

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

**Кафедра систем штучного інтелекту**

**Лабораторна робота  
з дисципліни  
«Алгоритмізація та програмування»**

**Виконав:**  
студент групи КН-  
109  
Кіндрат Володимир  
**Викладач:**  
Гасько Р.Т.

Львів – 2018р.

### Лабораторна робота №3.

**Тема: "Обчислення функцій з використанням їхнього розкладу в степеневий ряд" Мета: Практика в організації ітераційних й арифметичних циклів.**

1. Короткі теоретичні відомості Дійсна функція  $f(x)$  називається аналітичною в точці  $\epsilon$ , якщо в деякому околі  $|x-\epsilon|<R$  цієї точки функція розкладається в степеневий ряд (ряд Тейлора):

$$f(x) = f(\epsilon) + f'(\epsilon)(x-\epsilon) + \frac{f''(\epsilon)}{2!}(x-\epsilon)^2 + \frac{f'''(\epsilon)}{3!}(x-\epsilon)^3 + \dots + \frac{f^{(n)}(\epsilon)}{n!}(x-\epsilon)^n + \dots$$

2

(1)

При  $\epsilon=0$  отримаємо ряд Маклорена:

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \dots + \frac{f^{(n)}(0)}{n!}x^n + \dots$$

(2)

Різниця

$R(x, f(x))$

$f$

$k \leq n$

$k$

$k$

$n$

$(\cdot) (\cdot)$

$(\cdot) !$

$(\cdot)$

$(\cdot) = \dots = \sum \varepsilon$

$\varepsilon > 0$  (3) називається залишковим членом і  $\varepsilon$  помилкою при

заміні функції  $f(x)$  поліномом Тейлора. Для ряду Маклорена

$R(x)$

$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n + R_n(x)$

+

1

1

1  $\theta$

де  $0 < \theta < 1$ . (4) Таким чином, обчислення значення функції

можна звести до обчислення суми числового ряду  $a_1 + a_2 + \dots + a_n + \dots$

. (5) Відомо, що числовий ряд називається збіжним, якщо існує границя послідовності його часткових сум:

$S$

$S$

$\lim_{n \rightarrow \infty} S_n$

, (6) де  $S_n = a_1 + a_2 + \dots + a_n + \dots$ . Число  $S$  називається сумою ряду. З формули (13) отримаємо  $S = S_n + R_n$ , де  $R_n$  - залишок ряду, причому  $R \rightarrow 0$  при  $n \rightarrow \infty$ .

23 Для знаходження суми  $S$  збіжного ряду (5) із заданою точністю  $\varepsilon$  потрібно вибрати число доданків  $n$  настільки великим, щоб виконувалась нерівність  $|R_n| < \varepsilon$ . Тоді часткова сума  $S_n$  приблизно може бути прийнята за точну суму  $S$  ряду (5). Приблизно  $n$  вибрати так, щоб виконувалась нерівність  $|S_{n+1} - S_n| < \varepsilon$  або  $a_n < \varepsilon$ . Завдання зводиться до заміни функції степеневим рядом і знаходженню суми деякої кількості доданків  $S$  а  $x$   $n$   $n = \sum ( , )$  при різних параметрах підсумовування  $x$ . Кожен доданок суми залежить від параметра  $x$  і номера  $n$ , що визначає місце цього доданка в сумі. Зазвичай формула загального члена суми належить одному з таких трьох типів:

а)

$$x^n n!;$$

$$(-1)^n \frac{x^n}{n!} + \frac{x^{n-1}}{(n-1)!} + \frac{x^{n-2}}{(n-2)!} + \dots + \frac{x^2}{2!} + \frac{x}{1!} + 1;$$

$$x^n$$

$$n$$

$$n^2 2^n (-1)^n;$$

б)

$$\cos(x) \frac{x^n}{n!};$$

$$\sin(x) \frac{x^{2n-1}}{(2n-1)!} - \frac{x^{2n-3}}{(2n-3)!} + \dots - \frac{x^3}{3!} + \frac{x}{1!};$$

$$\cos(x) \frac{x^{2n}}{(2n)!} - \frac{x^{2n-2}}{(2n-2)!} + \dots - \frac{x^4}{4!} + \frac{x^2}{2!} - 1;$$

в)

$$x^n n! 4! 1! + \dots;$$

( )

$$\cos(\dots) - 1 2^n n! x^n;$$

n

$$n! x^n 2! 1! 2! + \dots! ( )$$

. У випадку а) для обчислення члена суми  $a_n$  доцільно використовувати рекурентні співвідношення, тобто представляти наступний член суми через попередній:  $a_{n+1} = \psi(x, n)a_n$ . Це дозволить істотно скоротити обсяг обчислювальної роботи. Крім того, обчислення члена суми за загальною формулою в деяких випадках неможливо (наприклад через наявність  $n!$ ). У випадку б) застосування рекурентних співвідношень недоцільно. Обчислення будуть найбільш ефективними, якщо кожен член суми обчислювати за загальною формулою  $a_n = \varphi(x, n)$ .

24 У випадку в) член суми доцільно представити у вигляді двох співмножників, один із яких обчислюється за рекурентним співвідношенням, а інший безпосередньо  $a_n = \varphi(x, n) \cdot c_n(x, n)$ , де  $c_n = c_{n-1} \psi(x, n)$ .

## Варіант 9.

### Завдання 1.

8	$y = \frac{x \sin \frac{\pi}{4}}{1 - 2x \cos \frac{\pi}{4} + x^2}$	$0,1 \leq x \leq 0,8$	40	$S = x \sin \frac{\pi}{4} + x^2 \sin 2 \frac{\pi}{4} + \dots + x^n \sin n \frac{\pi}{4}$
9	$y = \frac{1}{4} \ln \frac{1+x}{1-x} + \frac{1}{2} \operatorname{arctg} X$	$0,1 \leq x \leq 0,8$	3	$S = x + \frac{x^5}{5} + \dots + \frac{x^{4n+1}}{4n+1}$
10	$y = e^{\cos x} \cos(\sin x)$	$0,1 \leq x \leq 1$	20	$S = 1 + \frac{\cos x}{1!} + \dots + \frac{\cos nx}{n!}$
11	$y = (1 + 2x^2)e^{x^2}$	$0,1 \leq x \leq 1$	10	$S = 1 + 3x^2 + \dots + \frac{2n+1}{n!} x^{2n}$
12	$\frac{1}{\ln(1+x)}$	$0,1 \leq x \leq 0,8$	35	$x \cos \frac{\pi}{4} - x^2 \cos 2 \frac{\pi}{4} + \dots + x^n \cos n \frac{\pi}{4}$

## Розв'язання:

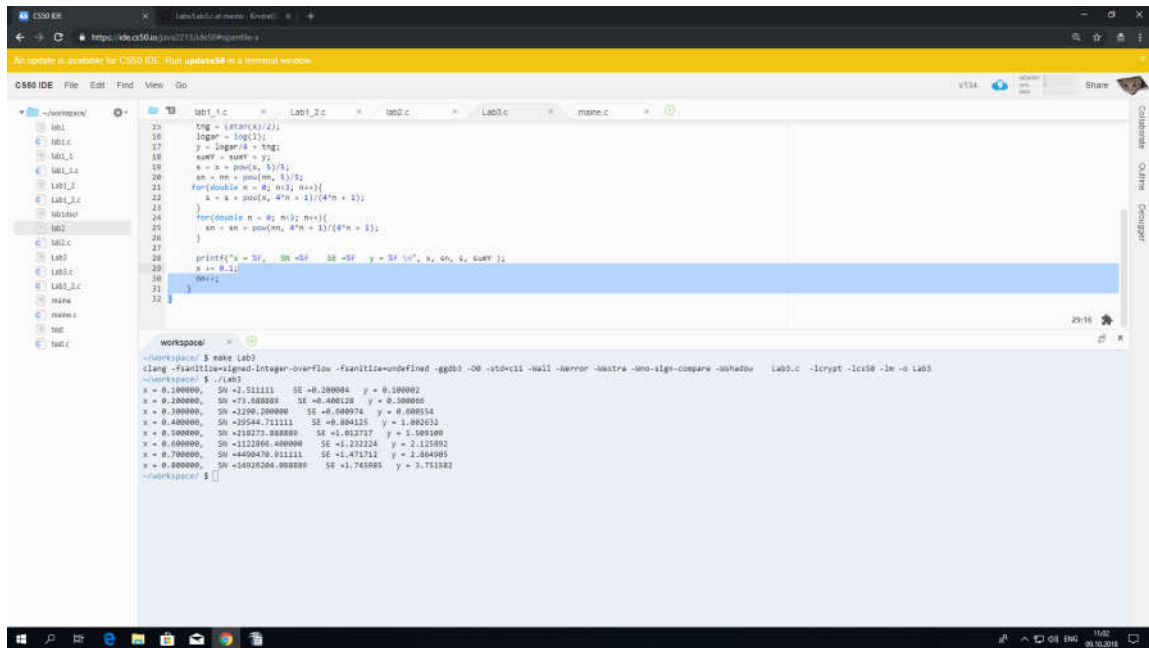
```
#include<stdio.h>
#include<math.h>

int main(void)
{

    double  logar, l, tng, y, s, sn;
    double x = 0.1;
    double sumY = 0;
    double nn = 1;
    for(int a = 1; a <= 8; a++)
    {
        l = (1 + x) / (1 - x);
        tng = (atan(x)/2);
        logar = log(l);
        y = logar/4 + tng;
        sumY = sumY + y;
        s = x + pow(x, 5)/5;
        sn = nn + pow(nn, 5)/5;
        for(double n = 0; n<3; n++){
            s = s + pow(x, 4*n + 1)/(4*n + 1);
        }
        for(double n = 0; n<3; n++){
            sn = sn + pow(nn, 4*n + 1)/(4*n + 1);
        }

        printf("x = %f,  SN =%f  SE =%f  y = %f\n", x, sn, s, sumY );
        x += 0.1;
        nn++;
    }
}
```

# Результат роботи програми:



The screenshot shows the CS50 IDE interface. The main editor displays a C program with the following code:

```
15 trig = atan(x/z);
16 logar = log(1);
17 y = logpar/4 + tngi;
18 sqrt = sqrt(x);
19 n = x + pow(x, 3)/5;
20 an = mn + pow(mn, 3)/5;
21 for(double n = 0; n/2; n++){
22     i = x + pow(x, 4^n + 3)/(4^n + 3);
23 }
24 for(double n = 0; n/2; n++){
25     an = an + pow(mn, 4^n + 1)/(4^n + 3);
26 }
27 printf("n = %f, SN = %f SE = %f y = %f la", x, an, i, sqrt);
28 x = 0.1;
29 while(1)
30 }
```

The terminal output shows the results of running the program:

```
~/workspace/ $ make lab0
clang -fsanitize=signed-integer-overflow -fsanitize=undefined -ggdb0 -O0 -std=c11 -Wall -Werror -Wextra -Wno-sign-compare -Wshadow -lcrypt -lc50 -lm -o lab0
~/workspace/ $ ./lab0
x = 0.100000, SN = 2.511111 SE = 0.200004 y = 0.100002
x = 0.200000, SN = 73.680000 SE = 0.400128 y = 0.300000
x = 0.300000, SN = 2290.200000 SE = 0.600074 y = 0.600054
x = 0.400000, SN = 29544.711111 SE = 0.800435 y = 1.802632
x = 0.500000, SN = 216273.800000 SE = 1.012717 y = 1.509100
x = 0.600000, SN = 1122066.400000 SE = 1.232224 y = 2.125892
x = 0.700000, SN = 4400470.811111 SE = 1.471712 y = 2.864901
x = 0.800000, SN = 14920204.800000 SE = 1.745905 y = 3.751582
~/workspace/ $
```

**Висновок:** Під час роботи навчився створювати найпростіші комп'ютерні програми та обчислювати за допомогою них математичні вирази. Зрозумів різницю у використанні різних типів змінних та деяких математичних функцій.

## Прогрес курсу CS50:

