Міністерство освіти і науки України
НТУУ «Київський політехнічний інститут»
Фізико-технічний інститут

Програмування 4

Лабораторна робота №2

«Умовна операція та множинний вибір»

Виконав:
Студент II курсу ФТІ
групи ФЕ-81

Мєркєлов Ігор
Володимирович

2020

1. Завдання лабораторної роботи

Використовуючи оператор циклу while з передумовою та постумовою, знайти суму ряду з точністю $\varepsilon = 10^{-4}$, загальний член якого

$$a_n = \frac{1}{2^n} + \frac{1}{3^n}$$

2. Аналіз умови задачі.

У першому випадку, використовуючи модуль mpmath для точних обчислень на мові Python, та вбудованної функції nsum. lambda-запис дозволяє автоматично повертати та відразу записувати результат у функцію nsum. Такий специфічний запис забезпечує вміщення логіки всього лише в один рядок коду. Результатом виконання функції zetoToinf() є число 3.5 (3.5001). У другому випадку, використовуючи вбудовані в мову Python цикли while або for, досягається отримання суми ітеративним шляхом. На відмінно від першого прикладу, користувач може задати N-число, тобто кінцеве. Сума проходить формат {0:.4f}, тобто дотримується умови: e = 10^-4.

Результатом виконання программи є число 3.5 (починаючи з числа i=16).

Тож можна зробити висновок про те, що ряд — збіжний. За правилом Лопіталя - ряд збігається.

- $\varepsilon = 0{,}001$.
- Ряд збігається.

3. Код реалізації

```
# If we want to get result in [0, inf):
'''
from mpmath import nsum, inf

def zetoToinf():
    s = nsum(lambda i: (1 / (pow(2, i) ) + (1 / (pow(3, i)))), [0, inf])
    print(s)
zetoToinf()
'''


# If we want to get result in [0, N]:
N = int(input("Number of elements: "))
sum = 0
i = 0
while i < N:
    elem = (1 / (pow(2, i) ) + (1 / (pow(3, i))))
    print(i,"- element is: ", elem)
    sum = sum + elem
    i = i + 1
print(float("{0:.4f}".format(sum)))
```

4. **Виконання програми**:

1.

```
Result is:  3.5
kvant@ubl:~/Deskto
```

2.

```
Number of elements:  16
0 - element is:  2.0
1 - element is:  0.8333333333333333
2 - element is:  0.3611111111111111
3 - element is:  0.16203703703703703
4 - element is:  0.07484567901234568
5 - element is:  0.03536522633744856
6 - element is:  0.01699674211248285
7 - element is:  0.008269747370827618
8 - element is:  0.004058665790275873
9 - element is:  0.002003930263425291
10 - element is:  0.0009934975878084303
11 - element is:  0.0004939262792694768
12 - element is:  0.00024602230142315893
13 - element is:  0.00012269753797438632
14 - element is:  6.124423140812877e-05
15 - element is:  3.058726984437626e-05
3.5
```

1. Завдання лабораторної роботи

Використовуючи оператор циклу for, розв’язати наступні задачі: Знайти суму членів ряду, у якому $a_n = e^{-\sqrt{n}}$.

2. Аналіз умови задачі.

У першому випадку, використовуючи модуль mpmath для точних обчислень на мові Python, та вбудованної функції nsum. lambda-запис дозволяє автоматично повертати та відразу записувати результат у функцію nsum. Такий специфічний запис забезпечує вміщення логіки всього лише в один рядок коду. Результатом виконання функції zetoToinfExp() є число 2.6704 (2.67040681796633). У другому випадку, використовуючи вбудовані в мову Python цикли while або for, досягається отримання суми ітеративним шляхом. На відмінно від першого прикладу, користувач може задати N-число, тобто кінцеве. Сума проходить формат {0:.4f}, тобто дотримується умови: e = 10^-4. Результатом виконання программи є число 2.6704 (починаючи з числа i=173).Тож можна зробити висновок про те, що ряд - збіжний.

- $n \in N$
- for

3. **Код реалізації:**

```
# If we want to get result in [0, inf):
'''
from mpmath import nsum, inf, sqrt, exp

def zetoToinfExp():
    s = nsum(lambda n: exp(-sqrt(n)), [0, inf])
    print(s)
zetoToinfExp()
'''

# If we want to get result in [0, N]:
import math as mt

N = int(input("Number of elements: "))
sum = 0
i = 0
for i in range(0, N):
    elem = mt.exp(-mt.sqrt(i))
    print(i,"- element is: ", elem)
    sum = sum + elem
print(float("{0:.4f}".format(sum)))
```

4. **Виконання програми:**

1.

```
2.67040681796633
kvant@ubl:~/Deskto
```

2.

```
989 - element is:  2.1986002654177317e-14
990 - element is:  2.163929662114625e-14
991 - element is:  2.1298228867900002e-14
992 - element is:  2.0962704840557925e-14
993 - element is:  2.063263162026496e-14
994 - element is:  2.0307917894041544e-14
995 - element is:  1.998847392616972e-14
996 - element is:  1.9674211530104995e-14
997 - element is:  1.936504404090291e-14
998 - element is:  1.9060886288152863e-14
999 - element is:  1.876165456940705e-14
2.6704
```