Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**Звіт**

Про виконання лабораторної роботи №6

на тему:

**«**Особливості роботи з функціями в C.

Директиви препроцесора**»**

з дисципліни «Основи програмування»

**Лектор:**

ст. викл. каф. ПЗ

Муха Т.О.

**Виконав:**

ст. гр. ПЗ-11

Ясногородський Н.В.

**Прийняв:**

асист. каф. ПЗ

Дивак І.В.

« \_\_ » \_\_\_\_\_\_\_\_ 2021 р.

∑ = \_\_\_\_\_ \_\_\_\_\_\_\_\_\_ .

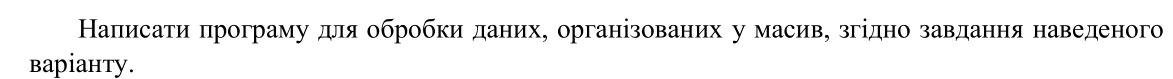
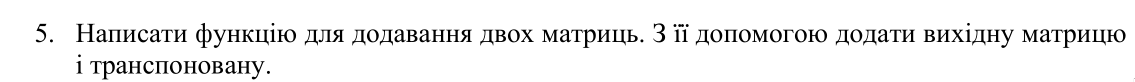
Львів – 2021

**Тема:** Особливості роботи з функціями в C. Директиви препроцесора.

**Мета:** Поглиблене вивчення можливостей функцій в мові С та основ роботи з препроцесором.

ЗАВДАННЯ

***Завдання 1.***



***Завдання 2.***

***Завдання 3.***



ТЕКСТ ПРОГРАМИ

*Завдання 1*

#include <stdio.h>

#include <math.h>

#define \_USE\_MATH\_DEFINES

typedef struct Matrix

{

double \*\*values;

int x;

int y;

} matrix;

void initMatrix(matrix \*matrix, int fillFromStd)

{

matrix->values = (double \*\*)malloc(matrix->y \* sizeof(double \*));

for (int i = 0; i < matrix->y; i++)

{

matrix->values[i] = (double \*)calloc(matrix->x, sizeof(double));

if (fillFromStd)

{

printf("Enter %d elements for %d row:\n", matrix->x, i);

for (int j = 0; j < matrix->x; j++)

{

printf("\tidx %d: ", j);

scanf("%lf", &matrix->values[i][j]);

}

printf("\n");

}

}

}

void freeMatrix(matrix \*matrix)

{

for (int i = 0; i < matrix->x; i++)

{

free(matrix->values[i]);

}

free(matrix->values);

}

void addMatrix(matrix \*a, matrix \*b, matrix \*result)

{

if (a->x != b->x || a->y != b->y)

{

printf("Error: matrix sizes aren't equal!");

return;

}

result->x = a->x;

result->y = a->y;

initMatrix(result, 0);

for (int i = 0; i < a->y; i++)

{

for (int j = 0; j < a->x; j++)

{

result->values[i][j] = a->values[i][j] + b->values[i][j];

}

}

}

void printMatrix(matrix \*matrix, char \*name)

{

printf("\"%s\" matrix:\n", name);

for (int i = 0; i < matrix->y; i++)

{

for (int j = 0; j < matrix->x; j++)

{

printf("\t%.1lf", matrix->values[i][j]);

}

printf("\n");

}

printf("\n");

}

void transposeSquareMatrix(matrix \*initial, matrix \*out)

{

if (initial->x != initial->y)

return;

out->x = initial->x;

out->y = initial->y;

initMatrix(out, 0);

for (int i = 0; i < initial->y; i++)

for (int j = 0; j < initial->x; j++)

out->values[j][i] = initial->values[i][j];

}

int main(void)

{

printf("Task 5, Section 1\n");

printf("Enter matrix heigth and width: ");

matrix a, b, result;

int size = 0;

scanf("%d", &size);

a.x = a.y = size;

initMatrix(&a, 1);

printMatrix(&a, "Initial");

transposeSquareMatrix(&a, &b);

printMatrix(&b, "Transposed");

addMatrix(&a, &b, &result);

printMatrix(&result, "Sum");

freeMatrix(&result);

freeMatrix(&a);

freeMatrix(&b);

return 0;

}

*Завдання 2*

#include <stdio.h>

#include <stdarg.h>

void parseNumber(int);

void calcArgsDigits(int firstNumber, ...)

{

int currentNum = firstNumber;

va\_list args;

va\_start(args, &firstNumber);

// read args while current element not -1

do

{

parseNumber(currentNum);

} while ((currentNum = va\_arg(args, int)) != -1);

va\_end(args);

}

typedef struct DigitFrequency

{

int digit;

int count;

} digitFrequency;

void parseNumber(int number)

{

int numberCopy = number;

digitFrequency map[10];

for (int i = 0; i < 10; i++)

{

map[i].digit = i;

map[i].count = 0;

}

while (numberCopy)

{

int digit = numberCopy % 10;

map[digit].count++;

numberCopy /= 10;

}

printf("Parsing number \"%d\":\n", number);

for (int i = 0; i < 10; i++)

{

if (map[i].count)

{

printf("\t\"%d\" used %d times\n", map[i].digit, map[i].count);

}

}

}

int main(void)

{

printf("Task 5, Section 2\n");

calcArgsDigits(1000, 22394, 3393939, 4999, 51111, 90756, -1);

return 0;

}

*Завдання 3*

#include <stdio.h>

#include <math.h>

// Задати макрос обчислення n - ого члена геометричної прогресії за введеними користувачем першим членом і знаменником

#define GET\_N\_GEOM\_PROGRESSION\_ELEMENT(firstEl, base, n) firstEl \*pow(base, n - 1)

int main(void)

{

printf("Task 5, Section 3\n");

double base, firstElement, idx;

printf("Enter base, first element, idx of desired element (space-separated): ");

scanf("%lf %lf %lf", &base, &firstElement, &idx);

printf("Result: %.3lf\n", GET\_N\_GEOM\_PROGRESSION\_ELEMENT(firstElement, base, idx));

return 0;

}

РЕЗУЛЬТАТИ

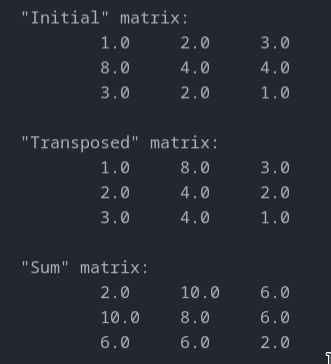


Рис 1. Результат виконання програми №1

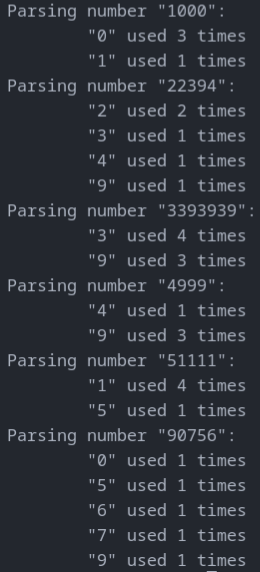


Рис 2. Результат виконання програми №2



Рис 3. Результат виконання програми №3

ВИСНОВКИ

Здобуто практичні навички створення та застосування функцій та макросів у мові С.