**КПІ ім. Ігоря Сікорського**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

# Звіт до комп'ютерного практикуму з курсу «Основи програмування»

Прийняв
асистент кафедри ІПІ
Ахаладзе А. Е.
«1» листопада 2024 р.

Виконав
студент групи ІП-43
Дутов І. А.

**Київ 2024**

# Комп'ютерний практикум №4

***Тема:*** *Оператори циклу. Робота з масивами*

**Завдання:**

Написати програму, яка повинна виводити таблицю значень синусів **або** косинусів (розрахованих за допомогою розкладання функції в ряд Тейлора) і табличних значень, а також їх різницю в заданому діапазоні із заданою точністю..

## Текст програми

***../src/main.c***

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

#include "input/input.h"

#define MAX_CHARS_FLOAT 50
#define MAX_CHARS_DEGREES MAX_CHARS_FLOAT
#define MAX_CHARS_PRECISION MAX_CHARS_FLOAT

#define MAX_DEGREES FLT_MAX
#define MIN_DEGREES FLT_MIN

#define MAX_PRECISION 1.0f
#define MIN_PRECISION FLT_MIN
#define TOLERANCE 1e-7f

#define COLUMN_WIDTH 20
#define COLUMN_COUNT 4
#define BORDER_COUNT COLUMN_COUNT + 1
#define SEPARATOR '='
#define DECIMAL_PLACES 15

float mod360(float value) { return fmodf(value, 360.0f); }

float degrees_to_radians(float degrees) { return degrees / 180.0f * M_PI; }

float calculate_taylor_sin(float x, float e) {
  float d, sin_;
  d = sin_ = x;
  float n = 1.0f;
  do {
    d *= -x * x / (n + 1.0f) / (n + 2.0f);
    sin_ += d;
    n += 2.0f;
  } while (fabs(d) > e);
  return sin_;
}

void print_separator(int length) {
  for (int i = 0; i < length; i++) {
    putchar(SEPARATOR);
  }
  putchar('\n');
}

void print_calculations_row(float x, float e) {
  float rad_x = degrees_to_radians(mod360(x));
  float sin_x = sinf(rad_x);
  float taylor_sin = calculate_taylor_sin(rad_x, e);
  float diff = sin_x - taylor_sin;
  printf("|%-*.*g|%-*.*g|%-*.*g|%-*.*g|\n", COLUMN_WIDTH, DECIMAL_PLACES, x,
         COLUMN_WIDTH, DECIMAL_PLACES, sin_x, COLUMN_WIDTH, DECIMAL_PLACES,
         taylor_sin, COLUMN_WIDTH, DECIMAL_PLACES, diff);
}

void print_demo() {
  printf(
      "Ця програма знаходить sin(x) для відрізку [x1; x2] у (градусах) з \n"
```

```
    "кроком dx у (градусах) двома способами: \n вбудованою функцією та "
    "розкладом у ряд Тейлора\n");

 printf("Наші обмеження такі:\n");

 printf("\n%g ≤ x1, x2, dx ≤ %g або \n %g ≤ x1, x2, dx ≤ %g.\n", MIN_DEGREES,
      MAX_DEGREES, -MAX_DEGREES, -MIN_DEGREES);

 printf("Якщо x1 = x2, то dx = 0\n");
 printf("Якщо x1 ≠ x2, то dx ≠ 0\n");
 printf("Якщо x1 < x2, то dx > 0\n");
 printf("Якщо x1 > x2, то dx < 0\n");

 printf("\n%g ≤ e < %g\n", MIN_PRECISION, MAX_PRECISION);
 printf("Довжина x1, x2, dx в символах ≤ %u.\n", MAX_CHARS_DEGREES);
 display_warning(
    "Числа, різниця між якими менша за %g, вважаються рівними. \n"
    "Максимальна гарантована точність 1e-6\n"
    "Введіть '-' при будьякому- вводі для виходу з програми\n",
    TOLERANCE);
}

int main() {
 float x1, x2, dx, e;
 x1 = x2 = dx = e = 0;
 print_demo();

 int is_repeat;

 do {
  is_repeat = 0;
  while (read_float(&x1, "Нижня межа діапазону x1", "x1", MAX_CHARS_DEGREES,
                1, MAX_DEGREES, -MAX_DEGREES, 1, 1, TOLERANCE, 1) != 0 ||
       (x1 > 0 &&
        check_float_meets_restrictions(&x1, "x1", MAX_DEGREES, MIN_DEGREES,
                                1, 1, TOLERANCE) != 0) ||
       (x1 < 0 &&
        check_float_meets_restrictions(
           &x1, "x1", -MIN_DEGREES, -MAX_DEGREES, 1, 1, TOLERANCE) != 0));

  while (read_float(&x2, "Верхня межа діапазону x2", "x2", MAX_CHARS_DEGREES,
                1, MAX_DEGREES, -MAX_DEGREES, 1, 1, TOLERANCE, 1) != 0 ||
       (x2 > 0 &&
        check_float_meets_restrictions(&x2, "x2", MAX_DEGREES, MIN_DEGREES,
                                1, 1, TOLERANCE) != 0) ||
       (x2 < 0 &&
        check_float_meets_restrictions(
           &x2, "x2", -MIN_DEGREES, -MAX_DEGREES, 1, 1, TOLERANCE) != 0));

  while (
     read_float(&dx, "Крок зміни аргументу dx", "dx", MAX_CHARS_DEGREES, 1,
             MAX_DEGREES, -MAX_DEGREES, 1, 1, TOLERANCE, 1) != 0 ||
     (dx > 0 &&
      check_float_meets_restrictions(&dx, "dx", MAX_DEGREES, MIN_DEGREES, 1,
                              1, TOLERANCE) != 0) ||
     (dx < 0 &&
      check_float_meets_restrictions(&dx, "dx", -MIN_DEGREES, -MAX_DEGREES,
                              1, 1, TOLERANCE) != 0) ||
     (x1 != x2 && dx == 0 &&
      display_error("Крок зміни аргумента не може бути нульовим, якщо "
                 "значення меж різні.")) ||
     (x1 == x2 && dx != 0 &&
      display_error("Крок зміни аргумента не може бути не 0, якщо значення "
                 "меж однакові.")) ||
     (x1 < x2 && dx < 0 &&
      display_error(
         "Крок зміни аргумента dx має бути додатнім, якщо x1 < x2.")) ||
     (x1 > x2 && dx > 0 &&
      display_error(
         "Крок зміни аргумента dx має бути відємним', якщо x1 > x2.")));

  while (read_float(&e, "Точність e", "e", MAX_CHARS_PRECISION, 1, 1,
                MIN_PRECISION, 0, 1, TOLERANCE, 1) != 0);

  printf("\n");
  print_separator(COLUMN_WIDTH * COLUMN_COUNT + BORDER_COUNT);
```

```
    printf("|%-*s|%-*s|%-*s|%-*s|\n", COLUMN_WIDTH, "x", COLUMN_WIDTH, "sin(x)",
         COLUMN_WIDTH, "taylor_sin", COLUMN_WIDTH, "diff");
    print_separator(COLUMN_WIDTH * COLUMN_COUNT + BORDER_COUNT);

    int iteration_limit = (dx == 0) ? 1 : (int)fabsf((x1 - x2) / dx) + 1;
    int i = 0;

    if (x1 <= x2) {
      for (float x = x1; x <= x2 && i < iteration_limit; x += dx) {
        print_calculations_row(x, e);
        i++;
      }
    } else {
      for (float x = x1; x >= x2 && i < iteration_limit; x += dx) {
        print_calculations_row(x, e);
        i++;
      }
    }

    print_separator(COLUMN_WIDTH * COLUMN_COUNT + BORDER_COUNT);
    if (request_repeat() == 0) {
      is_repeat = 1;
    };
    print_separator(COLUMN_WIDTH * COLUMN_COUNT + BORDER_COUNT);
  } while (is_repeat);
  return 0;
}
```

## *../src/input/input.h*

```
#ifndef INPUT_H
#define INPUT_H

#define RED "\033[31m"
#define GREEN "\033[32m"
#define YELLOW "\033[33m"
#define RESET "\033[0m"
#define EXIT_KEY '-'

void clear_input();
int request_repeat();
int display_error(const char *format, ...)
    __attribute__((format(printf, 1, 2)));
int display_warning(const char *format, ...)
    __attribute__((format(printf, 1, 2)));
int display_success(const char *format, ...)
    __attribute__((format(printf, 1, 2)));

void replace_commas_with_dots(char *string);
int validate_float_input_precision(const char *input, float value,
                           int max_significant_digits,
                           const char *short_name);

void mock_input(const char *input);

int check_float_meets_restrictions(float *value, const char *name,
                           float max_value, float min_value,
                           int is_max_included, int is_min_included,
                           float tolerance);

int read_float(float *value, const char *full_name, const char *short_name,
            int max_char_count, int is_restricted, float max_value,
            float min_value, int is_max_included, int is_min_included,
            float tolerance, int is_exit);
#endif
```

## *../src/input/common_utils.c*

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

```c
#include <string.h>

#include "input.h"

void replace_commas_with_dots(char *string) {
  while (*string) {
    if (*string == ',') {
      *string = '.';
    }
    string++;
  }
}

int validate_float_input_precision(const char *input, float value,
                                   int max_significant_digits,
                                   const char *short_name) {
  if (value < 1e-6f && value > -1e-6f && value != 0) {
    display_warning(
        "%s має значення з більш ніж 6 знаками після коми. "
        "Точність може бути втрачена.",
        short_name);
    return 1;
  }

  char *dot = strchr(input, '.');
  if (dot != NULL) {
    int decimal_places = 0;
    for (char *p = dot + 1; *p != '\0' && *p != '\n' && *p != 'e' && *p != 'E';
         p++) {
      if (*p >= '0' && *p <= '9') {
        decimal_places++;
      }
    }
    if (decimal_places > max_significant_digits) {
      display_warning(
          "%s має значення з більш ніж %d знаками після коми. "
          "Точність може бути втрачена.",
          short_name, max_significant_digits);
      return 1;
    }
  }
  return 0;
}

void clear_input() {
  int c;
  while ((c = getchar()) != '\n' && c != EOF);
}

int request_repeat() {
  char choice[3];
  printf(
      "Продовжити? Введіть '+' для продовження або будьяку - іншу клавішу, якщо "
      "не погоджуєтесь: ");

  if (!fgets(choice, sizeof(choice), stdin)) {
    printf("Помилка читання вводу.\n");
    return 1;
  }

  if (choice[0] == '+' && choice[1] == '\n') {
    return 0;
  }

  clear_input();

  return 1;
}

int display_error(const char *format, ...) {
  va_list args;
  va_start(args, format);

  printf("\n" RED "ПОМИЛКА! ");
  vprintf(format, args);
  printf(RESET "\n");
```

```
  va_end(args);
  return 1;
}

int display_warning(const char *format, ...) {
  va_list args;
  va_start(args, format);

  printf("\n" YELLOW "УВАГА!␣");
  vprintf(format, args);
  printf(RESET "\n");

  va_end(args);
  return 1;
}

int display_success(const char *format, ...) {
  va_list args;
  va_start(args, format);

  printf("\n" GREEN "ПЕРЕМОГА!␣");
  vprintf(format, args);
  printf(RESET "\n");

  va_end(args);
  return 1;
}
```

## *../src/input/float_utils.c*

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

#include "input.h"

int check_float_meets_restrictions(float *value, const char *name,
                           float max_value, float min_value,
                           int is_max_included, int is_min_included,
                           float tolerance) {
  if (is_min_included) {
    if (*value < min_value - tolerance) {
      display_error("%s␣має␣бути␣більшийрівний-␣%g.", name, min_value);
      return 1;
    }
  } else {
    if (*value <= min_value + tolerance) {
      display_error("%s␣має␣бути␣більший␣за␣%g.", name, min_value);
      return 1;
    }
  }

  if (is_max_included) {
    if (*value > max_value + tolerance) {
      display_error("%s␣має␣бути␣меншийрівний-␣%g.", name, max_value);
      return 1;
    }
  } else {
    if (*value >= max_value - tolerance) {
      display_error("%s␣має␣бути␣менший␣за␣%g.", name, max_value);
      return 1;
    }
  }

  return 0;
}

int read_float(float *value, const char *full_name, const char *short_name,
            int max_char_count, int is_restricted, float max_value,
            float min_value, int is_max_included, int is_min_included,
            float tolerance, int is_exit) {
```

```
  if (is_restricted == 1) {
    printf("%s␣від␣(␣%g␣до␣%g):␣", full_name, min_value, max_value);
  } else {
    printf("%s:␣", full_name);
  }

  char input[max_char_count + 2];
  char *endptr;

  if (!fgets(input, max_char_count + 2, stdin)) {
    display_error("Не␣вдалося␣прочитати␣ввід␣для␣%s.", short_name);
    return 1;
  }

  if (input[strlen(input) - 1] != '\n') {
    display_error("Довжина␣%s␣в␣символах␣має␣бути␣меншою␣за␣%u.", short_name,
                  max_char_count);
    clear_input();
    return 1;
  }

  if (is_exit && input[0] == EXIT_KEY && input[1] == '\n') {
    exit(0);
  }

  replace_commas_with_dots(input);

  *value = strtof(input, &endptr);
  if (endptr == input || *endptr != '\n') {
    display_error("%s␣має␣бути␣числом␣і␣не␣містити␣додаткових␣символів!",
                  short_name);
    return 1;
  }

  if (*value == 0.0f && errno == ERANGE) {
    display_error("%s␣настільки␣малий,␣що␣точність␣не␣може␣бути␣збережена",
                  short_name);
    return 1;
  }

  if (validate_float_input_precision(input, *value, 6, short_name) != 0) {
    if (request_repeat() != 0) {
      return 1;
    }
  }
  if (is_restricted == 1 &&
     check_float_meets_restrictions(value, short_name, max_value, min_value,
                                    is_max_included, is_min_included,
                                    tolerance) == 1) {
    return 1;
  }

  return 0;
}
```

## *../meson.build*

```
project('ln(x␣+␣sqrt(1␣+␣x^2))', 'c')

# Compiler flags and include directories
c_args = ['-I.', '-Iunity/src', '-Isrc', '-g' ]
inc_dirs = include_directories('.', 'unity/src', 'src')

# Main program sources
main_sources = [
  'src/main.c',
  # Input
  'src/input/common_utils.c',
  'src/input/float_utils.c',
]

# Test program sources
test_sources = [
```

```
  # Input
  'src/input/common_utils.c',
  'src/input/float_utils.c',

  # Input tests
  'test/input/test_common_utils.c',
  'test/input/test_float_utils.c',
  'test/test.c',
  'test/input/utils.c',

  # Testing library import
  'unity/src/unity.c',
]

# Define the main executable
executable('main.out', main_sources,
  c_args: c_args,
  include_directories: inc_dirs,
  link_args: ['-lm']
)

# Define the test executable
executable('test.out', test_sources,
  c_args: c_args,
  include_directories: inc_dirs,
  link_args: ['-lm']
)
```

---

## Введені та одержані результати

```
Running test 1
Ця програма знаходить sin(x) для відрізку [x1; x2] (у градусах) з
 кроком dx (у градусах) двома способами:
 вбудованою функцією та розкладом у ряд Тейлора
Наші обмеження такі:

1.17549e-38 ≤ x1, x2, dx ≤ 3.40282e+38 або
 -3.40282e+38 ≤ x1, x2, dx ≤ -1.17549e-38.
Якщо x1 = x2, то dx = 0
Якщо x1 ≠ x2, то dx ≠ 0
Якщо x1 < x2, то dx > 0
Якщо x1 > x2, то dx < 0

1.17549e-38 ≤ e < 1
Довжина x1, x2, dx в символах ≤ 50.

УВАГА! Числа, різниця між якими менша за 1e-07, вважаються рівними.
Максимальна гарантована точність 1e-6
Введіть '-' при будь-якому вводі для виходу з програми

Нижня межа діапазону x1 (від -3.40282e+38 до 3.40282e+38): 1e40

ПОМИЛКА! x1 має бути менший-рівний 3.40282e+38.
Нижня межа діапазону x1 (від -3.40282e+38 до 3.40282e+38): -1e40

ПОМИЛКА! x1 має бути більший-рівний -3.40282e+38.
Нижня межа діапазону x1 (від -3.40282e+38 до 3.40282e+38): 70
Верхня межа діапазону x2 (від -3.40282e+38 до 3.40282e+38): 100
Крок зміни аргументу dx (від -3.40282e+38 до 3.40282e+38): -10

ПОМИЛКА! Крок зміни аргумента dx має бути додатнім, якщо x1 < x2.
Крок зміни аргументу dx (від -3.40282e+38 до 3.40282e+38): 10
Точність e (від 1.17549e-38 до 1): 1e-6

================================================================================
|x                  |sin(x)             |taylor_sin         |diff               |
================================================================================
|70                 |0.939692616462708  |0.939692556858063  |5.96046447753906e-08|
|80                 |0.984807729721069  |0.984807729721069  |0                  |
|90                 |1                  |0.999999940395355  |5.96046447753906e-08|
|100                |0.984807729721069  |0.984807848930359  |-1.19209289550781e-07|
================================================================================
Продовжити? Введіть '+' для продовження або будь-яку іншу клавішу, якщо не погоджуєтесь: --
```

```
Running test 2
Ця програма знаходить sin(x) для відрізку [x1; x2] (у градусах) з
 кроком dx (у градусах) двома способами:
 вбудованою функцією та розкладом у ряд Тейлора
Наші обмеження такі:

1.17549e-38 ≤ x1, x2, dx ≤ 3.40282e+38 або
 -3.40282e+38 ≤ x1, x2, dx ≤ -1.17549e-38.
Якщо x1 = x2, то dx = 0
Якщо x1 ≠ x2, то dx ≠ 0
Якщо x1 < x2, то dx > 0
Якщо x1 > x2, то dx < 0

1.17549e-38 ≤ e < 1
Довжина x1, x2, dx в символах ≤ 50.

УВАГА! Числа, різниця між якими менша за 1e-07, вважаються рівними.
Максимальна гарантована точність 1e-6
Введіть '-' при будь-якому вводі для виходу з програми

Нижня межа діапазону x1 (від -3.40282e+38 до 3.40282e+38): 1000
Верхня межа діапазону x2 (від -3.40282e+38 до 3.40282e+38): 1e40

ПОМИЛКА! x2 має бути менший-рівний 3.40282e+38.
Верхня межа діапазону x2 (від -3.40282e+38 до 3.40282e+38): -1e40

ПОМИЛКА! x2 має бути більший-рівний -3.40282e+38.
Верхня межа діапазону x2 (від -3.40282e+38 до 3.40282e+38): 70
Крок зміни аргументу dx (від -3.40282e+38 до 3.40282e+38): 100

ПОМИЛКА! Крок зміни аргумента dx має бути від'ємним, якщо x1 > x2.
Крок зміни аргументу dx (від -3.40282e+38 до 3.40282e+38): -200
Точність e (від 1.17549e-38 до 1): 1e-8

УВАГА! e має значення з більш ніж 6 знаками після коми. Точність може бути втрачена.
Продовжити? Введіть '+' для продовження або будь-яку іншу клавішу, якщо не погоджуєтесь: +

================================================================================
|x                  |sin(x)              |taylor_sin          |diff               |
================================================================================
|1000               ├0.984807729721069   ├0.984808504581451  |7.74860382080078e-07|
|800                |0.984807729721069   |0.984807729721069   |0                  |
|600                ├0.866025447845459   ├0.866025865077972  |4.17232513427734e-07|
|400                |0.642787575721741   |0.642787635326385   ├5.96046447753906e-08|
|200                ├0.342020392417908   ├0.342020094394684  ├2.98023223876953e-07|
================================================================================
Продовжити? Введіть '+' для продовження або будь-яку іншу клавішу, якщо не погоджуєтесь: --



Running test 3
Ця програма знаходить sin(x) для відрізку [x1; x2] (у градусах) з
 кроком dx (у градусах) двома способами:
 вбудованою функцією та розкладом у ряд Тейлора
Наші обмеження такі:

1.17549e-38 ≤ x1, x2, dx ≤ 3.40282e+38 або
 -3.40282e+38 ≤ x1, x2, dx ≤ -1.17549e-38.
Якщо x1 = x2, то dx = 0
Якщо x1 ≠ x2, то dx ≠ 0
Якщо x1 < x2, то dx > 0
Якщо x1 > x2, то dx < 0

1.17549e-38 ≤ e < 1
Довжина x1, x2, dx в символах ≤ 50.

УВАГА! Числа, різниця між якими менша за 1e-07, вважаються рівними.
Максимальна гарантована точність 1e-6
Введіть '-' при будь-якому вводі для виходу з програми

Нижня межа діапазону x1 (від -3.40282e+38 до 3.40282e+38): 1e7
Верхня межа діапазону x2 (від -3.40282e+38 до 3.40282e+38): 1e8
Крок зміни аргументу dx (від -3.40282e+38 до 3.40282e+38): -1e40

ПОМИЛКА! dx має бути більший-рівний -3.40282e+38.
Крок зміни аргументу dx (від -3.40282e+38 до 3.40282e+38): 1e40

ПОМИЛКА! dx має бути менший-рівний 3.40282e+38.
Крок зміни аргументу dx (від -3.40282e+38 до 3.40282e+38): 0

ПОМИЛКА! dx настільки малий, що точність не може бути збережена
Крок зміни аргументу dx (від -3.40282e+38 до 3.40282e+38): 1e7
Точність e (від 1.17549e-38 до 1): 1e-10

УВАГА! e має значення з більш ніж 6 знаками після коми. Точність може бути втрачена.
Продовжити? Введіть '+' для продовження або будь-яку іншу клавішу, якщо не погоджуєтесь: +

================================================================================
|x                  |sin(x)              |taylor_sin          |diff               |
================================================================================
|10000000           ├0.984807729721069   ├0.984808504581451  |7.74860382080078e-07|
|20000000           ├0.342020392417908   ├0.342020094394684  ├2.98023223876953e-07|
|30000000           |0.866025388240814   |0.86602520942688    |1.78813934326172e-07|
|40000000           |0.642787575721741   |0.642787635326385   ├5.96046447753906e-08|
|50000000           ├0.642787754535675   ├0.642789959907532  |2.20537185668945e-06|
|60000000           ├0.866025447845459   ├0.866025865077972  |4.17232513427734e-07|
|70000000           |0.342020213603973   |0.342020332813263   ├1.19209289550781e-07|
|80000000           |0.984807729721069   |0.984807729721069   |0                  |
|90000000           |0                   |0                   |0                  |
|100000000          ├0.984807729721069   ├0.984808504581451  |7.74860382080078e-07|
================================================================================
Продовжити? Введіть '+' для продовження або будь-яку іншу клавішу, якщо не погоджуєтесь: --
```

```
Running test 4
Ця програма знаходить sin(x) для відрізку [x1; x2] (у градусах) з
 кроком dx (у градусах) двома способами:
 вбудованою функцією та розкладом у ряд Тейлора
Наші обмеження такі:

1.17549e-38 ≤ x1, x2, dx ≤ 3.40282e+38 або
 -3.40282e+38 ≤ x1, x2, dx ≤ -1.17549e-38.
Якщо x1 = x2, то dx = 0
Якщо x1 ≠ x2, то dx ≠ 0
Якщо x1 < x2, то dx > 0
Якщо x1 > x2, то dx < 0

1.17549e-38 ≤ e < 1
Довжина x1, x2, dx в символах ≤ 50.

УВАГА! Числа, різниця між якими менша за 1e-07, вважаються рівними.
Максимальна гарантована точність 1e-6
Введіть '-' при будь-якому вводі для виходу з програми

Нижня межа діапазону x1 (від -3.40282e+38 до 3.40282e+38): 20
Верхня межа діапазону x2 (від -3.40282e+38 до 3.40282e+38): 20
Крок зміни аргументу dx (від -3.40282e+38 до 3.40282e+38): 100

ПОМИЛКА! Крок зміни аргумента не може бути не 0, якщо значення меж однакові.
Крок зміни аргументу dx (від -3.40282e+38 до 3.40282e+38): 0
Точність e (від 1.17549e-38 до 1): 100

ПОМИЛКА! e має бути менший за 1.
Точність e (від 1.17549e-38 до 1): 1

ПОМИЛКА! e має бути менший за 1.
Точність e (від 1.17549e-38 до 1): 1e-12

УВАГА! e має значення з більш ніж 6 знаками після коми. Точність може бути втрачена.
Продовжити? Введіть '+' для продовження або будь-яку іншу клавішу, якщо не погоджуєтесь: +

================================================================================
|x                  |sin(x)             |taylor_sin         |diff               |
================================================================================
|20                 |0.342020124197006  |0.342020124197006  |0                  |
================================================================================
Продовжити? Введіть '+' для продовження або будь-яку іншу клавішу, якщо не погоджуєтесь: --
```

## Теоретичні розрахунки

| $x$ | Вбудований sin | Тейлоровий sin | Точний sin | Похибка вбудованого | Похибка Тейлорового |
|---|---|---|---|---|---|
| 70 | 0.939692616462708 | 0.939692556858063 | 0.939692620785908 | 0.000000004323200 | 0.000000063927845 |
| 80 | 0.984807729721069 | 0.984807729721069 | 0.984807753012208 | 0.000000023291139 | 0.000000023291139 |
| 90 | 1.000000000000000 | 0.999999940395355 | 1.000000000000000 | 0.000000000000000 | 0.000000059604645 |
| 100 | 0.984807729721069 | 0.984807848930359 | 0.984807753012208 | 0.000000023291139 | 0.000000095918151 |
| 1000 | -0.984807729721069 | -0.984808504581451 | -0.984807753012208 | 0.000000023291139 | 0.000000751569243 |
| 800 | 0.984807729721069 | 0.984807729721069 | 0.984807753012208 | 0.000000023291139 | 0.000000023291139 |
| 600 | -0.866025447845459 | -0.866025865077972 | -0.866025403784439 | 0.000000044061020 | 0.000000461293533 |
| 400 | 0.642787575721741 | 0.642787635326385 | 0.642787609686539 | 0.000000033964798 | 0.000000025639846 |
| 200 | -0.342020392417908 | -0.342020094394684 | -0.342020143325669 | 0.000000249092239 | 0.000000048930985 |
| 10000000 | -0.984807729721069 | -0.984808504581451 | -0.984807753015484 | 0.000000023294415 | 0.000000751565967 |
| 20000000 | -0.342020392417908 | -0.342020094394684 | -0.342020143290210 | 0.000000249127698 | 0.000000048895526 |
| 30000000 | 0.866025388240814 | 0.866025209426880 | 0.866025403783635 | 0.000000015542821 | 0.000000194356755 |
| 40000000 | 0.642787575721741 | 0.642787635326385 | 0.642787609628727 | 0.000000033906986 | 0.000000025697658 |
| 50000000 | -0.642787754535675 | -0.642789959907532 | -0.642787609714215 | 0.000000144821460 | 0.000002350193317 |
| 60000000 | -0.866025447845459 | -0.866025865077972 | -0.866025403786045 | 0.000000044059414 | 0.000000461291927 |
| 70000000 | 0.342020213603973 | 0.342020332813263 | 0.342020143285681 | 0.000000070318292 | 0.000000189527582 |
| 80000000 | 0.984807729721069 | 0.984807729721069 | 0.984807752985998 | 0.000000023264929 | 0.000000023264929 |
| 90000000 | 0.000000000000000 | 0.000000000000000 | -0.000000000111596 | 0.000000000111596 | 0.000000000111596 |
| 100000000 | -0.984807729721069 | -0.984808504581451 | -0.984807753024755 | 0.000000023303686 | 0.000000751556696 |
| 20 | 0.342020124197006 | 0.342020124197006 | 0.342020143325669 | 0.000000019128663 | 0.000000019128663 |

***Висновки:*** Теоретичні розрахунки відповідають отриманим. Програма працює коректно. Програма вирішує поставлене завдання. Точність достатньо збереженя (похибка $\leq 10^{-6}$)