

Лабораторна робота №3. Організація циклічних процесів. Ітераційні цикли

Міністерство освіти і науки України
Національний технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації
і управління

Звіт

з лабораторної роботи № 1 з дисципліни
«Основи програмування»

«Обчислення арифметичних виразів»

Варіант __23__

Виконав студент _____ Берлінський Ярослав Владленович _____
(шифр, прізвище, ім'я, по батькові)

Перевірів _____
(прізвище, ім'я, по батькові)

Київ 2020

Назва роботи: організація циклічних процесів. Ітераційні цикли

Варіант: 23

Умова задачі:

23. З точністю 10^{-5} обчислити значення суми

$$\sum_{k=0}^{\infty} \frac{x^{2k}}{(2^k k!)}$$

Визначити кількість доданків.

Постановка задачі. Формулою задана послідовність загального члена нескінченного ряду чисел.

Результатом розв'язку є сума збіжного ряду чисел, представлених вищезазначеною формулою k-го члена, яка обчислюється за допомогою ітераційного циклу.

Результатом є вивід кількості доданків та суми членів ряду.

Розв'язок:

1. C++:

```
3 // LAB3
4 //
5 // Created by Берлинский Ярослав Владленович on 11.11.2020.
6 //
7 #include <iostream>
8 #include <cmath> // модуль для використання математичних функцій
9 #include <iomanip> // модуль форматування виводу стандартної бібліотеки C++
10 using namespace std;
11 int main(){
12     float x;
13     cout << "Print x: ";
14     cin >> x;
15     float eps = 0.00001;
16     //для простоти розуміння введемо змінні a=x/(b*d); d=k!; b=2^k; x1=x^(2*k)
17     int k=0;
18     double a = 1; // при k=0 a=1/(1*1)=1
19     double s = 1;
20     float d = 1; //факторіал k!(спочатку дорівнює 0, адже k=0 -> k!=1)
21     float b = 1; //2^0 = 1
22     float x1;
23     while (a>=eps){
24         k=k+1;
25         x1=pow(x,2*k);
26         b = 2*b;
27         d = d*k;
28         a = x1/(b*d);
29         s = s + a;
30         //cout <<setw(10)<<fixed<< setprecision(10) <<"A="<< a<<" S="<< s<<endl;
31     }
32     printf("Значення суми дорівнює S=%-11.10f\nКількість доданків K=%-4d\n", s,k);
33 }
34
35
```

2. Python:

```
1 x = float(input("Please input X: "))
2 eps = 0.00001
3 k=0
4 a = s = 1
5 d = 1
6 b = 1
7
8 while a>=eps:
9     k = k+1
10    x1 = x**(2*k)
11    d = d*k
12    b = 2*b
13    a = x1/(b*d)
14    s = s+a
15    print("%d) A=%-5.10f S=%-5.10f"%(k, a,s))
16    print("Значення суми дорівнює S=%-5.10f\nКількість доданків K=%-4d\n"%(s,k))
17
18
```

Тестування вхідних даних(C++):

1) x=0:

```
Print x: 0
Значення суми дорівнює S=1.0000000000
Кількість доданків K=1
Program ended with exit code: 0
```

All Output ▾ Filter [] []

2) x=1:

```
Print x: 1
Значення суми дорівнює S=1.6487211689
Кількість доданків K=7
Program ended with exit code: 0
```

All Output ▾ Filter [] []

3) $x=2$:

```
Print x: 2
Значення суми дорівнює S=7.3890546429
Кількість доданків K=12
Program ended with exit code: 0|
```

All Output ▾

Filter



Тестування аналогічних вхідних даних(Python):

1) $x=0$:

```
70513/yaroslav/Pycharm Projects/python Projects/
Please input X: 0
1) A=0.0000000000 S=1.0000000000
Значення суми дорівнює S=1.0000000000
Кількість доданків K=1
```

2) $x=1$:

```
Please input X: 1
1) A=0.5000000000 S=1.5000000000
2) A=0.1250000000 S=1.6250000000
3) A=0.0208333333 S=1.6458333333
4) A=0.0026041667 S=1.6484375000
5) A=0.0002604167 S=1.6486979167
6) A=0.0000217014 S=1.6487196181
7) A=0.0000015501 S=1.6487211682
Значення суми дорівнює S=1.6487211682
Кількість доданків K=7
```

3) $x=2$:

```
Please input X: 2
1) A=2.0000000000 S=3.0000000000
2) A=2.0000000000 S=5.0000000000
3) A=1.3333333333 S=6.3333333333
4) A=0.6666666667 S=7.0000000000
5) A=0.2666666667 S=7.2666666667
6) A=0.0888888889 S=7.3555555556
7) A=0.0253968254 S=7.3809523810
8) A=0.0063492063 S=7.3873015873
9) A=0.0014109347 S=7.3887125220
10) A=0.0002821869 S=7.3889947090
11) A=0.0000513067 S=7.3890460157
12) A=0.0000085511 S=7.3890545668
Значення суми дорівнює S=7.3890545668
Кількість доданків K=12
```

Нескладно перевірити вірність результатів. Інші дані можна перевірити самотійно, перейшовши безпосередньо до коду програми:

[GitHub](#)

Висновок: отже, програма розрахунку суми ряду чисел із заданою точністю для збіжної послідовності була виконана за допомогою ітераційного циклу з передумовою. Варто помітити, що при $\varepsilon = 0.00001$ (точність) сума буде точною до 5-го розряду включно. Внутрішні вирази(такі, як факторіал та степінь числа) були також пораховані в тілі циклу.

Щодо безпосередньо роботи самої структури ітераційного циклу, то його зручно використовувати, коли кількість операцій наперед невідома – що наочно демонструє і ця задача.