

## Лабораторна робота (Комп'ютерний практикум) № 2: «Рекурсивний виклик функції»

### 2.1. Завдання

Ознайомитись з рекурсивним викликом функції. Розробити алгоритмом розрахунку значення функції за її розкладенням у ряд за умови отримання результату з заданою точністю. Врахувати діапазон дозволених значень для змінної  $x$ .

У якості індивідуального завдання необхідно написати програмний код, що реалізує алгоритм розрахунку значень функцій за їх розкладенням в ряд із заданою користувачем точністю.

Використати математичні моделі геометричних фігур з Додатку В-2. Завдання обрати згідно свого варіанта.

#### РЕЗУЛЬТАТ РОБОТИ ПОТРІБНО:

1. Роздрукувати (вивести на екран) точне значення функції, її приблизне значення та точність розрахунку.
2. Відкритий для редагування програмний код розмістити на сайті <https://replit.com/> (посилання через кнопку «+ Invite»).
3. Звіт до комп'ютерного практикуму № 2 додати в свій Клас на ресурсі <https://classroom.google.com/>.

### 2.2. Теоретичні відомості

Рекурсія – це виклик функції чи процедури з неї самої з іншими значеннями вхідних параметрів безпосередньо чи опосередковано через інші функції (наприклад, функція А викликає функцію В, а функція В викликає функцію А). Кількість вкладених викликів функції чи процедури називається глибиною рекурсії.

Принциповими властивостями рекурсивного визначення є:

1. Визначення об'єкту самого через себе, але з іншими параметрами;
2. Завершеність ланцюжка визначень на деякому значенні аргументів.

При програмуванні рекурсії слід впевнитися у тому, що задачу не можна легше розв'язати за допомогою ітеративних процедур. Рекурсію використовувати варто тоді, коли нерекурсивний підхід сильно ускладнює програмування або не призводить до суттєвого зменшення часу роботи алгоритму або об'єму пам'яті необхідної для роботи.

Наприклад задачу знаходження факторіалу заданого числа можна

розв'язати ітераційним або рекурсивним способом. Нижче наведено два приклади.

**Приклад\_1** знаходження факторіалу числа ітераційним способом:

```
//factorial iteration
```

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

```
int main(void)
```

```
{
```

```
double dnumber = 5.0;
```

```
double dresult;
```

```
printf("\nФакторіал числа %5.0lf", dnumber);
```

```
for (dresult = 1; dnumber > 1; dresult *= (dnumber--))
```

```
{;}
```

```
printf("дорівнює: %5.0lf\n", dresult);
```

```
return(0);
```

```
}
```

**Приклад\_2** знаходження факторіалу числа рекурсивним способом:

```
//factorial recursion
```

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

```
double dfactorial(double danswer);
```

```
int main(void)
```

```
{
```

```
double dnumber = 5.0;
```

```
double dresult;
```

```
dresult = dfactorial(dnumber);
```

```
printf("\nФакторіал числа %5.0lf дорівнює: %5.0lf\n", dnumber, dresult);
```

```
return(0);
```

```
}
```

```
double dfactorial(double danswer)
```

```
{
```

```
if(danswer <= 1.0)
```

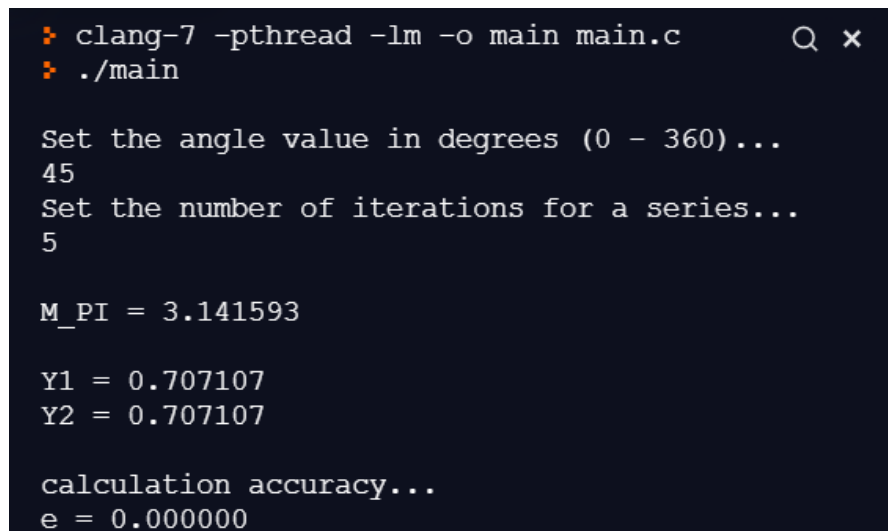
```
return(1.0);  
else  
return(danswer * dfactorial(danswer - 1.0));  
}
```

### 2.3. Опис програми

Алгоритм реалізує наступні функції: виклик функції, яка обчислює значення ряду та рекурсивний виклик функції для обчислення факторіалу математичного виразу.

### 2.4. Результати виконання програми

На рисунку 2.1 копію екрану під час виконання програмного коду.



```
❖ clang-7 -pthread -lm -o main main.c  
❖ ./main  
  
Set the angle value in degrees (0 - 360)...  
45  
Set the number of iterations for a series...  
5  
  
M_PI = 3.141593  
  
Y1 = 0.707107  
Y2 = 0.707107  
  
calculation accuracy...  
e = 0.000000
```

Рисунок 2.1 – Результат виконання програми

### 2.5. Висновки по роботі

У результаті виконання комп'ютерного практикуму було досліджено особливості використання рекурсивного виклику функції. У якості прикладу, виконано розробку коду програми, що під час виконання зчитує значення кута в градусах і кількість ітерацій для розрахунку приблизного значення функції  $\sin(x)$  та, в залежності від цього, реалізує алгоритм розрахунку значення функції  $\sin(x)$  за її розкладенням в ряд із заданою користувачем точністю.

### 2.6. Лістинг програми

// lab2.c 2022

```

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

double dfactorial(double danswer);

int main(void) {
    float Xd, Xr;
    float Y1, Y2;
    float e;
    int k;
    int N;

    printf("\nSet the angle value in degrees (0 - 360)...\n");
    scanf("%f", &Xd);

    printf("Set the number of iterations for a series...\n");
    scanf("%d", &N);

    Xr = Xd * 2.0 * M_PI / 360.0;

    Y1 = sin(Xr);

    for(k = 0; k <= N; k++)
    {
        Y2 += pow(-1, k) * ((pow(Xr, 2*k + 1))/(dfactorial(2*k + 1)));
    }
    e = Y1 - Y2;

    printf("\nM_PI = %lf\n", M_PI);
    printf("\nY1 = %lf\n", Y1);
    printf("Y2 = %lf\n", Y2);
    printf("\ncalculation accuracy...\n");
    printf("e = %lf\n", e);

    return 0;
}

double dfactorial(double danswer)
{
    if(danswer <= 1.0)
        return(1.0);
    else
        return(danswer * dfactorial(danswer - 1.0));
}

```

1. Надати загальний опис функції, що приймає та повертає значення.
2. Що таке рекурсія? В яких задачах виникає рекурсивний виклик функції?
3. Що таке пряма рекурсія?
4. Що таке опосередкована рекурсія?
5. Привести приклад алгоритму виконання розрахунку з заданою точністю.