Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

Кафедра радіоелектронних і комп’ютерних систем

Звіт

Про виконання лабораторної роботи №4

“ Інструментальні засоби системного програмування”

Виконала:

Студентка групи ФеІ-23

Лісова С.О.

Викладач:

Сінькевич О.О.

Львів – 2019

**Тема:** Інструментальні засоби системного програмування.

**Мета:** Освоєння засобів створення системного програмного забезпечення.

**Завдання №1.** Створити програму на мові С, в якій наведено приклади використання п’яти функцій для роботи із рядковими змінними.

**Хід роботи:**

1. Створюю текстовий файл part1.c, у якому прописую наступний код:

#include <stdio.h>

#include <string.h>

int main ()

{

char array1[20] = "";

char array2[20] = "";

printf("Input your first word: ");

scanf("%s", array1);

printf("Input your second word: ");

scanf("%s", array2);

printf("\nYour words are:\n");

printf("%s\t", array1);

printf("%s\n", array2);

printf("Number of characters in each word: " );

printf("%lu,\t",strlen(array1));

printf("%lu\n",strlen(array2));

//Comparing two lines in the alphabetical order:

if (strcmp(array1, array2) > 0)

printf("The first word is located after the second one in the alphabetical order.\n");

else if (strcmp(array1, array2) < 0)

printf("The first word is located before the second one in the alphabetical order.\n");

else

printf("Two words are the same.\n");

printf("Combination of your words is: ");

strcat(array1, array2);

printf("%s\n\n", array1);

printf("Copying your second word and putting it on first word's place...\n");

printf("Now two of your words look like: \n");

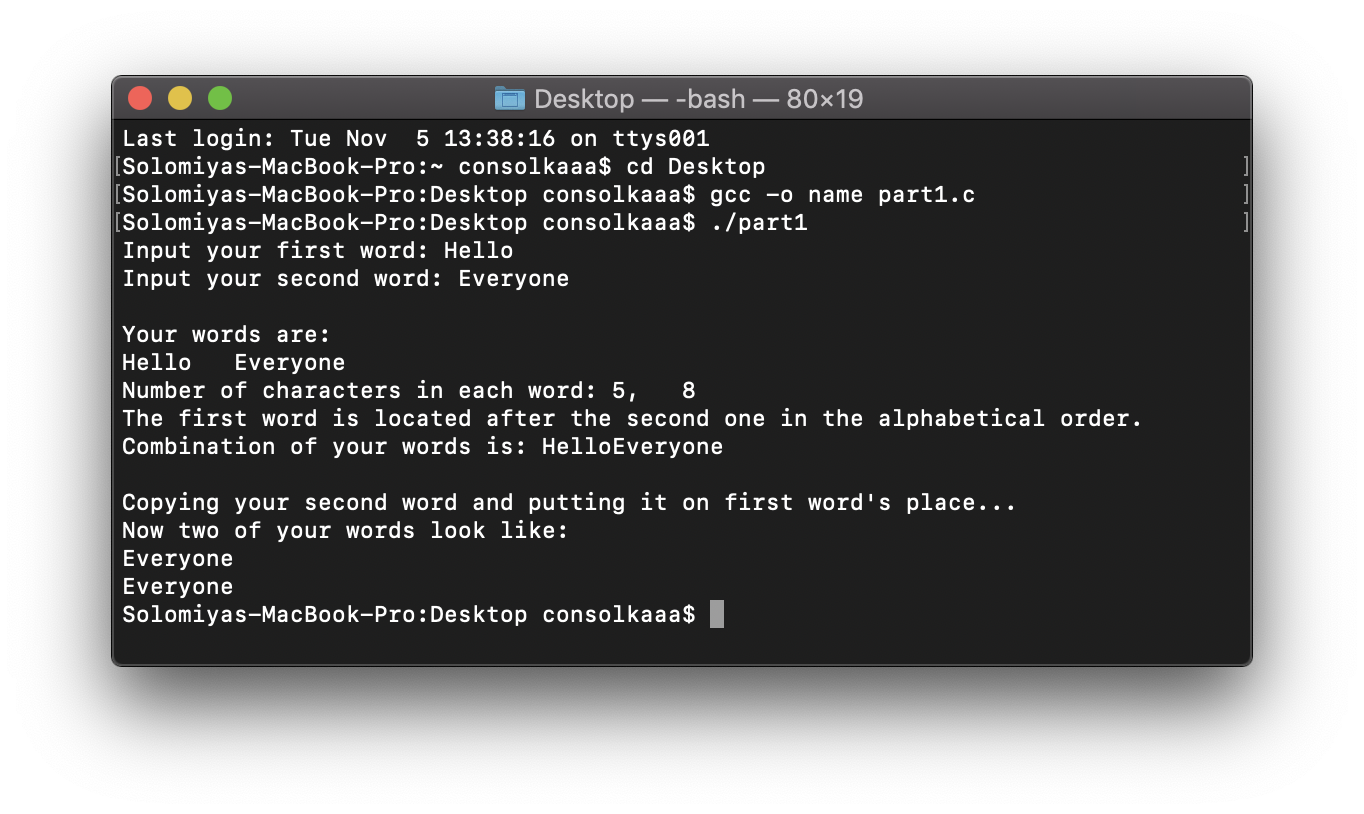
strcpy(array1, array2);

printf("%s\n",array1);

printf("%s\n",array2);

}

1. Зберігаємо текстовий файл. Компілюю його за допомогою команди gcc –o name <filename>
2. Запускаємо програму командою ./part1



**Завдання №2.** Створити програмний проект з головною програмою і мінімум двома функціями в окремих файлах і зібрати його за допомогою утиліти make.

**Хід роботи:**

1. Отже, нам потрібно створити кілька окремих файлів. Моя програма - це калькулятор, основні функції якого реалізовані в файлі Functions.c, а ще одна функція для знаходження факторіалу прописана у файлі Factorial.c.

* Main.c:

#include <stdio.h>

#include <stdlib.h>

float sum(float n1, float n2);

float riz(float n1, float n2);

float dobutok(float n1, float n2);

float chastka(float n1, float n2);

float factorial(int n);

int main(){

float n1=0;

float n2=0;

float n=0;

int a=0;

printf("Sum - 0\n");

printf("Riz - 1\n");

printf("Dobutok - 2\n");

printf("Chastka - 3\n");

printf("Factorial - 4\n");

printf("Select fuction:\n ");

scanf("%d",&a);

switch(a){

case 0:

printf("Input first integer number : ");

scanf("%f",&n1);

printf("Input second integer number : ");

scanf("%f",&n2);

printf("Sum: %f\n ", sum(n1,n2));

break;

case 1:

printf("Input first integer number : ");

scanf("%f",&n1);

printf("Input second integer number : ");

scanf("%f",&n2);

printf("Riz: %f\n ", riz(n1,n2));

break;

case 2:

printf("Input first integer number : ");

scanf("%f",&n1);

printf("Input second integer number : ");

scanf("%f",&n2);

printf("Dobutok: %f\n ", dobutok(n1,n2));

break;

case 3:

printf("Input first integer number : ");

scanf("%f",&n1);

printf("Input second integer number : ");

scanf("%f",&n2);

printf("Chastka: %f\n ", chastka(n1,n2));

break;

case 4:

printf("Input n : ");

scanf("%f",&n);

printf("Factorial: %f\n ",factorial(n));

break;

}

return 0;

}

* Functions.c:

float sum(float n1, float n2){

return n1+n2;

}

float riz(float n1, float n2){

return n1-n2;

}

float dobutok(float n1, float n2){

return n1\*n2;

}

float chastka(float n1, float n2){

return n1/n2;

}

* Factorial.c:

float factorial(int n){

int a = 1;

if(n==0){

return 1;

}

for(int i=1;i<=n;i++){

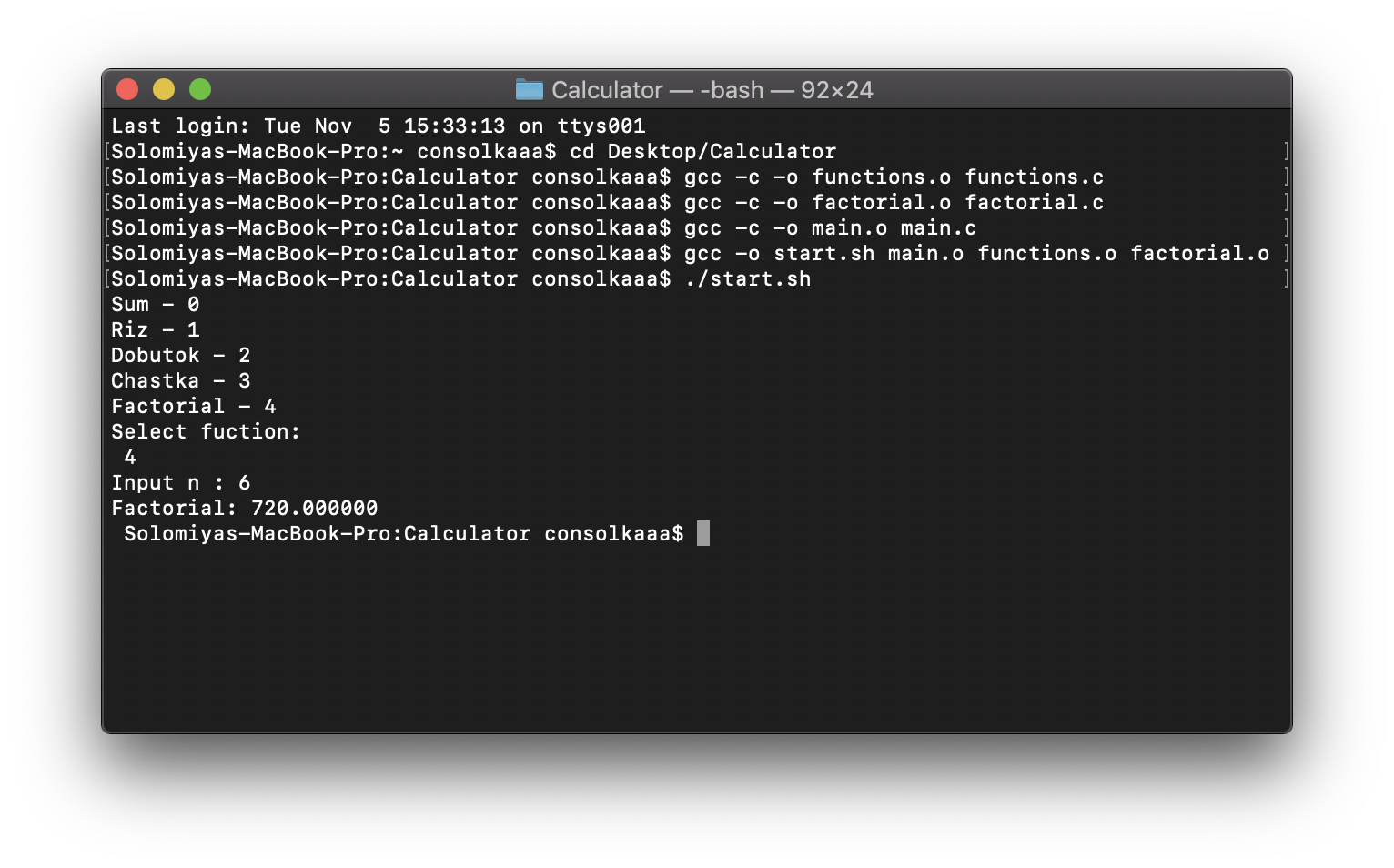
a=a\*i;

}

return a;

}

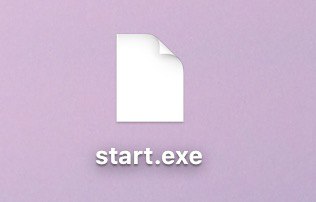
1. Відкомпільовую окремо кожен файл мовою Сі у відповідний об'єктний файл:

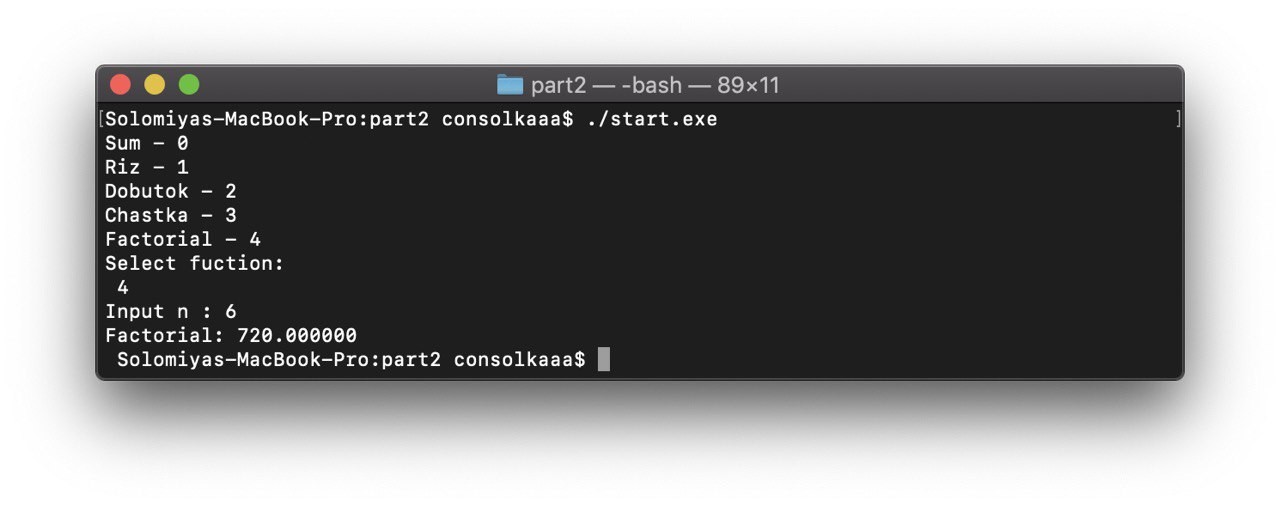


1. Команда $ gcc -o name.exe name1.o name2.o name3.o виконує збирання створених об'єктних файлів у виконуваний файл з іменем name.exe



1. В результаті отримаємо готову до виконання програму та запустимо її





**Завдання №3.** Створити програму на мові С, в якій наведено приклади використання аргументів програми та коротких і довгих опцій.

**Хід роботи:**

1. Створюємо файл options.c із наступним кодом:

#include <stdio.h>

#include <stdlib.h>

#include <getopt.h>

#include <stdbool.h>

int main(int argc, char \*\*argv){

int opt;

struct option long\_options[] =

{

{"f", 0, 0,'f'},

{"j", 0, 0, 'j'},

{"add", 1, 0, 'a'},

{0, 0, 0, 0}

};

printf("Program started!\n");

while(true){

int option\_index;

opt = getopt\_long(argc, argv, "fjz:r", long\_options, &option\_index);

if(opt == -1)

break;

if(opt == 0){

printf("%s: used long option --%s", argv[0], long\_options[option\_index].name);

if(optarg)

printf("with argument: %s", optarg);

printf("\n");

}

else

if(opt == 'f'){

printf("Option \'f\' used\n");

printf("f means friend\n");

}

else

if(opt == 'j'){

printf("Option \'j\' used\n");

printf("j means joke\n");

}

else

if(opt == 'z'){

printf("Optionz \'z\' used with argument: %s\n", optarg);

}

else

if(opt == 'r'){

printf("Option \'r\' used\n");

}

else

if(opt == 'a'){

printf("Long option --add is used with arg: %s\n", optarg);

}

else{

printf("Incorrect option!\n");

break;

}

}

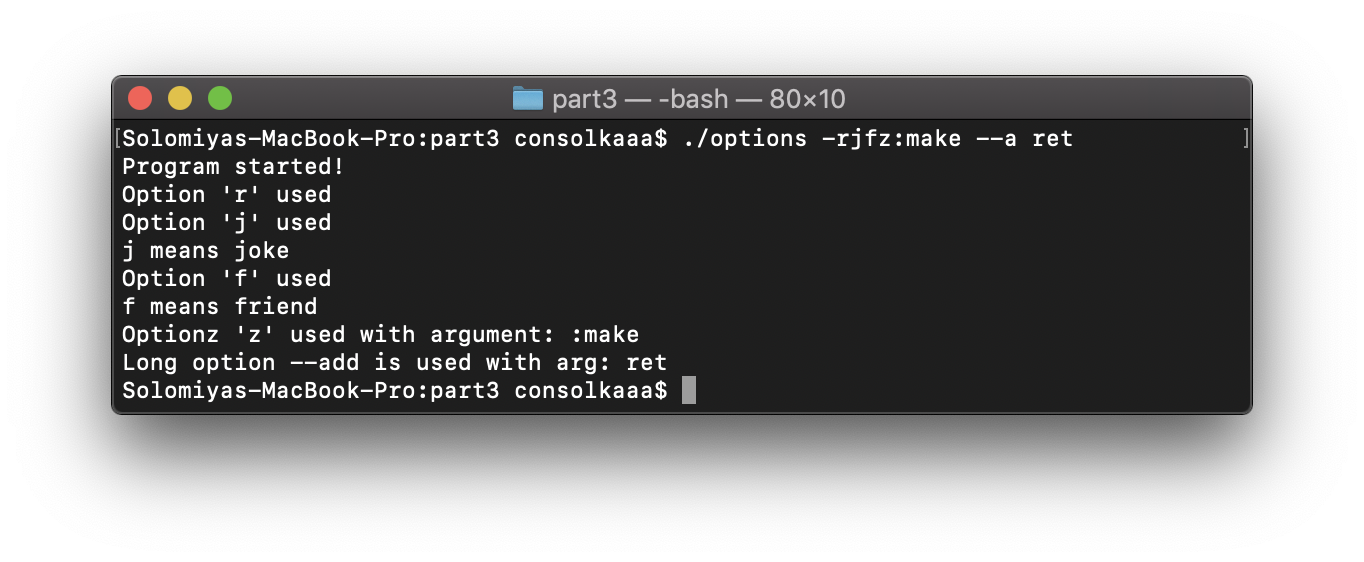
return 0;

}

У цій програмі є короткі опції f, j, r, а також коротка опція z з аргументом. Додатково присутня довга опція a з агрументом.

1. Запуск та робота програми





**Висновок:** Під час виконання лабораторної роботи №4 я навчилася використовувати функції для роботи із символьними рядками, cтворила програмний проект і зібрала його за допомогою утиліти make, а також створила програму на мові С, в якій наведено приклади використання аргументів програми та коротких і довгих опцій.