

# **НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

## **ЛАБОРАТОРНА РОБОТА №4**

### **Структури повторення**

Виконала:  
студентка групи ІПЗ-210076  
Соколовська Софія

## ВАРІАНТ – 18

**Тема:** Структури повторення.

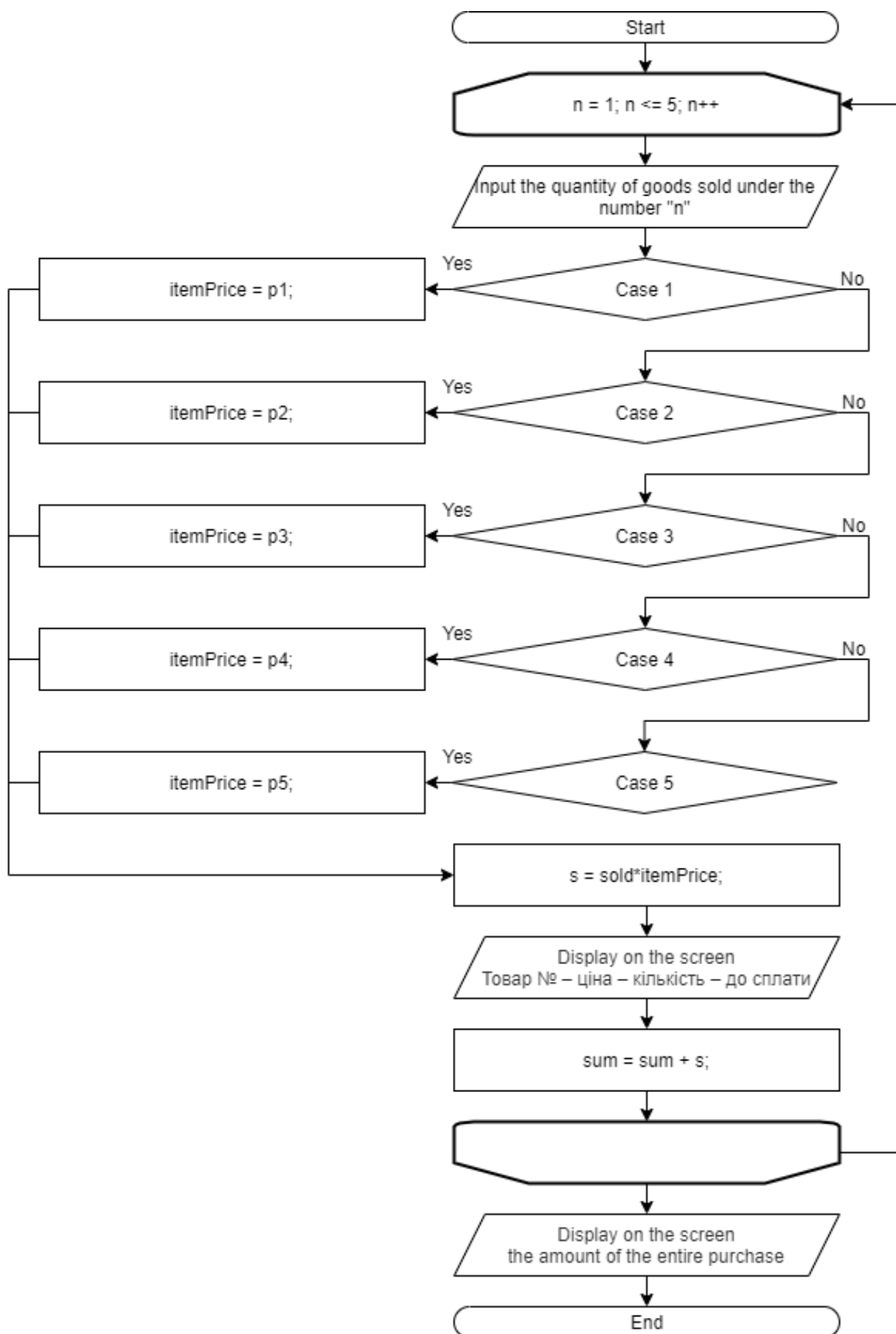
**Мета:** Опанування практичними навичками розробки програм із використанням структур повторення: for, while, do/while.

### Завдання 1

Доробити програму №1 з лабораторної роботи №3 таким чином, щоб на початку роботи користувач міг ввести кількість придбаних товарів, та отримати чек у такому вигляді:

Товар №	Ціна	Кількість	До сплати
...	...	...	...
Загальна сума покупки: ... грн.			

- 1) На рис. 1.1 наведена блок-схема алгоритму програми, яка виводить таблицю на екран та виконує дії.



2) Код програми:

```
#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main()

{

    setlocale(LC_ALL, "Ukr");

    float p1 = 29.80;

    float p2 = 45.50;

    float p3 = 9.98;

    float p4 = 44.90;

    float p5 = 67.80;

    float s;

    float sum = 0;

    int sold;

    float itemPrice;

    int n;

    for(n = 1; n <= 5; n++){

        printf("\nВведіть кількість проданого товару №%i\n", n);

        scanf("%i", &sold);

        switch (n){

        case 1:

            itemPrice = p1;

            break;
```

```
case 2:

    itemPrice = p2;

    break;

case 3:

    itemPrice = p3;

    break;

case 4:

    itemPrice = p4;

    break;

case 5:

    itemPrice = p5;

    break;

}

s = sold*itemPrice;

printf("\nТовар №%i - Ціна: %0.2f ГРН - Кількість: %i - До сплати: %0.2f
ГРН\n", n, itemPrice, sold, s);

sum = sum + s;

}

printf("\nЗагальна сума покупки: %0.2f ГРН", sum);

}
```

3) На рис 1.2 наведений результат програми.

```
Введіть кількість проданого товару №1
1
Товар №1 - Ціна: 29,80 ГРН - Кількість: 1 - До сплати: 29,80 ГРН

Введіть кількість проданого товару №2
23
Товар №2 - Ціна: 45,50 ГРН - Кількість: 23 - До сплати: 1046,50 ГРН

Введіть кількість проданого товару №3
4
Товар №3 - Ціна: 9,98 ГРН - Кількість: 4 - До сплати: 39,92 ГРН

Введіть кількість проданого товару №4
17
Товар №4 - Ціна: 44,90 ГРН - Кількість: 17 - До сплати: 763,30 ГРН

Введіть кількість проданого товару №5
28
Товар №5 - Ціна: 67,80 ГРН - Кількість: 28 - До сплати: 1898,40 ГРН

Загальна сума покупки: 3777,92 ГРН
```

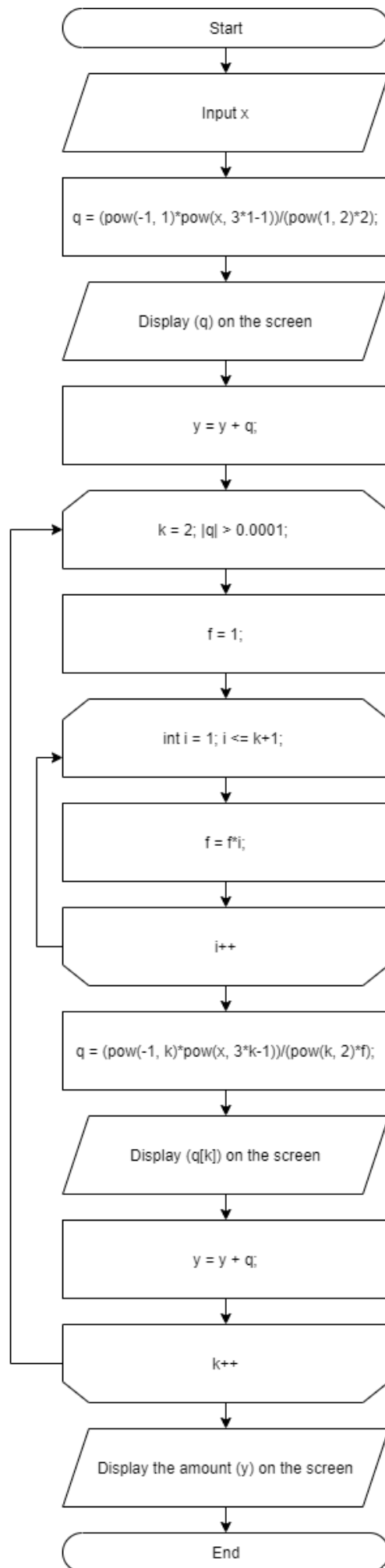
## Завдання 2

- Написати програму для обчислення суми нескінченного ряду, підсумовуючи члени ряду, значення яких за модулем перевищують задану точність  $s = 10^{-4}$ .
- Визначити кількість доданків.
- Обчислення виконати для  $x$  ( $-2 < x < 2$ ).
- Створити варіанти програм, що будуть містити різні структури повторення (for, while, do/while).

$$f(x) = \sum_{k=1}^{\infty} \frac{(-1)^k x^{3k-1}}{k^2(k+1)!}$$

## СПОСІБ – I

- 1) На рис. 1.1 наведена блок-схема алгоритму програми, яка обчислює суму нескінченного ряду та виводить на екран.



2) Код программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

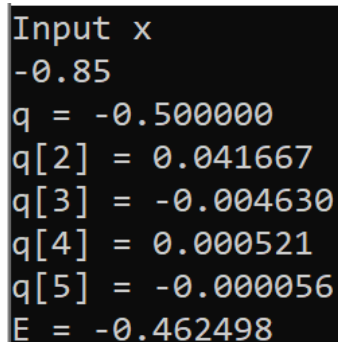
int main()
{
    int k;
    float q;
    float y = 0;

    printf("Input x\n");
    float x = scanf("%f", &x);

    q = (pow(-1, 1)*pow(x, 3*1-1))/(pow(1, 2)*2);
    printf("q = %f\n", q);
    y = y + q;

    for(int k = 2; fabs(q) > 0.0001; k++) {
        int f = 1;
        for(int i = 1; i <= k + 1; i++) {
            f = f*i;
        }
        q = (pow(-1, k)*pow(x, 3*k-1))/(pow(k, 2)*f);
        printf("q[%i] = %f\n", k, q);
        y = y + q;
    }
    printf("E = %f", y);
}
```

3) На рис. 1.2 наведений результат програми.

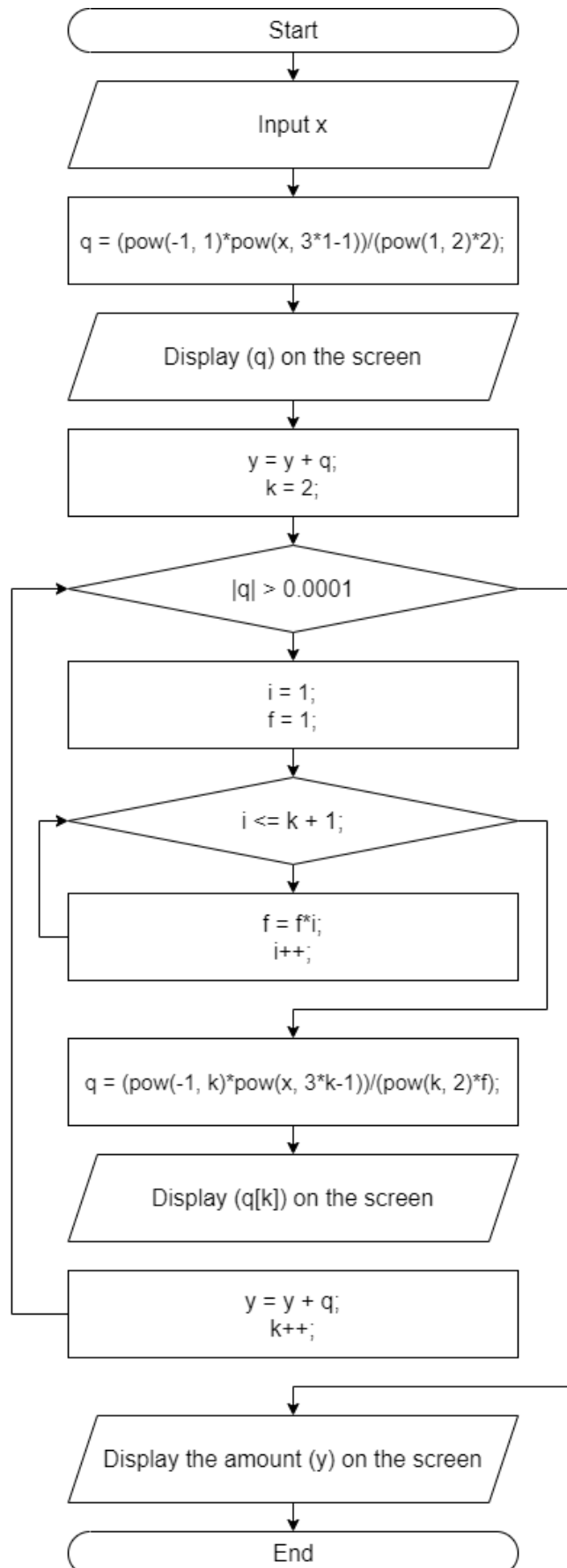


```
Input x
-0.85
q = -0.500000
q[2] = 0.041667
q[3] = -0.004630
q[4] = 0.000521
q[5] = -0.000056
E = -0.462498
```



## СПОСІБ – II

- 1) На рис. 2.1 наведена блок-схема алгоритму програми, яка обчислює суму нескінченного ряду та виводить на екран.



## 2) Код програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

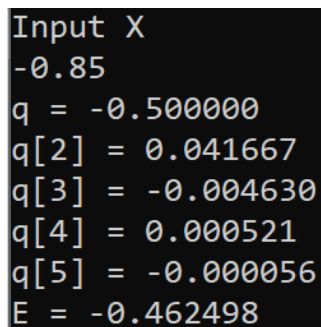
int main()
{
    float q;
    float y = 0;
    int k = 1;

    printf("Input X\n");
    float x = scanf("%f", &x);

    q = (pow(-1, 1)*pow(x, 3*1-1))/(pow(1, 2)*2);
    printf("q = %f\n", q);
    y = y + q;
    k = 2;

    while(fabs(q) > 0.0001) {
        int i = 1;
        int f = 1;
        while(i <= k + 1) {
            f = f*i;
            i++;
        }
        q = (pow(-1, k)*pow(x, 3*k-1))/(pow(k, 2)*f);
        printf("q[%i] = %f\n", k, q);
        y = y + q;
        k++;
    }
    printf("E = %f", y);
}
```

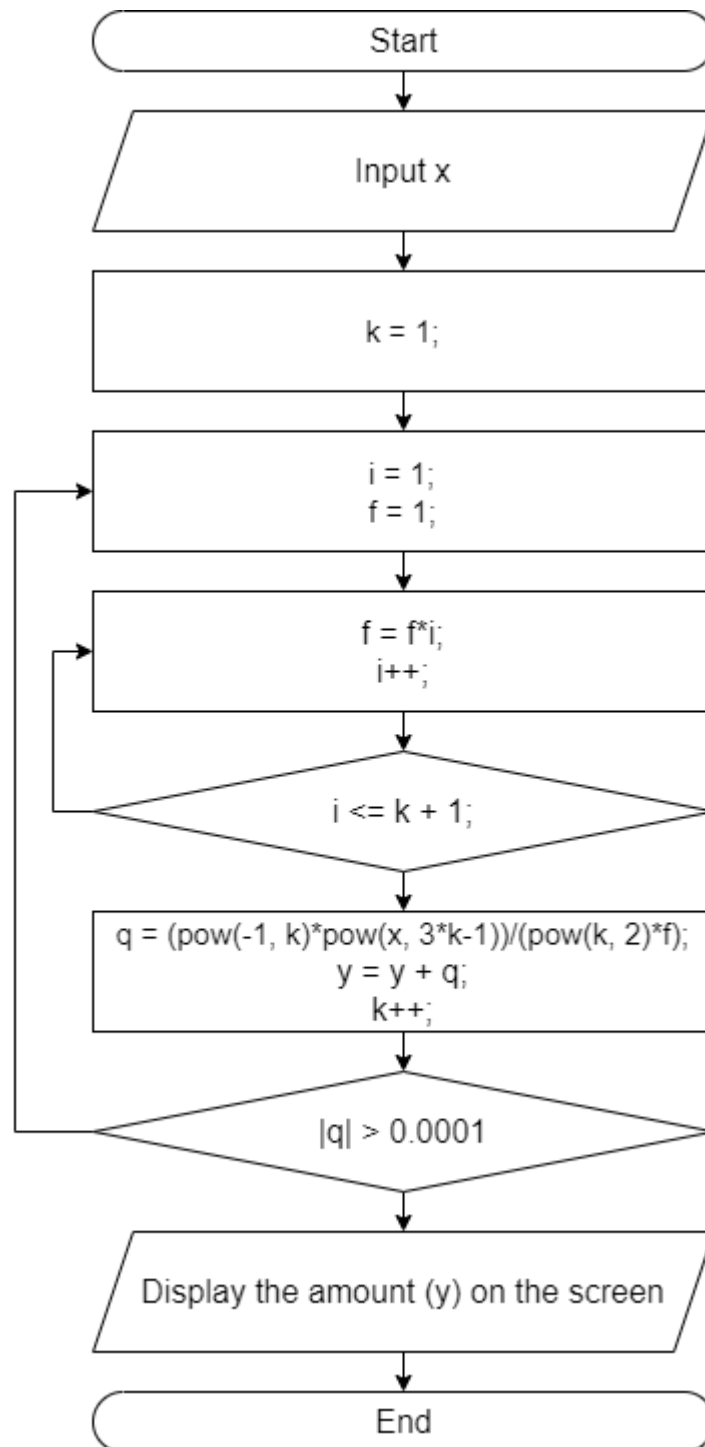
## 3) На рис. 2.2 наведений результат програми.



```
Input X
-0.85
q = -0.500000
q[2] = 0.041667
q[3] = -0.004630
q[4] = 0.000521
q[5] = -0.000056
E = -0.462498
```

### СПОСІБ – III

- 1) На рис. 3.1 наведена блок-схема алгоритму програми, яка обчислює суму нескінченного ряду та виводить на екран.



- 2) Код програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```

int main()
{
    float q;
    float y = 0;

    printf("Input x\n");
    float x = scanf("%f", &x);

    int k = 1;
    do {
        int i = 1;
        int f = 1;
        do {
            f = f*i;
            i++;
        } while (i <= k+1);
        q = (pow(-1, k)*pow(x, 3*k-1))/(pow(k, 2)*f);
        y = y + q;
        k++;
    } while (fabs(q) > 0.0001);
    printf("E = %f", y);
}

```

3) На рис. 3.2 наведений результат програми.

```

Input x
-0.85
E = -0.462498

```

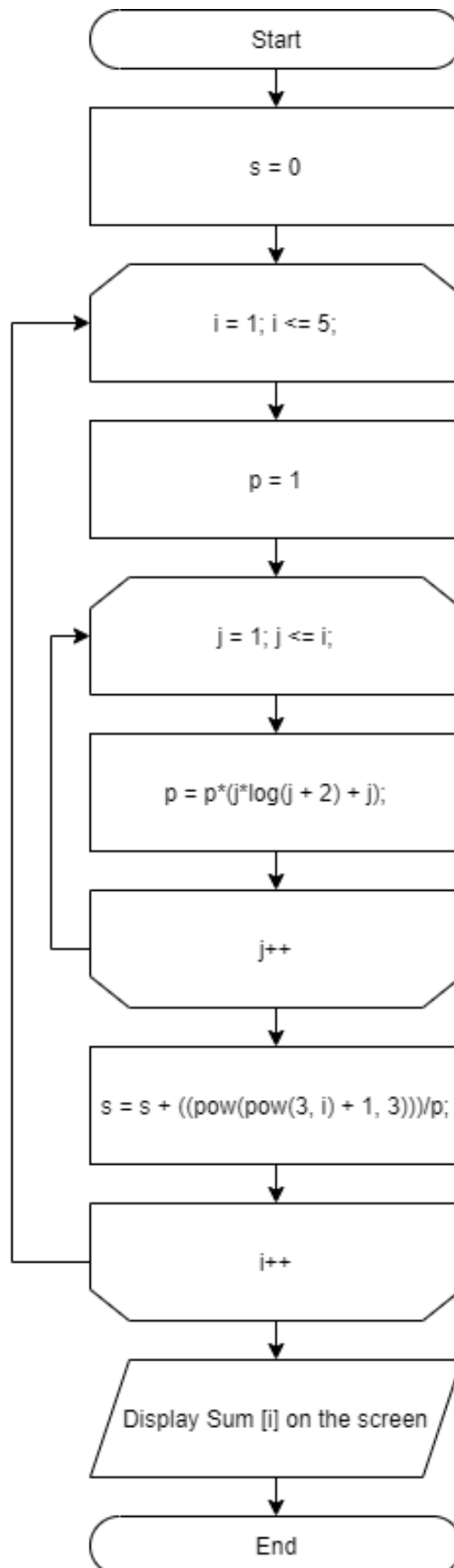
### Додаткове завдання

Відповідно до варіанту обчислити значення функції для  $n = 1, 2, 3, 4, 5$ . Програму написати за допомогою циклів **for**, **while** та **do/while**. Порівняти результати.

$$y = \sum_{i=1}^n \frac{(3^i + 1)^3}{\prod_{j=1}^i (j \ln(j+2) + j)}$$

## СПОСІБ – I

- 1) На рис. 4.1 наведена блок-схема алгоритму програми, яка обчислює значення функції та виводить на екран.




2) Код програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float p;
    float s = 0;
    for(int i = 1; i <= 5; i++) {
        p = 1;
        for(int j = 1; j <= i; j++) {
            p = p*(j*log(j + 2) + j);
        }
        s = s + ((pow(pow(3, i) + 1, 3)))/p;
        printf("Sum[%i] = %f\n", i, s);
    }
}
```

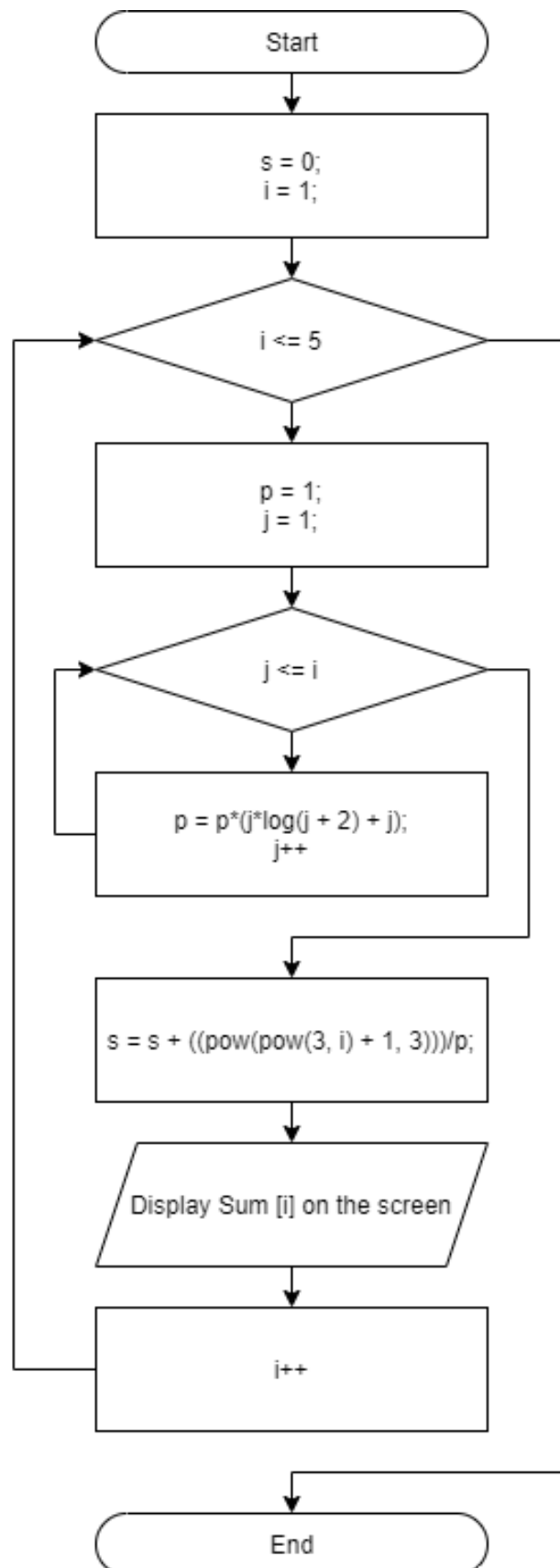
3) На рис. 4.2 наведений результат програми.



```
Sum[1] = 30.496342
Sum[2] = 130.338455
Sum[3] = 410.313690
Sum[4] = 1040.036011
Sum[5] = 2166.422607
```

## СПОСІБ – II

- 1) На рис. 5.1 наведена блок-схема алгоритму програми, яка обчислює значення функції та виводить на екран.

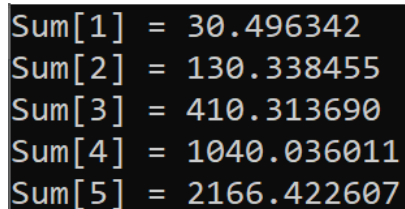


2) Код програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float p;
    float s = 0;
    int i = 1;
    int j;
    while(i <= 5) {
        p = 1;
        j = 1;
        while(j <= i) {
            p = p*(j*log(j + 2) + j);
            j++;
        }
        s = s + ((pow(pow(3, i) + 1, 3)))/p;
        printf("Sum[%i] = %f\n", i, s);
        i++;
    }
}
```

3) На рис. 5.2 наведений результат програми.

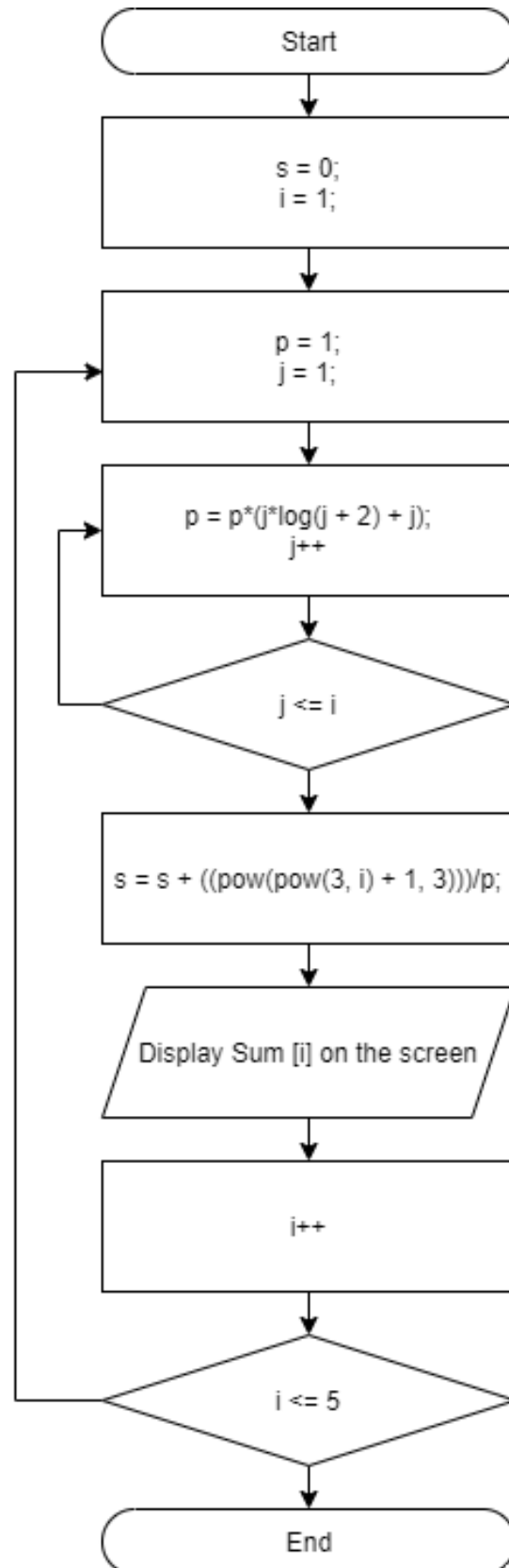


```
Sum[1] = 30.496342
Sum[2] = 130.338455
Sum[3] = 410.313690
Sum[4] = 1040.036011
Sum[5] = 2166.422607
```



### СПОСІБ – III

- 1) На рис. 6.1 наведена блок-схема алгоритму програми, яка обчислює значення функції та виводить на екран.

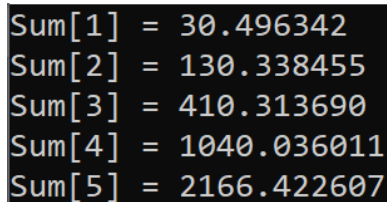


## 2) Код програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float p;
    float s = 0;
    int i = 1;
    int j;
    do {
        p = 1;
        j = 1;
        do {
            p = p*(j*log(j + 2) + j);
            j++;
        } while(j <= i);
        s = s + ((pow(pow(3, i) + 1, 3)))/p;
        printf("Sum[%i] = %f\n", i, s);
        i++;
    } while(i <= 5);
}
```

## 3) На рис. 6.2 наведений результат програми.



```
Sum[1] = 30.496342
Sum[2] = 130.338455
Sum[3] = 410.313690
Sum[4] = 1040.036011
Sum[5] = 2166.422607
```

**Висновок:** В ході лабораторної роботи я опанувала практичними навичками розробки програм із використанням структур повторення: for, while, do/while.