

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

**Лабораторна робота
№1**

з дисципліни
«Алгоритмізації та програмування»

Виконав:
студент групи КН-108
Жеребецький Олег

Львів – 2018 р.

5. Зміст звіту

1. Постановка завдання.
2. Програма розв'язку завдання1.
3. Результати роботи .
4. Пояснення результатів.

Постановка завдання

1. Для x , що змінюється від a до b з кроком $(b-a)/k$, де $(k=10)$, обчислити функцію $f(x)$, використовуючи її розклад в степеневий ряд у двох випадках:

а) для заданого n ;

б) для заданої точності ϵ ($\epsilon=0.0001$).

Для порівняння знайти точне значення функції.

$$y = -\frac{1}{2} \ln(1 - 2x \cos \frac{\pi}{3} + x^2)$$

Функція

Діапазон зміни аргументу $0,1 \leq x \leq 0,8$

N = 35

Сума
$$S = \frac{x \cos \frac{\pi}{3}}{1} + \frac{x^2 \cos 2 \frac{\pi}{3}}{2} + \dots + \frac{x^n \cos n \frac{\pi}{3}}{n}$$

2. Результати розрахунків надрукувати у такому вигляді:

Обчислення функції

X=..... SN=..... SE=..... Y=.....

X=..... SN=..... SE=..... Y=.....

.....

X=..... SN=..... SE=..... Y=.....

Тут X- значення параметра; SN- значення суми для заданого n ; SE- значення суми для заданої точності; Y-точне значення функції.

Програма розв'язання завдання №1

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

int main (void)
{
    double a=0.1,b=0.8,s=0.0,s1=0.0,y , x=b;
    int n=35,j=1;
    for (; x>=a-0.03; x-=((b-a)/10))
    {
        printf( "X= %lf ",x);

        for (int i=1; i<=n; i++)
            s+=pow(x,i)*cos(i*3.14/3.0)/i;
        printf( "SN= %lf ",s);

        do { y=pow(x,j)*cos(j*3.14/3.0)/j;
            s1+=y;j++;}
        while (y>0.0001);
        printf( "SE= %lf ",s1);

        printf( "Y= %f \n",(-1)*log(1-2*x*cos (3.14/3)+pow(x,2))/2);
        s=0;s1=0;j=1;

    } return 0;
}
```

Результат роботи програми

X= 0.800000	SN= 0.087608	SE= 0.240662	Y= 0.087615
X= 0.730000	SN= 0.110180	SE= 0.232356	Y= 0.110181
X= 0.660000	SN= 0.127450	SE= 0.221604	Y= 0.127451
X= 0.590000	SN= 0.138828	SE= 0.208406	Y= 0.138828
X= 0.520000	SN= 0.143893	SE= 0.192763	Y= 0.143893
X= 0.450000	SN= 0.142452	SE= 0.174675	Y= 0.142452
X= 0.380000	SN= 0.134561	SE= 0.154141	Y= 0.134561
X= 0.310000	SN= 0.120517	SE= 0.131162	Y= 0.120517
X= 0.240000	SN= 0.100826	SE= 0.105737	Y= 0.100826
X= 0.170000	SN= 0.076142	SE= 0.077866	Y= 0.076142
X= 0.100000	SN= 0.047206	SE= 0.047551	Y= 0.047206

Пояснення результатів

Легко побачити , обрахунок за нормальною формулою і формулою суми з використання заданого n відповіді співпадають практично для усіх x , а розбіжність для перших трьох іксів викликана тим, що точність формули суми пов'язана з кількістю n доданків ряду, а саме: чи більше n тим більша точність, а у нас $n = 35$, що не є досить великим числом. Плюс, для більших іксів за $n=35$ членів ряду не встигає досягнутись достатня точність. Розбіжність при використанні вежі точності , пов'язана з тим, що

Формула під назвою Сума містить ряд, у якому при певних N з однаковою періодичністю міняється знак на протилежний, тобто умова того, що точність > 0.0001 мінусові значення одразу не задовільняють цю умову, тобто ряд не виконує потрібну роль, і

одоховує не повністю. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ::(

Підходяча програма

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

int main (void)
{
    double a=0.1,b=0.8,s=0.0,s1=0.0,y , x=b;
    int n=35,j=1;
    for (; x>=a-0.03; x-=((b-a)/10))
    {
        printf( "X= %lf  ",x);

        for (int i=1; i<=n; i++)
            s+=pow(x,i)*cos(i*3.14/3.0)/i;
        printf( "SN= %lf  ",s);

        do { y=pow(x,j)*cos(j*3.14/3.0)/j;
            s1+=y;j++;}
        while (y>0.0001 || y<-0.0001);
        printf( "SE= %lf  ",s1);

        printf( "Y= %f  \n",(-1)*log(1-2*x*cos (3.14/3)+pow(x,2))/2);
        s=0;s1=0;j=1;

    }
}
```

Підходячі відповіді)

```
3-1
jharvard@appliance (~/.Dropbox/Labs): ./lab3-1
X= 0.800000    SN= 0.087608    SE= 0.087739    Y= 0.087615
X= 0.730000    SN= 0.110180    SE= 0.110280    Y= 0.110181
X= 0.660000    SN= 0.127450    SE= 0.127397    Y= 0.127451
X= 0.590000    SN= 0.138828    SE= 0.138873    Y= 0.138828
X= 0.520000    SN= 0.143893    SE= 0.143826    Y= 0.143893
X= 0.450000    SN= 0.142452    SE= 0.142456    Y= 0.142452
X= 0.380000    SN= 0.134561    SE= 0.134607    Y= 0.134561
X= 0.310000    SN= 0.120517    SE= 0.120525    Y= 0.120517
X= 0.240000    SN= 0.100826    SE= 0.100792    Y= 0.100826
X= 0.170000    SN= 0.076142    SE= 0.076138    Y= 0.076142
X= 0.100000    SN= 0.047206    SE= 0.047205    Y= 0.047206
jharvard@appliance (~/.Dropbox/Labs):
```

Пояснення 2.0

Тепер ми поставили умову що точність така член ряду може лежати будь де на числовій прямій окрім $-0.0001 > y > 0.0001$. Тепер програма працює правильно. Тепер можемо поміти, що при обрахунку за точністю , за такої межі у-ка , чило виходить не достатньо точне в межах визначення Double щоб зрівнятись з обрахунком за точною формулою.