**Міністерство освіти і науки України**
**Національний технічний університет України**
**«Київський політехнічний інститут імені Ігоря Сікорського»**
**Факультет інформатики та обчислювальної техніки**
**Кафедра обчислювальної техніки**

# Лабораторна робота №7

з дисципліни
«Алгоритми і структури даних»

Виконав:
студент групи ІМ-42
Туров Андрій Володимирович
номер у списку групи: 28

Перевірив:
Сергієнко А. М.

Київ 2025

# Завдання

Дане натуральне число $n$. Знайти суму перших $n$ членів ряду чисел, заданого рекурентною формулою. Розв'язати задачу трьома способами:

1. у програмі використати рекурсивну функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному спуску.
2. у програмі використати рекурсивну функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному поверненні.
3. у програмі використати рекурсивну функцію, яка виконує обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні.

# Варіант 28

$$F_1 = 1; F_{i+1} = F_1 * x^2 / \left(4i^2 - 2i\right), \quad i > 0;$$
$$\sum_{i=1}^{n} = \operatorname{ch} x;$$

# Текст програм

```c
#include "input.h"
#include <stdio.h>

double sum_nth_desc(const unsigned int n, const double x,
 ↪  unsigned int i,
                    double previous, double sum) {
  if (i ⩾ n) {
    return sum;
  }

  double current;
  if (i == 0) {
    current = 1.0;
  } else {
    current = previous * x * x / (4 * i * i - 2 * i);
  }
  printf("i = %u: %lf\n", i, current);

  sum += current;

  return sum_nth_desc(n, x, i + 1, current, sum);
}
```

```c
int main(int argc, char **argv) {
  unsigned int n;
  double x;

  get_input(argc, argv, &n, &x);

  double sum = sum_nth_desc(n, x, 0, 0, 0);

  printf("ch(%lf) = %lf\n", x, sum);
  return 0;
}
```

Файл 1: a.c

```c
#include "input.h"
#include <stdio.h>

typedef struct {
  double term;
  double sum;
} Result;

Result sum_nth_asc(unsigned int i, double x) {
  Result result;
  if (i == 0) {
    result.term = 1.0;
    result.sum = 1.0;
    printf("i = 0: %lf\n", result.term);

    return result;
  } else {
    Result previous = sum_nth_asc(i - 1, x);
    result.term = previous.term * x * x / (4 * i * i - 2 * i);
    result.sum = previous.sum + result.term;
    printf("i = %u: %lf\n", i, result.term);

    return result;
  }
}

int main(int argc, char **argv) {
  unsigned int n;
  double x;

  get_input(argc, argv, &n, &x);

  if (n == 0) {
    printf("ch(%lf) = 0\n", x);
    return 0;
  }
```

```
  Result result = sum_nth_asc(n - 1, x);
  printf("ch(%lf) = %lf\n", x, result.sum);
  return 0;
}
```

Файл 2: b.c

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

double sum_nth_mixed(unsigned int i, unsigned int n, double x,
 ↪  double prev) {
  double current;
  if (i == 0) {
    current = 1.0;
  } else {
    current = prev * x * x / (4 * i * i - 2 * i);
  }
  printf("i = %u: %lf\n", i, current);

  if (i >= n - 1) {
    return current;
  }

  double sum_rest = sum_nth_mixed(i + 1, n, x, current);
  return current + sum_rest;
}

int main(int argc, char **argv) {
  unsigned int n;
  double x;
  get_input(argc, argv, &n, &x);

  if (n == 0) {
    printf("ch(%lf) = 0\n", x);
    return 0;
  }

  double total = sum_nth_mixed(0, n, x, 0);
  printf("ch(%lf) = %lf\n", x, total);
  return 0;
}
```

Файл 3: c.c

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

```c
void get_input(const int argc, char **argv, unsigned int *n,
 ↪  double *x) {
  if (argc ⩽ 1) {
    printf("\033[2mYou can use the program as a CLI: \033[1;2m%s
     ↪  [n] "
           "[x]\n\033[0m\n\n",
           argv[0]);
    printf("Enter precision (n): ");
    scanf("%u", n);

    printf("Enter argument  (x): ");
    scanf("%lf", x);
  } else if (argc == 2) {
    *n = atoi(argv[1]);

    printf("Enter argument (x): ");
    scanf("%lf", x);
  } else {
    *n = atoi(argv[1]);
    *x = atof(argv[2]);
  }
}
```

Файл 4: input.c

```c
#pragma once

void get_input(const int argc, char **argv, unsigned int *n,
 ↪  double *x);
```

Файл 5: input.h

```c
#include "input.h"
#include <stdio.h>

double series_next(unsigned int i, double x, double previous) {
  double result;
  if (i == 0) {
    result = 1;
  } else {
    result = previous * x * x / (4 * i * i - 2 * i);
  }
  printf("i = %u: %lf\n", i, result);
  return result;
}

int main(int argc, char **argv) {
  unsigned int n;
  double x;
```

```c
  get_input(argc, argv, &n, &x);

  double sum = 0;
  double previous = 0;
  for (unsigned int i = 1; i ⩽ n; i++) {
    previous = series_next(i - 1, x, previous);
    sum += previous;
  }
  printf("ch(%lf) = %lf\n", x, sum);
  return 0;
}
```

Файл 6: linear.c

# Результати тестування програми

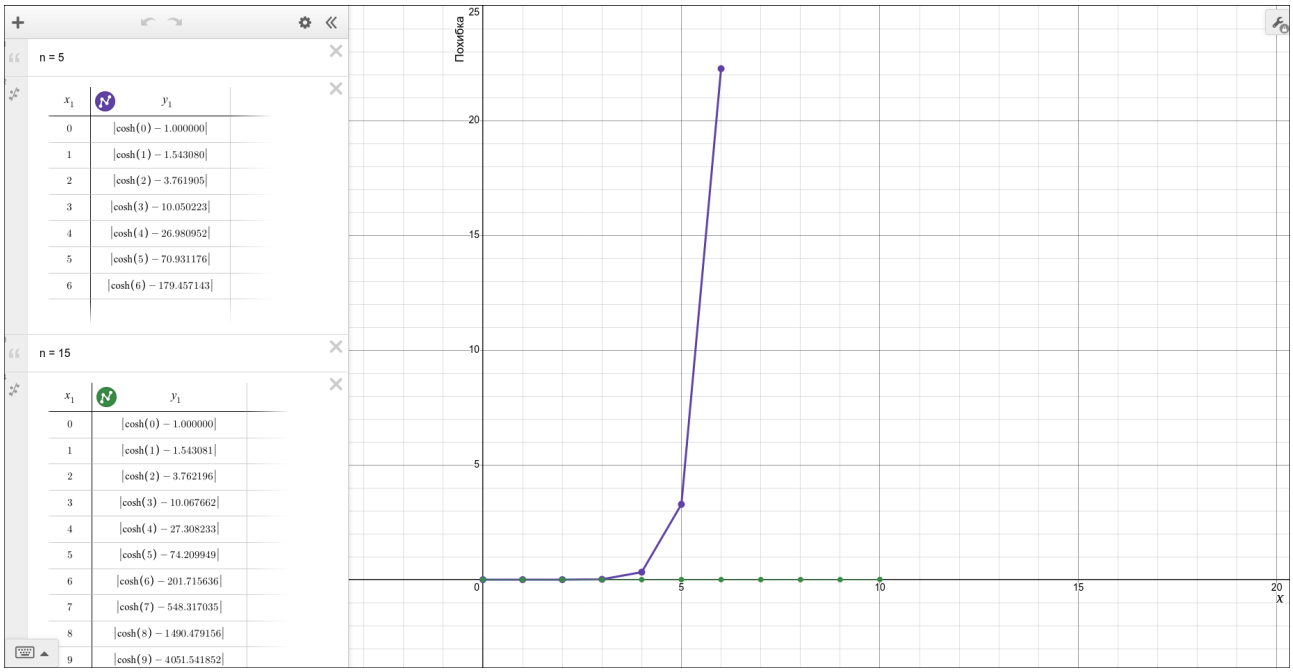


Рис. 1: Графік похибки $x$ за різних $n$

```
>>> ./linear 5 1
i = 0: 1.000000
i = 1: 0.500000
i = 2: 0.041667
i = 3: 0.001389
i = 4: 0.000025
ch(1.000000) = 1.543080
```

```
>>> ./a 5 1
i = 0: 1.000000
i = 1: 0.500000
i = 2: 0.041667
i = 3: 0.001389
i = 4: 0.000025
ch(1.000000) = 1.543080
```

```
>>> ./b 5 1
i = 0: 1.000000
i = 1: 0.500000
i = 2: 0.041667
i = 3: 0.001389
i = 4: 0.000025
ch(1.000000) = 1.543080
```

```
>>> ./c 5 1
i = 0: 1.000000
i = 1: 0.500000
i = 2: 0.041667
i = 3: 0.001389
i = 4: 0.000025
ch(1.000000) = 1.543080
```

```
>>> ./a 10 3.1415
i = 0: 1.000000
i = 1: 4.934511
i = 2: 4.058233
i = 3: 1.335027
i = 4: 0.235275
i = 5: 0.025799
i = 6: 0.001929
i = 7: 0.000105
i = 8: 0.000004
i = 9: 0.000000
ch(3.141500) = 11.590883
```

```
>>> ./b 10 3.1415
i = 0: 1.000000
i = 1: 4.934511
i = 2: 4.058233
i = 3: 1.335027
i = 4: 0.235275
i = 5: 0.025799
i = 6: 0.001929
i = 7: 0.000105
i = 8: 0.000004
i = 9: 0.000000
ch(3.141500) = 11.590883
```

# Висновок

Навчився використовувати рекурсивні алгоритми для розв'язання задач. Попрактикувався у написанні рекурсивних функцій, що виконують різні операції на різних етапах виконання. Ще в LaTeX навчився це все діло оформлювати.