

$$Y = \ln|x+t|, \quad x = t^2 + p, \quad t = p^2 + \sqrt{M}, \quad \text{де } M = 3.8; \quad p = 2$$

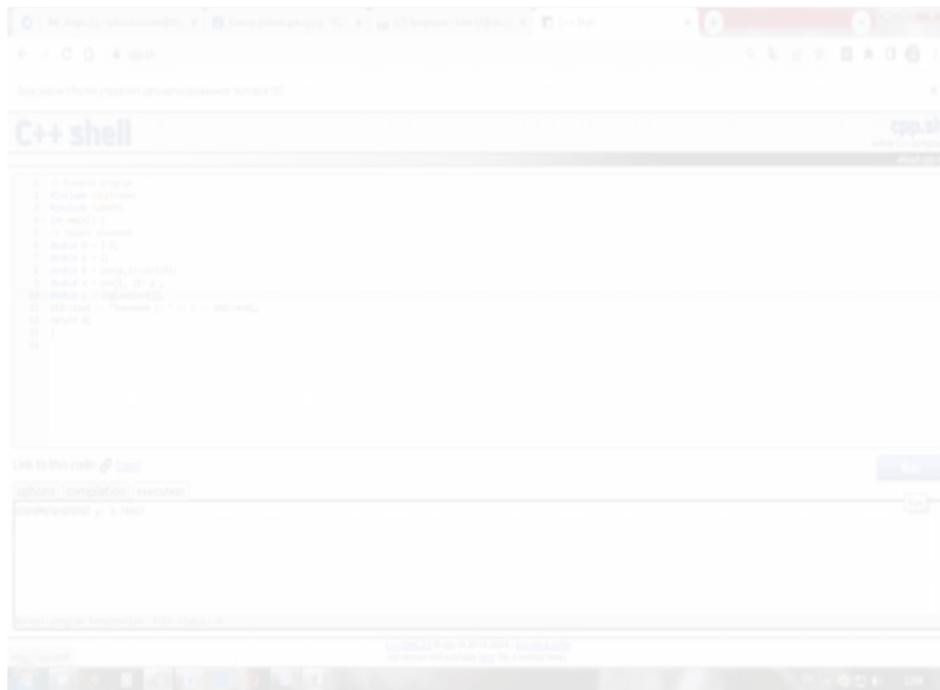
Блок-схема алгоритму:



Програмний код та результат:

```

#include <iostream>
#include <cmath>
int main() {
    // Задані значення
    double M = 3.8;
    double p = 2;
    double t = pow(p,2)+sqrt(M);
    double x = pow(t, 2)+ p ;
    double y = log(abs(x+t));
    std::cout << "Значення y: " << y << std::endl;
    return 0;
}
  
```



The screenshot shows a web browser window with a C++ online compiler. The page title is "C++ shell" and the URL is "cpp.sh". The code editor contains the following C++ code:

```
1 // Простий алгоритм
2 #include <iostream>
3 #include <string>
4 #include <string>
5 #include <string>
6 #include <string>
7 #include <string>
8 #include <string>
9 #include <string>
10 #include <string>
11 #include <string>
12 #include <string>
13 #include <string>
14 #include <string>
15 #include <string>
16 #include <string>
17 #include <string>
18 #include <string>
19 #include <string>
20 #include <string>
21 #include <string>
22 #include <string>
23 #include <string>
24 #include <string>
25 #include <string>
26 #include <string>
27 #include <string>
28 #include <string>
29 #include <string>
30 #include <string>
31 #include <string>
32 #include <string>
33 #include <string>
34 #include <string>
35 #include <string>
36 #include <string>
37 #include <string>
38 #include <string>
39 #include <string>
40 #include <string>
41 #include <string>
42 #include <string>
43 #include <string>
44 #include <string>
45 #include <string>
46 #include <string>
47 #include <string>
48 #include <string>
49 #include <string>
50 #include <string>
51 #include <string>
52 #include <string>
53 #include <string>
54 #include <string>
55 #include <string>
56 #include <string>
57 #include <string>
58 #include <string>
59 #include <string>
60 #include <string>
61 #include <string>
62 #include <string>
63 #include <string>
64 #include <string>
65 #include <string>
66 #include <string>
67 #include <string>
68 #include <string>
69 #include <string>
70 #include <string>
71 #include <string>
72 #include <string>
73 #include <string>
74 #include <string>
75 #include <string>
76 #include <string>
77 #include <string>
78 #include <string>
79 #include <string>
80 #include <string>
81 #include <string>
82 #include <string>
83 #include <string>
84 #include <string>
85 #include <string>
86 #include <string>
87 #include <string>
88 #include <string>
89 #include <string>
90 #include <string>
91 #include <string>
92 #include <string>
93 #include <string>
94 #include <string>
95 #include <string>
96 #include <string>
97 #include <string>
98 #include <string>
99 #include <string>
100 #include <string>
```

The output window shows the result of the compilation: "Compilation successful".

Завдання 2

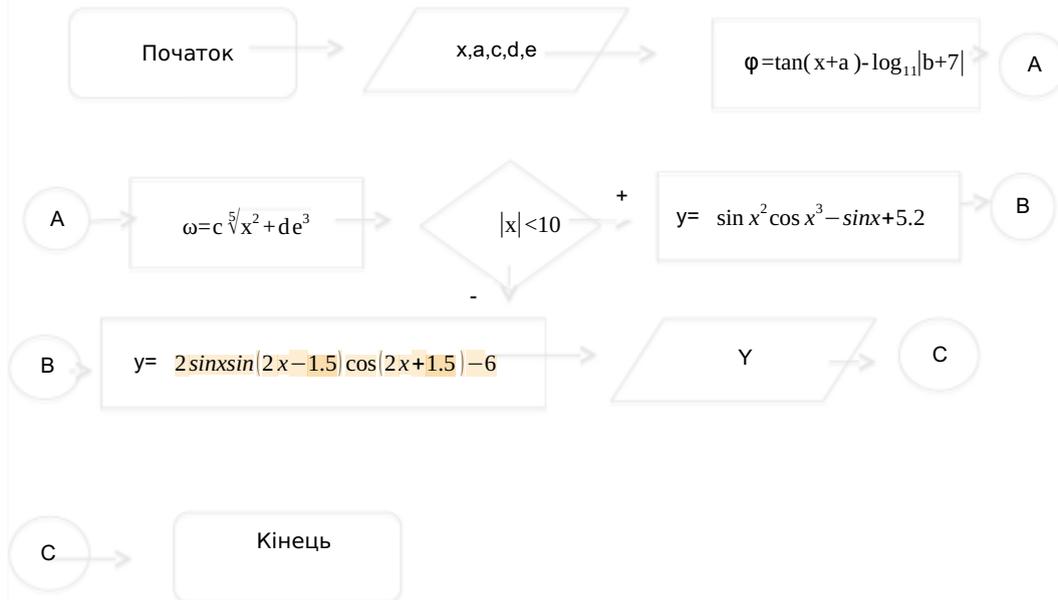
Скласти блок-схему розгалуженого алгоритму, програму на мові C++ та реалізувати її. Увести довільне значення x та обчислити значення функції.

$$\varphi = \tan(x+a) - \log_{11}|b+7|; \omega = c\sqrt[5]{x^2 + de^3};$$

$$y = \begin{cases} \sin x^2 \cos x^3 - \sin x + 5.2, & \text{при } |x| < 10 \\ 2 \sin x \sin |2x - 1.5| \cos |2x + 1.5| - 6, & \text{при } |x| \geq 10 \end{cases}$$

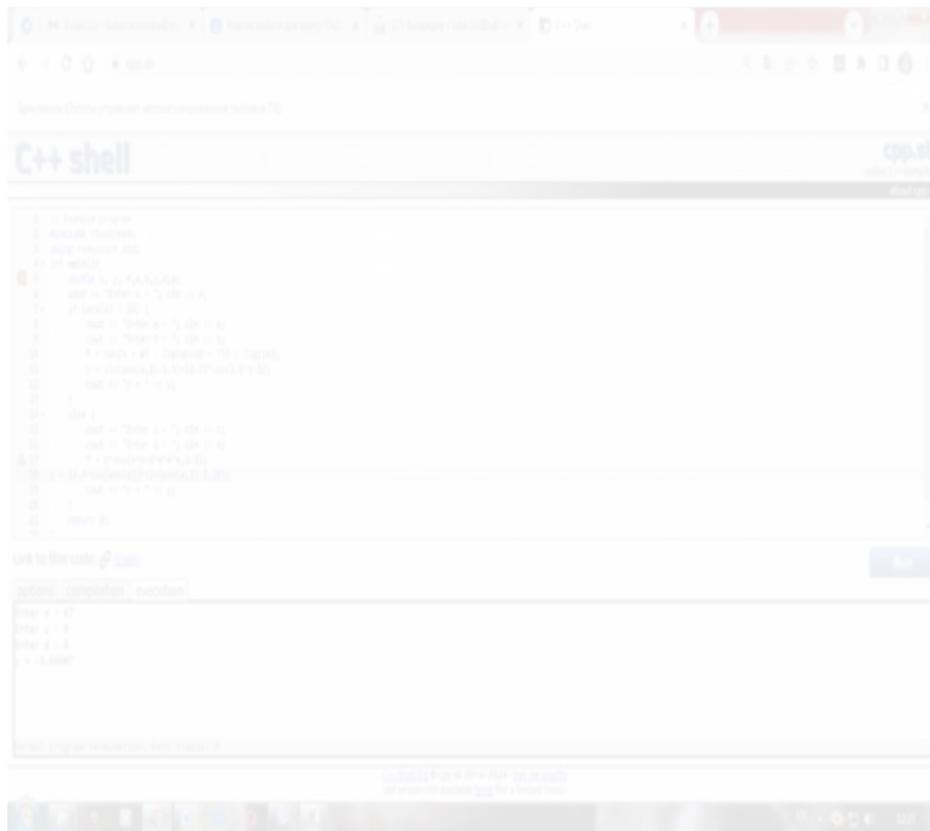
Вхідні дані (x, a, b, c, d) ввести з клавіатури довільні, але такі, щоб значення x задовольняло обидва випадки умови розгалуження. Вивести на екран результати обчислень.

Блок-схема алгоритму:



Програмний код та результат:

```
#include <iostream>
using namespace std;
int main(){
    double x, y, f,a,b,c,d,e;
    cout << "Enter x - "; cin >> x;
    if (abs(x) < 10) {
        cout << "Enter a - "; cin >> a;
        cout << "Enter b - "; cin >> b;
        f = tan(x + a) - log(abs(b + 7)) / log(16);
        y = sin( pow(x,2)/1.5)+11.73*cos(1.6*x-1);
        cout << "y = " << y;
    }
    else {
        cout << "Enter c - "; cin >> c;
        cout << "Enter d - "; cin >> d;
        f = c*pow(x*x+d*e*e*e,1/5);
        y = 13.4*cos(abs( x)*sin(pow(x,2)-2.25));
        cout << "y = " << y;
    }
    return 0;
}
```



The screenshot shows a web-based C++ shell interface. The main area contains the following C++ code:

```
1 // Simple program
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int n, j, k, a, b, c;
7     cout << "Enter n = "; cin >> n;
8     if (n <= 0)
9     {
10        cout << "Enter n = "; cin >> n;
11        cout << "Enter n = "; cin >> n;
12    }
13    k = n; a = 0; b = 0; c = 0;
14    j = 0; while (j <= n)
15    {
16        cout << "Enter k = "; cin >> k;
17        cout << "Enter k = "; cin >> k;
18        k = j * k * k * k;
19        j = j + 1; while (j <= n)
20        {
21            cout << "j = " << j << " k = " << k << endl;
22            cout << "j = " << j << " k = " << endl;
23        }
24    }
25    return 0;
26 }
```

Below the code, there is a section for execution results:

Link to this code [code](#)

options compilation execution

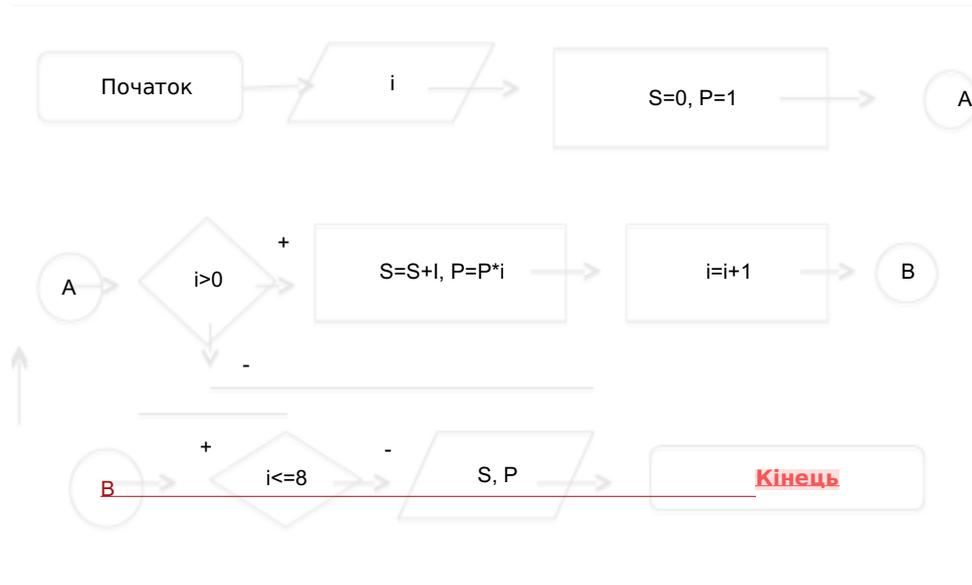
```
Enter n = 10
Enter k = 1
Enter k = 1
j = 1.00000
```

At the bottom of the interface, there are links for "C++ Shell" and "C++ Compiler".

Завдання 3

Розробка та програмна реалізація циклічного алгоритму. При виконанні даного завдання передбачити використання двох із трьох операторів циклу (*for*, *while* або *do-while*). Обчислити суму та добуток перших восьми додатніх чисел.

Блок-схема алгоритму:



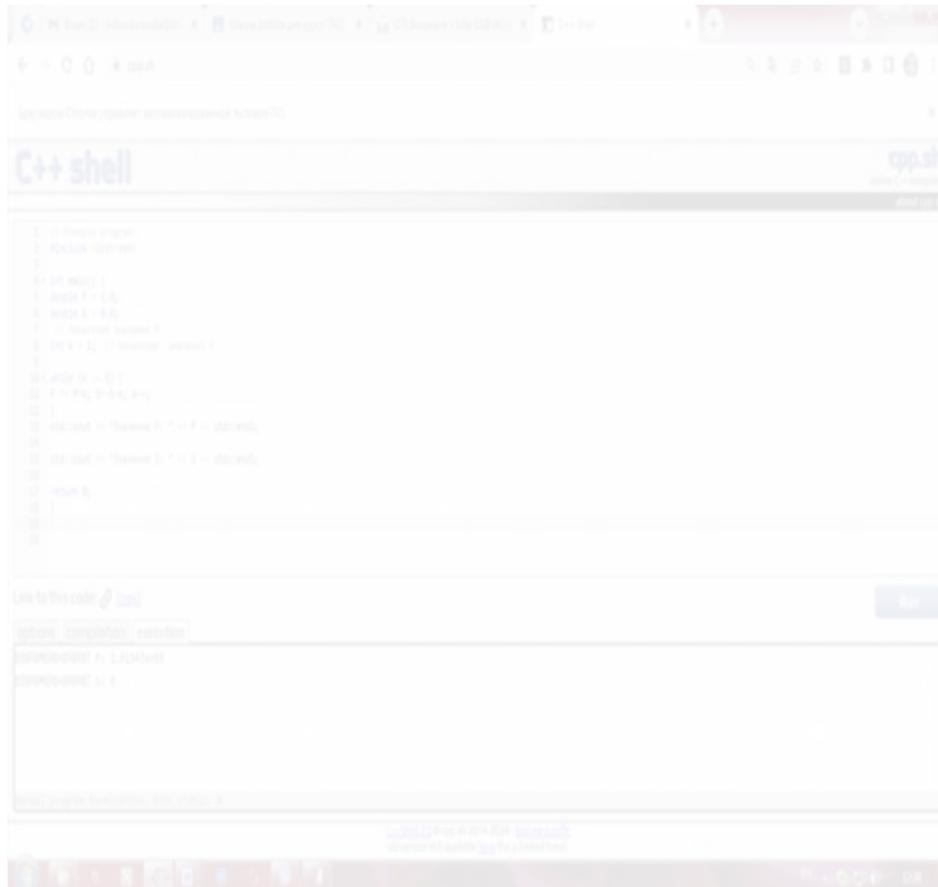
Програмний код та результат:

```

#include <iostream>
int main() {
    double P = 1.0;
    double S = 0.0;
    // початкове значення P
    int k = 1; // початкове значення k
    while (k <= 8) {
        P *= P*k; S*=S+k; k++;
    }
    std::cout << "Значення P: " << P << std::endl;
}
  
```

15

```
std::cout << "Значення S: " << S << std::endl;  
return 0;  
}
```



```
1 // Введіть програму  
2 // Введіть заголовки  
3  
4 #include <iostream>  
5 using namespace std;  
6  
7 // Введіть функцію f  
8 int f(int x) { // Введіть функцію f  
9  
10 int y = x + 1;  
11 f = f(x, y);  
12  
13 std::cout << "Функція f: " << f << std::endl;  
14  
15 std::cout << "Функція g: " << g << std::endl;  
16  
17 return 0;  
18 }  
19  
20
```

Link to this code: [Link](#)

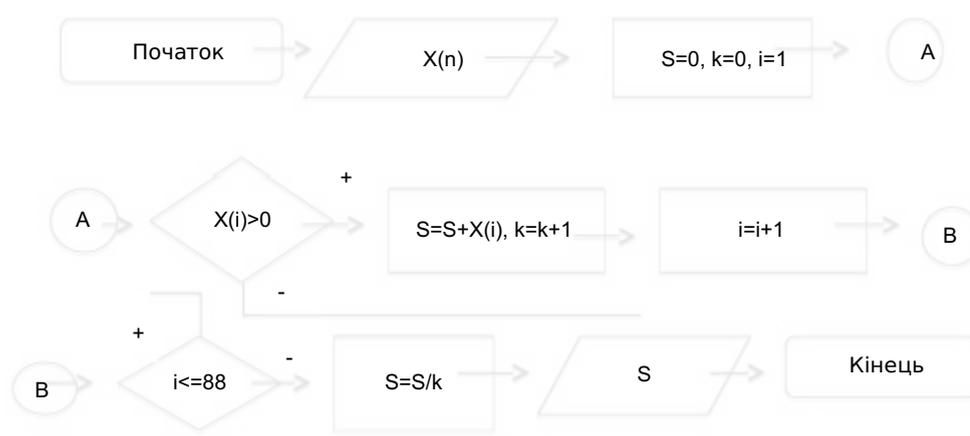
options compilation execution

```
COMPILATION: 0.131040s  
COMPILATION: 0.131040s
```

stderr: stderr: compilation: 0.131040s

Завдання 4

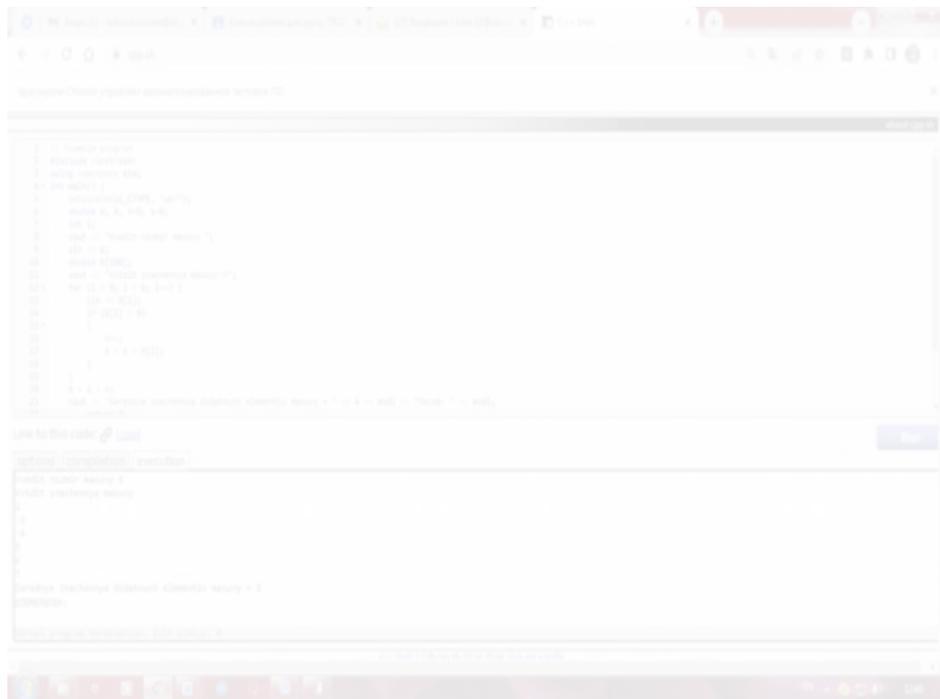
Дано одновимірний масив дійсних чисел X та дійсне число a . Скласти програму обробки масиву. Ввести елементів масиву X довільного розміру. В програмі передбачити виведення масиву та результатів обробки масиву на екран. Знайти середнє арифметичне додатних елементів масиву.

Блок-схема алгоритму:**Програмний код та результат:**

```

#include <iostream>
using namespace std;
int main() {
    setlocale(LC_CTYPE, "ukr");
    double b, A, n=0, s=0;
    int i;
    cout << "Vvedit rozmir masuvy ";
    cin >> b;
    double X[100];
    cout << "Vvtdit znachennya masuvy\n";
    for (i = 0; i < b; i++) {
        cin >> X[i];
    }
}
  
```

```
        if (X[i] > 0)
        {
            n++;
            s = s + X[i];
        }
    }
    A = s / n;
    cout << "Serednye znachennya dodatnuch elementiv masovy = " << A << endl
    << "Масив: " << endl;
    return 0;
}
```

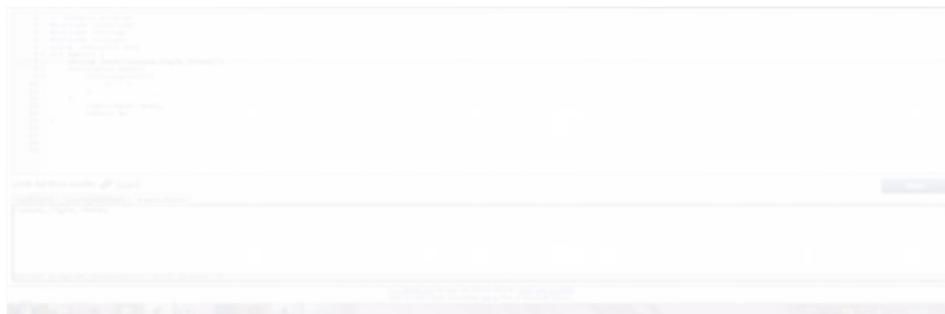


Завдання 5

Створити програмний код обробки символьних величин на м. С++. Вивести на екран результати виконання програми. У введеному рядку замінити всі цифри на знаки оклику.

Програмний код алгоритму та результат:

```
#include <iostream>
#include <string>
#include <cctype>
using namespace std;
int main() {
    string text="1sdsda,2fgfb,3hhhbc";
    for(char&c:text){
        if(isdigit(c)){
            c='!';
        }
    }
    cout<<text<<endl;
    return 0;
}
```



Схожість

Джерела з Бібліотеки

41

1	Студентська робота	ID файлу: 1016154850	Навчальний заклад: Lviv Polytechnic National University	3 Джерело	40.8%
2	Студентська робота	ID файлу: 1016147759	Навчальний заклад: Lviv Polytechnic National University	4 Джерело	12.4%
3	Студентська робота	ID файлу: 1016147764	Навчальний заклад: Lviv Polytechnic National University	3 Джерело	11.1%
4	Студентська робота	ID файлу: 1016154849	Навчальний заклад: Lviv Polytechnic National University		10.5%
5	Студентська робота	ID файлу: 1016154858	Навчальний заклад: Lviv Polytechnic National University		9.41%
6	Студентська робота	ID файлу: 1016147765	Навчальний заклад: Lviv Polytechnic National University		9.31%
7	Студентська робота	ID файлу: 1016147762	Навчальний заклад: Lviv Polytechnic National University		8.76%
8	Студентська робота	ID файлу: 1016163964	Навчальний заклад: Lviv Polytechnic National University	2 Джерело	8.71%
9	Студентська робота	ID файлу: 1016147758	Навчальний заклад: Lviv Polytechnic National University		8.66%
10	Студентська робота	ID файлу: 1016163955	Навчальний заклад: Lviv Polytechnic National University		7.87%
11	Студентська робота	ID файлу: 1016147761	Навчальний заклад: Lviv Polytechnic National University		7.67%
12	Студентська робота	ID файлу: 1016154846	Навчальний заклад: Lviv Polytechnic National University		6.39%
13	Студентська робота	ID файлу: 1016154851	Навчальний заклад: Lviv Polytechnic National University		5.3%
14	Студентська робота	ID файлу: 1016147760	Навчальний заклад: Lviv Polytechnic National University		4.65%
15	Студентська робота	ID файлу: 1008295687	Навчальний заклад: National Technical University of Ukraine	5 Джерело	2.38%
16	Студентська робота	ID файлу: 1000329846	Навчальний заклад: Lviv Polytechnic National University		0.5%
17	Студентська робота	ID файлу: 1004076496	Навчальний заклад: Lviv Polytechnic National University	7 Джерело	0.45%
18	Студентська робота	ID файлу: 1003773011	Навчальний заклад: National University of Water Management	2 Джерело	0.4%
19	Студентська робота	ID файлу: 1015090466	Навчальний заклад: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.4%
20	Студентська робота	ID файлу: 1015677506	Навчальний заклад: National University Ostroh Academy	2 Джерело	0.4%

21	Студентська робота	ID файлу: 1003349622	Навчальний заклад: National Aviation University	0.4%
----	--------------------	----------------------	---	------