

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут телекомунікацій, радіоелектроніки та електронної техніки
кафедра «Радіоелектронні пристрої та системи»



Звіт з лабораторної роботи №5

з дисципліни «Програмування»

Підготував:
ст. групи АП-11
Білоніжко В. А.

Прийняла:
Чайковський І. Б.

Тема:

Базові алгоритмічні задачі.

Мета:

ознайомитися з базовими алгоритмічними задачами, навчитися будувати блок-схеми їх розв'язку.

Теоретичні відомості:

Вкладені цикли. Цикл, до складу якого не входять інші цикли, називається простим. При розв'язанні задач може виникнути необхідність організувати цикл усередині циклу. Якщо до складу циклу входить інший цикл, то говорять про пару вкладених циклів. При цьому перший цикл називається зовнішнім, а вкладений у нього – внутрішнім. Кожний з пари вкладених циклів має свою керуючу змінну і свої параметри.

При виконанні вкладених циклів діє правило: у першу чергу завжди виконується самий внутрішній цикл. Таким чином, для кожного значення керуючої змінної зовнішнього циклу керуюча змінна внутрішнього циклу послідовно пробігає усі свої значення. Усередині вкладеного циклу може знаходитися ще один вкладений цикл і т.д

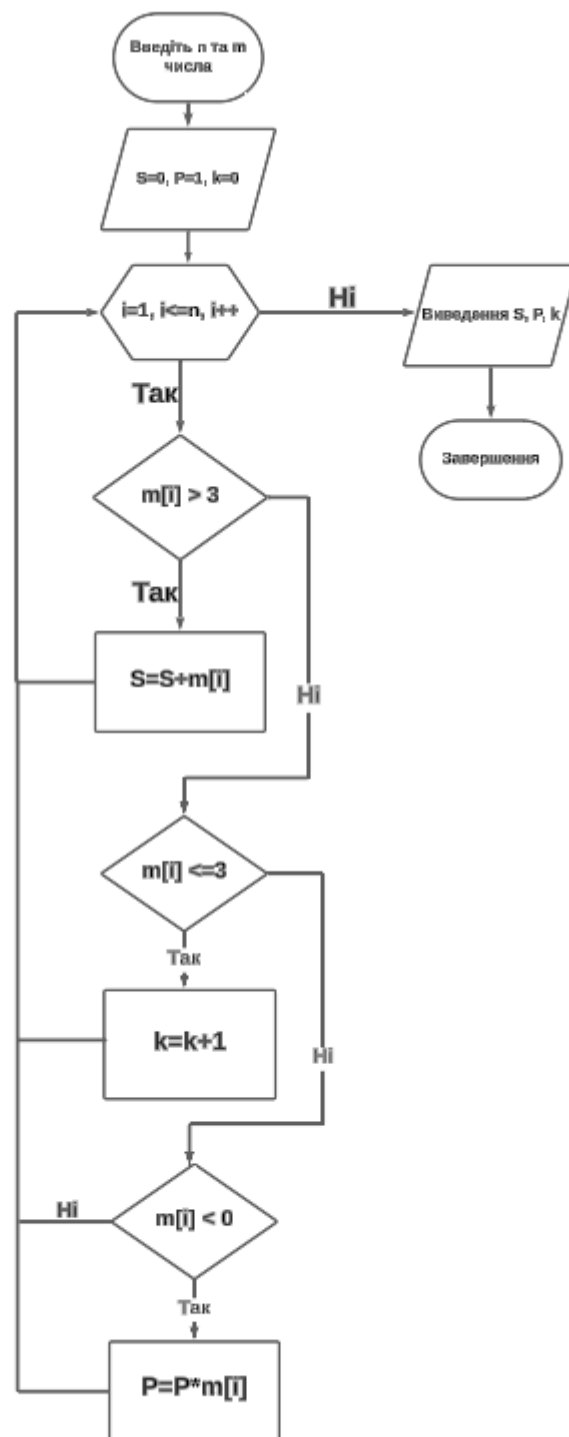
Хід роботи:

1. Ознайомитися з теоретичними відомостями.
2. Здійснити побудову блок-схеми обчислення суми/добутку послідовності чисел для наступного завдання: «Ввести n чисел. Обчислити: S – суму тих з них, що більше 3, P – добуток тих, які менше 0, і k – кількість чисел, менших або рівних 3».

Примітка. Вхідні дані в цій задачі: n – кількість чисел і самі числа.

3. Оформити звіт.

Виконання роботи:



Контрольні запитання:

1. Приклад зовнішнього і внутрішнього циклу (мова C):

```
#include <stdio.h>
int main() {
    int i, j;
    for (i = 0; i < 3; i++) { // зовнішній цикл
        for (j = 0; j < 2; j++) { // внутрішній цикл
            printf("(%d, %d)\n", i, j);
        }
    }
    return 0;
}
```

2. Приклад вкладеного циклу (мова C):

```
#include <stdio.h>
int main() {
    int i, j;
    for (i = 0; i < 3; i++) {
        printf("Outer loop: %d\n", i);
        for (j = 0; j < 2; j++) {
            printf("Inner loop: %d\n", j);
        }
    }
    return 0;
}
```

3. Суть обробки одномірних масивів (мова C):

При обробці одномірного масиву в мові C кожен елемент масиву послідовно обробляється за допомогою індексів. Можна використовувати цикл `for` або `while` для проходження крізь кожен елемент масиву і виконання необхідних операцій з ним.

4. Суть обробки двовимірних масивів (мова C):

При обробці двовимірного масиву в мові C потрібно використовувати вкладені цикли. Один цикл буде відповідати за ітерацію крізь ряди, а інший - за ітерацію крізь стовпці. Кожен елемент масиву можна звертати за допомогою пари індексів (ряд і стовпець).

5. Принцип використовується при обчисленні суми/добутку (мова C):

У мові C при обчисленні суми або добутку значень у масиві кожен елемент масиву додається до (або множиться на) попередній результат. Наприклад, при обчисленні суми:

```
#include <stdio.h>
int main() {
    int array[5] = {1, 2, 3, 4, 5};
    int sum = 0;
```

```
int i;  
for (i = 0; i < 5; i++) {  
    sum += array[i];  
}  
printf("Sum: %d\n", sum);  
return 0;  
}
```