**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут: **ІКНІ**

Кафедра: **Пз**

**Звіт**

До лабораторної роботи №7

**На тему: “** ВКАЗІВНИКИ НА ФУНКЦІЇ. РЕКУРСИВНІ ФУНКЦІЇ **”**

**З дисципліни:** «Основи програмування»

**Лектор:**

ст. викл. каф. ПЗ

Муха.Т.О

**Виконав:**

Ст. гр. ПЗ-15

Бабіля О.О.

**Прийняла:**

Асист. каф. Пз

Заводовська Н.О

15.11.2021

∑= \_\_\_\_ .

Львів – 2021

**Тема** **роботи**: ВКАЗІВНИКИ НА ФУНКЦІЇ. РЕКУРСИВНІ ФУНКЦІЇ ПРЕПРОЦЕСОРА.

**Мета** **роботи**: –поглиблене вивчення можливостей функцій в мові С з використанням механізмів рекурсії та вказівників

**Індивідуальне завдання 7-1**

**Використовуючи вищенаведені функції swap та qs\_sort, які реалізують алгоритм швидкого сортування масиву, написати програму мовою С для порівняння ефективності алгоритмів сортування масивів великих обсягів (наприклад, 100000 елементів). Програма повинна також реалізувати один з класичних алгоритмів сортування масиву згідно з варіантом індивідуального завдання. У програмі використати два одинакових масиви, які заповнити випадковими числами, здійснити перевірку впорядкованості елементів масиву, перевірку ідентичності масивів до і після сортування, а також за допомогою стандартної функції time, оцінити час виконання реалізованих алгоритмів сортування.**

7)Сортування в порядку спадання “бульбашковим” методом з додатковою перевіркою чи масив вже відсортований

**Індивідуальне завдання 7-2**

Написати мовою С три функції, щоб протабулювати, задану згідно варіанту, функцію на проміжку [a , b] з кроком h, використавши: а) для першої функції оператор циклу for; б) для другої – оператор циклу while; в) для третьої – – оператор циклу do…while

7)f =(1+sin2 x) 2 ,a=0,b=π/4 ;

**Індивідуальне завдання 6-3**

7)Задати макрос обчислення периметра еліпса за відомими радіусами a та b. Ввести арифметичний вираз із дужками. Використовуючи рекурсивну функцію, перевірити правильність розстановки дужок.

**Теоритичні відомості**

У першій програмі створив 5 функцій: для рондомного заповнення масиву , сортування та перевірки .

Визвав ці функції , та за допомогою функцій з бібліотеки time.h вивів час виконання певних дій.

У другій програмі зробив 3 функції для табулювання 3 різними циклами,визвав ці функції та зробив вибір на табулювання потрібної функції.

У третій програмі створив 3 функції для перевірки кожного типу дужок, за допомогою функцій з бібліотеки stdlib.h перевірив чи є дужки в функціях. В кінці вивів результат перевірки.

**Код програми 1**

Source.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include "function.h"

#define SIZE 10000

void main()

{

int mas[SIZE+1];

int mas1[SIZE+1];

printf("%d\n",randomly(&mas, &mas1));

time\_t t;

time(&t);

printf("Bubble started: %s", ctime(&t));

BUBBLE(mas);

BUBBLE(mas1);

time(&t);

printf("Bubble ended: %s", ctime(&t));

time(&t);

printf("%d\n", COMPEAR(mas, mas1));

randomly(&mas, &mas1);

printf("Default sorting started: %s", ctime(&t));

qs\_sort(mas, 0, 9999);

qs\_sort(mas1, 0, 9999);

time(&t);

printf("Default sorting ended: %s", ctime(&t));

/\*for (int j = 0; j < SIZE - 1; j++)

{

printf("%d ", mas[j]);

}

printf("\n");

for (int j = 0; j < SIZE - 1; j++)

{

printf("%d ", mas1[j]);

}

printf("\n");\*/

printf("%d",COMPEAR(mas, mas1));

}

**Function.c**

**#define \_CRT\_SECURE\_NO\_WARNINGS**

#include "function.h"

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#define SIZE 10000

void BUBBLE(int arr[SIZE])

{

for (int i = 0; i < SIZE - 1; i++)

{

for (int j = 0; j < SIZE - i - 1; j++)

{

if (arr[j] > arr[j + 1])

{

int tmp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = tmp;

}

}

}

//for (int j = 0; j < SIZE - 1; j++)

//{

// printf("%d ", arr[j]);

//}

//printf("\n")ж

}

int randomly(int\* array, int\* array2)

{

int cmp;

for (int i = 0; i < SIZE; i++)

{

array[i] = rand() % 100;

array2[i] = array[i];

if (array[i] == array2[i]) cmp = 1;

else cmp = 0;

}

return cmp;

}

void swap(int array[], long pos1, long pos2)

{

long tmp;

tmp = array[pos1];

array[pos1] = array[pos2];

array[pos2] = tmp;

}

void qs\_sort(int array[], long start, long end)

{

long head = start, tail = end - 1, tmp;

long diff = end - start;

long pe\_index;

// якщо залишилося менше двох елементів – кінець рекурсії

if (diff < 1) return;

if (diff == 1)

if (array[start] > array[end]) {

swap(array, start, end);

return;

}

// пошук індексу розділяючого елементу pe\_index

long m = (start + end) / 2;

if (array[start] <= array[m]) {

if (array[m] <= array[end]) pe\_index = m;

else if (array[end] <= array[start]) pe\_index = start;

else pe\_index = end;

}

else {

if (array[start] <= array[end]) pe\_index = start;

else if (array[end] <= array[m]) pe\_index = m;

else pe\_index = end;

}

long pe = array[pe\_index]; // сам розділяючий елемент

swap(array, pe\_index, end);

while (1) {

while (array[head] < pe)

++head;

while (array[tail] > pe && tail > start)

--tail;

if (head >= tail) break;

swap(array, head++, tail--);

}

swap(array, head, end);

long mid = head;

qs\_sort(array, start, mid - 1); // рекурсивний виклик для 1-ої підмножини

qs\_sort(array, mid + 1, end); // рекурсивний виклик для 2-ої підмножини

}

int COMPEAR(int ar1[SIZE], int ar2[SIZE])

{

int comp;

for (int i = 0; i < SIZE; i++)

{

if (ar1[i] == ar2[i]) comp = 1;

else comp = 0;

}

return comp;

**}**

**Function.h**

**#define FUNCTION\_H**

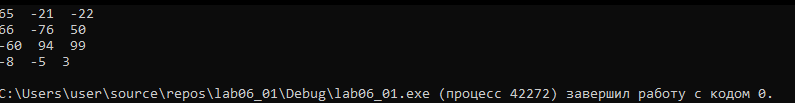
int randomly(int\*, int\*);

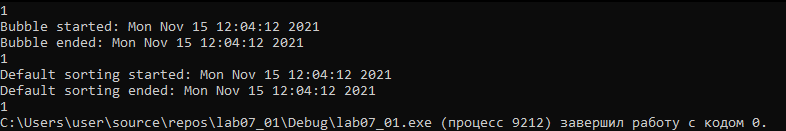
void BUBBLE(int[]);

void qs\_sort(int[], long, long);

void swap(int[], long, long);

int COMPEAR(int[], int[]);

**Результат виконання програми**

**Код програми 2**

**Source.c**

**#include** <stdio.h>

#include <math.h>

#include "function.h"

#define PI 3.14

//

//void TAB\_FOR(double, double, double);

//void TAB\_WHILE(double, double, double);

//void TAB\_DO\_WHILE(double, double, double);

void main()

{

int start = 0,n;

double end = PI / 4,step;

void(\*TAB)(double, double, double);

printf("Enter step:");

scanf\_s("%lf", &step);

printf("Enter 0 for tabulation using for , 1 for while , -1 for do while:");

scanf\_s("%d", &n);

if (n == 0)

{

TAB = &TAB\_FOR;

TAB(start, end, step);

}

else if(n==1)

{

TAB = &TAB\_WHILE;

TAB(start, end, step);

}

else if(n==-1)

{

TAB = &TAB\_DO\_WHILE;

TAB(start, end, step);

}

}

**Fucntion.c**

**#include <stdio.h>**

#include <math.h>

#include "function.h"

#define PI 3.14

void TAB\_FOR(double start, double end, double step)

{

double y = 0;

for (; start < end; )

{

y = pow((1 + sin(2 \* start)), 2.0);

printf("x=%lf y=%lf\n", start, y);

start = start + step;

}

}

void TAB\_WHILE(double start, double end, double step)

{

double y = 0;

while (start < end)

{

y = pow((1 + sin(2 \* start)), 2.0);

printf("x=%lf y=%lf\n", start, y);

start = start + step;

}

}

void TAB\_DO\_WHILE(double start, double end, double step)

{

double y = 0;

do

{

y = pow((1 + sin(2 \* start)), 2.0);

printf("x=%lf y=%lf\n", start, y);

start = start + step;

} while (start < end);

**}**

**Function.h**

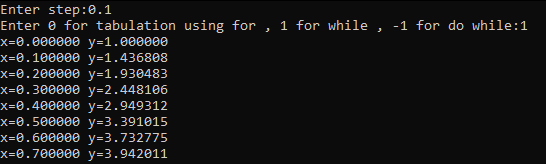
**#define FUNCTION\_H**

void TAB\_FOR(double, double, double);

void TAB\_WHILE(double, double, double);

**void TAB\_DO\_WHILE(double, double, double);**

**Результат виконання програми 2**



**Код програми 3**

Source.c

**#include** <stdio.h>

#include <stdlib.h>

#include "function.h"

#define SIZE 100

void main()

{

int a = 0, b = 0, c = 0, sum = 0;

char str[SIZE];

printf("Enter a sentence:");

gets(str);

if ((strchr(str, ('('))) || (strchr(str, (')'))))

{

a = 1;

sum = a + b + c;

}

if ((strchr(str, ('['))) || (strchr(str, (']'))))

{

const char\* res2 = match2(str);

b = 3;

sum = a + b + c;

}

if ((strchr(str, ('{'))) || (strchr(str, ('}'))))

{

const char\* res3 = match3(str);

c = 6;

sum = a + b + c;

}

if (sum == 0) printf("There are no parentheses ");

else if (sum == 1)

{

const char\* res1 = match1(str);

\*res1 == '\0' ? printf("Good!\n") : printf("Wrong!\n");

}

else if (sum == 3)

{

const char\* res2 = match2(str);

\*res2 == '\0' ? printf("Good!\n") : printf("Wrong!\n");

}

else if (sum == 6)

{

const char\* res3 = match3(str);

\*res3 == '\0' ? printf("Good!\n") : printf("Wrong!\n");

}

else if (sum == 4)

{

int x =0,y=0,z=0;

const char\* res1 = match1(str);

const char\* res2 = match2(str);

if (\*res1 == '\0') x = 1;

if (\*res2 == '\0') y = 1;

z = x + y;

z == 2 ? printf("Good!\n") : printf("Wrong!\n");

}

else if (sum == 7)

{

int x = 0, y = 0, z = 0;

const char\* res1 = match1(str);

const char\* res3 = match3(str);

if (\*res1 == '\0') x = 1;

if (\*res3 == '\0') y = 1;

z = x + y;

z == 2 ? printf("Good!\n") : printf("Wrong!\n");

}

else if (sum == 9)

{

int x = 0, y = 0, z = 0;

const char\* res2 = match2(str);

const char\* res3 = match3(str);

if (\*res2 == '\0') x = 1;

if (\*res3 == '\0') y = 1;

z = x + y;

z == 2 ? printf("Good!\n") : printf("Wrong!\n");

}

else if (sum == 10)

{

int x = 0, y = 0, z = 0,c=0;

const char\* res1 = match1(str);

const char\* res2 = match2(str);

const char\* res3 = match3(str);

if (\*res1 == '\0') c = 1;

if (\*res2 == '\0') x = 1;

if (\*res3 == '\0') y = 1;

z = x + y+c;

z == 2 ? printf("Good!\n") : printf("Wrong!\n");

}

}

**Function.c**

**#include <stdio.h>**

#include <stdlib.h>

#include "function.h"

#define SIZE 100

const char\* match1(const char\* str)

{

if (\*str == '\0' || \*str == ')') { return str; }

if (\*str == '(')

{

const char\* closer = match1(++str);

if (\*closer == ')')

{

return match1(++closer);

}

return str - 1;

}

return match1(++str);

}

const char\* match2(const char\* str)

{

if (\*str == '\0' || \*str == ']') { return str; }

if (\*str == '[')

{

const char\* closer = match2(++str);

if (\*closer == ']')

{

return match2(++closer);

}

return str - 1;

}

return match2(++str);

}

const char\* match3(const char\* str)

{

if (\*str == '\0' || \*str == '}') { return str; }

if (\*str == '{')

{

const char\* closer = match3(++str);

if (\*closer == '}')

{

return match3(++closer);

}

return str - 1;

}

return match3(++str);

**}**

**Function.h**

**#define FUNCTION\_H**

const char\* match1(const char\*);

const char\* match2(const char\*);

**const char\* match3(const char\*);**

**Результат виконання програми 3**



**Висновки:** В ході даної лабораторної роботи я поглиблив знання можливостей функцій в мові С з використанням механізмів рекурсії та вказівників