МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота
№3

з дисципліни
«Алгоритмізації та програмування»

Виконав:
студент групи КН-108
Лідзер Данило

Львів – 2018 р.

Тема

Обчислення функцій з використанням їхнього розкладу в степеневий ряд

Мета

Практика в організації ітераційних й арифметичних циклів.

Зміст звіту

1. Постановка завдання
2. Варіант завдання
3. Математична модель (формули, за якими виконуються обчислення доданків ряду)
4. Програма
5. Отримані результати
6. Стан проходження CS50

Постановка завдання

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 (ε=0.0001).

Для порівняння знайти точне значення функції.

Варіант завдання

17	$y = \frac{e^x + e^{-x}}{2}$	0.1 ≤ x ≤ 1	N=10	$S = 1 + \frac{x^2}{2!} + \ldots + \frac{x^{2n}}{(2n)!}$

Математична модель

Для обчислення суми ряду для заданого n використовувалась рекурентна формула: S = S + element, де element = element * ((x * x) / n), де n = 10.

Для обчислення суми ряду для заданної точності ε=0.0001 використовувалась рекурентна формула: S = S + element, де element = element * ((x * x) / m), де m - це значення, що збільшується на 1 з кожною ітерацією.

Код програми

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <math.h>
#include <conio.h>


int main()
{
    for(double x = 0.1; x <= 1.0; x += 0.1)
    {
        double summa1 = 1, summa2 = 1;
        double el1 = 1, el2 = 1;
        double y = (exp(x) + exp(-x)) / 2;
        double m = 1.0;


        for(double n = 1; n <= 10.0; n += 1.0)
        {
            el1 *= ((x * x) / n);
            summa1 += el1;
        }

        while(el2 > 0.0001)
        {
            el2 *= ((x * x) / m);
            summa2 += el2;
            m = m + 1.0;
        }

        printf("X = %f SN = %f SE = %f Y = %f\n", x, summa1, summa2, y);
    }

    _getch();
    return 0;
}

```

Результат роботи програми

```
X  = 0.100000 SN = 1.010050 SE = 1.010050 Y  = 1.005004
X  = 0.200000 SN = 1.040811 SE = 1.040811 Y  = 1.020067
X  = 0.300000 SN = 1.094174 SE = 1.094174 Y  = 1.045339
X  = 0.400000 SN = 1.173511 SE = 1.173510 Y  = 1.081072
X  = 0.500000 SN = 1.284025 SE = 1.284025 Y  = 1.127626
X  = 0.600000 SN = 1.433329 SE = 1.433326 Y  = 1.185465
X  = 0.700000 SN = 1.632316 SE = 1.632315 Y  = 1.255169
X  = 0.800000 SN = 1.896481 SE = 1.896471 Y  = 1.337435
X  = 0.900000 SN = 2.247908 SE = 2.247903 Y  = 1.433086
X  = 1.000000 SN = 2.718282 SE = 2.718279 Y  = 1.543081
```

Стан проходження CS50

Тиждень 3. Практичне завдання 3

Код програми

```c
#include <cs50.h>

#include "helpers.h"

/**
 * Returns true if value is in array of n values, else false.
 */
bool search(int value, int values[], int n)
{
    int lower = 0;
    int upper = n - 1;

    while(lower <= upper)
    {
        int middle = (lower + upper) / 2;

        if(values[middle] == value)
        {
            return true;
        }

        else if(values[middle] < value)
        {
            lower = middle + 1;
        }

        else if(values[middle] > value)
        {
            upper = middle - 1;
        }
    }
    return false;
}

```

```c
/**
 * Sorts array of n values.
 */
void sort(int values[], int n)
{
    for(int k = 0; k < n - 1; k++)
    {
       int swaps = 0;

       for(int i = 0; i < n - 1 - k; i++)
       {
           if(values[i] > values[i + 1])
           {
               int temp = values[i + 1];
               values[i + 1] = values[i];
               values[i] = temp;
               swaps++;
           }
       }

        if(!swaps)
        {
            break;
        }
    }
    return;
}
```

Результат роботи програми

```
haystack[996] =
haystack[997] =
haystack[998] =
haystack[999] =
haystack[1000] =

Found needle in haystack!

~/workspace/pset3/find/ $
```