**ЛАБОРАТОРНА РОБОТА № 24**

**Робота з рядками**

**Мета:** отримання практичних навиків роботи з рядками.

**Хід роботи:**

**Для обробки рядків використовувати стандартні функції з бібліотечного файлу.**

**Функція main() повинна містити тільки опис рядків і виклики функцій для формування, виводу на екран і обробки рядків.**

**Завдання 1.** Ввести з клавіатури рядок символів і обробити його відповідно до свого варіанту, використовуючи функції.

****

Лістинг програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

#include <windows.h>

void printstrings(char string1[]) {

printf("%s\n", string1);

}

void task(char string1[], char stringvowels[10][2]) {

for (int i = 0; i < strlen(string1); i++) {

for (int j = 0; j < 10; j++) {

if (string1[i] == stringvowels[j][0]) {

for (int k = i; k < strlen(string1); k++) {

string1[k] = string1[k + 1];

}

}

}

}

printstrings(string1);

}

int main() {

SetConsoleCP(1251); SetConsoleOutputCP(1251);

// Тема завдання

printf("Завдання 1:\nВвести з клавіатури рядок символів і обробити його\nвідповідно до свого варіанту, використовуючи функції:\n1) Видалити всі голосні літери з рядка.\n");

char string1[16] = "Andrey Babushko";

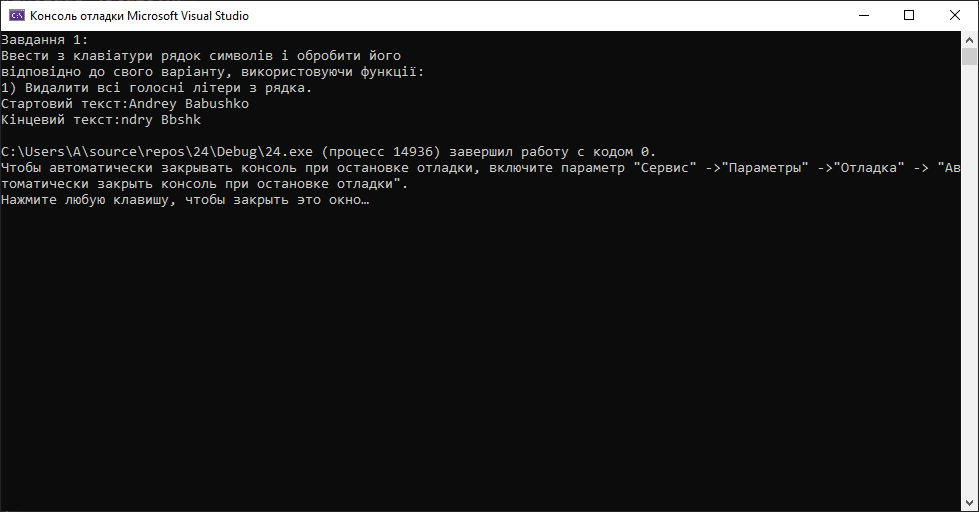
char stringvowels[10][2] = {"A", "E", "I", "O", "U", "a", "e", "i", "o", "u" };

printf("Стартовий текст:"); printstrings(string1);

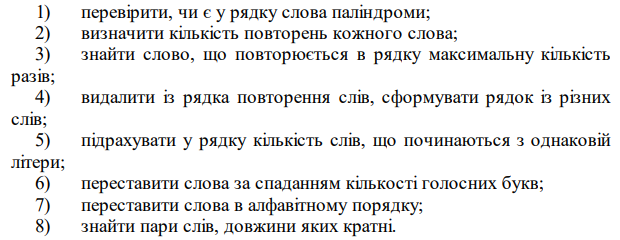
printf("Кінцевий текст:"); task(string1, stringvowels);

}

Результат програми:



**Завдання 2.** Дано рядок зі стандартним набором символів-роздільників між словами. Необхідно:

****

Лістинг програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

#include <windows.h>

void printstrings(char string[], int indexstring) {

if (indexstring == 1) printf(" %s\n", string);

if (indexstring == 2) printf(" %s\n", string);

if (indexstring == 3) printf(" %s\n", string);

if (indexstring == 4) printf(" %s\n", string);

if (indexstring == 5) printf(" %s\n", string);

if (indexstring == 6) printf(" %s\n", string);

if (indexstring == 7) printf(" %s\n", string);

if (indexstring == 8) printf(" %s\n", string);

}

void task1(char string1[]) {

int countrev = 0, raw = 0;

char\* string1p;

char temp\_string[20];

char tempstring1[20][20] = { 0 };

string1p = strtok(string1, " \_-:;().,/!?");

while (string1p != NULL) {

for (int i = 0; i < strlen(string1p); i++) {

tempstring1[raw][i] = \*(string1p + i);

}

strcpy(temp\_string, tempstring1[raw]);

\_strrev(temp\_string);

if (strcmp(temp\_string, tempstring1[raw]) == 0) {

countrev++;

}

raw++;

string1p = strtok(NULL, " \_-:;().,/!?");

}

if (countrev > 0) {

printf("У тексті є паліндроми. Кількість паліндромів у тексті: %d\n", countrev);

}

else {

printf("У тексті немає паліндромів.\n");

}

}

void task2(char string2[]) {

char tempstring2[50][50] = { 0 }, temp2string2[50][50] = { 0 }, \* string2p;

int raw = 0, count\_words = 0, count2\_words = 0, count\_repeated = 0;

char copy\_string[100] = { 0 };

strcpy(copy\_string, string2); // копіювання рядка. щоб не втратити значення при роботі

string2p = strtok(copy\_string, " \_-:;().,/!?"); // ділення рядок на слова

while (string2p != NULL) {

strcpy(tempstring2[raw], string2p);

raw++;

count\_words = raw;

string2p = strtok(NULL, " \_-:;().,/!?");

}

for (int i = 0; i < count\_words; i++) { // ще одне копіювання

strcpy(temp2string2[i], tempstring2[i]);

}

count2\_words = count\_words; // копіювання рядка. щоб не втратити значення при роботі

for (int i = 0; i < count2\_words; i++) { // цикл на запис слів без повторювань

for (int j = i; j < count2\_words; j++)

{

if (i != j && strcmp(temp2string2[i], temp2string2[j]) == 0)

{

for (int k = i; k <= count2\_words; k++)

{

strcpy(temp2string2[k], temp2string2[k + 1]);

}

count2\_words--;

}

}

}

for (int i = 0; i < count2\_words; i++) // цикл для обчислення кількості повторювань слова

{

for (int j = 0; j < count\_words; j++)

{

if (strcmp(temp2string2[i], tempstring2[j]) == 0)

{

count\_repeated++;

}

}

printf("Слово %s, яке повторюється %d разів.\n", temp2string2[i], count\_repeated);

count\_repeated = 0;

}

}

void task3(char string3[]) {

char tempstring3[50][50] = { 0 }, temp2string3[50][50] = { 0 }, \* string3p;

int raw = 0, count\_words = 0, count2\_words = 0, count\_repeated = 0, count\_repeated\_arr[100] = { }, max = 0, max\_index = 0;

char copy\_string[100] = { 0 };

strcpy(copy\_string, string3); // копіювання рядка. щоб не втратити значення при роботі

string3p = strtok(copy\_string, " \_-:;().,/!?"); // ділення рядок на слова

while (string3p != NULL) {

strcpy(tempstring3[raw], string3p);

raw++;

count\_words = raw;

string3p = strtok(NULL, " \_-:;().,/!?");

}

for (int i = 0; i < count\_words; i++) { // ще одне копіювання

strcpy(temp2string3[i], tempstring3[i]);

}

count2\_words = count\_words; // копіювання рядка. щоб не втратити значення при роботі

for (int i = 0; i < count2\_words; i++) { // цикл на запис слів без повторювань

for (int j = i; j < count2\_words; j++)

{

if (i != j && strcmp(temp2string3[i], temp2string3[j]) == 0)

{

for (int k = i; k <= count2\_words; k++)

{

strcpy(temp2string3[k], temp2string3[k + 1]);

}

count2\_words--;

}

}

}

for (int i = 0; i < count2\_words; i++) // цикл для переписування кількості повторень в масив(індекс слова і його кількість повторювань)

{

for (int j = 0; j < count\_words; j++)

{

if (strcmp(temp2string3[i], tempstring3[j]) == 0)

{

count\_repeated++;

}

}

count\_repeated\_arr[i] = count\_repeated;

count\_repeated = 0;

}

for (int i = 0; i < count2\_words; i++) // пошук слова з максимальним повторюванням

{

if (count\_repeated\_arr[i] > max)

{

max = count\_repeated\_arr[i];

max\_index = i;

}

}

printf("Слово %s яке повторюється %d разів.\n", temp2string3[max\_index], max);

}

void task4(char string4[], int indexstring) {

char tempstring4[100][100] = { 0 }, temp2string4[100][100] = { }, copy\_string[1000] = { 0 }, \* string4p;

int raw = 0, count\_words = 0, count2\_words = 0;

strcpy(copy\_string, string4); // копіювання рядка. щоб не втратити значення при роботі

string4p = strtok(copy\_string, " \_-:;().,/!?"); // ділення рядок на слова

while (string4p != NULL) {

strcpy(tempstring4[raw], string4p);

raw++;

count\_words = raw;

string4p = strtok(NULL, " \_-:;().,/!?");

}

for (int i = 0; i < count\_words; i++) { // ще одне копіювання

strcpy(temp2string4[i], tempstring4[i]);

}

count2\_words = count\_words; // копіювання рядка. щоб не втратити значення при роботі

for (int i = 0; i < count2\_words; i++) // цикл на запис слів без повторювань

{

for (int j = i; j < count2\_words; j++) {

if (i != j && strcmp(temp2string4[i], temp2string4[j]) == 0)

{

for (int k = i; k <= count2\_words; k++)

{

strcpy(temp2string4[k], temp2string4[k + 1]);

}

count2\_words--;

}

}

}

for (int i = 0; i < count2\_words; i++) { // ще одне копіювання

printf("%s ", temp2string4[i]);

}

}

void task5(char string5[]) {

char tempstring5[100][100] = { 0 }, temp2string5[100][100] = {}, copy\_string[1000] = { 0 }, \* string5p;

int raw = 0, count\_words = 0, count2\_words = 0;

strcpy(copy\_string, string5); // копіювання рядка. щоб не втратити значення при роботі

string5p = strtok(copy\_string, " \_-:;().,/!?"); // ділення рядок на слова

while (string5p != NULL) {

strcpy(tempstring5[raw], string5p);

raw++;

count\_words = raw;

string5p = strtok(NULL, " \_-:;().,/!?");

}

for (int i = 0; i < count\_words; i++) { // ще одне копіювання

strcpy(temp2string5[i], tempstring5[i]);

}

count2\_words = count\_words; // копіювання рядка. щоб не втратити значення при роботі

for (int i = 0; i < count2\_words; i++) // цикл на запис слів без повторювань

{

for (int j = i; j < count2\_words; j++) {

if (i != j && strcmp(temp2string5[i], temp2string5[j]) == 0)

{

for (int k = i; k <= count2\_words; k++)

{

strcpy(temp2string5[k], temp2string5[k + 1]);

}

count2\_words--;

}

}

}

char index\_first\_symb[100] = { };

for (int i = 0; i < count\_words; i++) // цикл на запис 1 літери слів у масив

{

index\_first\_symb[i] = tempstring5[i][0];

}

count2\_words = count\_words;

for (int i = 0; i < count2\_words; i++) // видалення однакових літер слів з масиву перших літер слів

{

for (int j = 0; j < count2\_words; j++)

{

if (i != j && index\_first\_symb[i] == index\_first\_symb[j])

{

for (int k = i; k < count2\_words; k++)

{

index\_first\_symb[k] = index\_first\_symb[k + 1];

}

count2\_words--;

}

}

}

int final\_count = 0;

for (int i = 0; i < count2\_words; i++) // обрахунок кількості слів які починаються з 1 літери

{

for (int j = 0; j < count\_words; j++)

{

if (index\_first\_symb[i] == tempstring5[j][0])

{

final\_count++;

}

}

printf("\nПерша літера ---> %c | %d <--- Кількість слів, що починаються з літери", index\_first\_symb[i], final\_count);

final\_count = 0;

}

}

void task6(char string6[]) {

char tempstring6[100][100] = { 0 }, temp2string6[100][100] = {}, copy\_string[1000] = { 0 }, \* string6p;

int raw = 0, count\_words = 0, count2\_words = 0;

strcpy(copy\_string, string6); // копіювання рядка. щоб не втратити значення при роботі

string6p = strtok(copy\_string, " \_-:;().,/!?"); // ділення рядок на слова

while (string6p != NULL) {

strcpy(tempstring6[raw], string6p);

raw++;

count\_words = raw;

string6p = strtok(NULL, " \_-:;().,/!?");

}

for (int i = 0; i < count\_words; i++) { // ще одне копіювання

strcpy(temp2string6[i], tempstring6[i]);

}

count2\_words = count\_words; // копіювання рядка. щоб не втратити значення при роботі

for (int i = 0; i < count2\_words; i++) // цикл на запис слів без повторювань

{

for (int j = i; j < count2\_words; j++) {

if (i != j && strcmp(temp2string6[i], temp2string6[j]) == 0)

{

for (int k = i; k <= count2\_words; k++)

{

strcpy(temp2string6[k], temp2string6[k + 1]);

}

count2\_words--;

}

}

}

char vowels[] = "eyuioaERYUIOA";

}

void task7(char string7[]) {

char tempstring7[100][100] = { 0 }, temp2string7[100][100] = {}, copy\_string[1000] = { 0 }, \* string7p;

int raw = 0, count\_words = 0, count2\_words = 0;

strcpy(copy\_string, string7); // копіювання рядка. щоб не втратити значення при роботі

string7p = strtok(copy\_string, " \_-:;().,/!?"); // ділення рядок на слова

while (string7p != NULL) {

strcpy(tempstring7[raw], string7p);

raw++;

count\_words = raw;

string7p = strtok(NULL, " \_-:;().,/!?");

}

for (int i = 0; i < count\_words; i++) { // ще одне копіювання

strcpy(temp2string7[i], tempstring7[i]);

}

count2\_words = count\_words; // копіювання рядка. щоб не втратити значення при роботі

for (int i = 0; i < count2\_words; i++) // цикл на запис слів без повторювань

{

for (int j = i; j < count2\_words; j++) {

if (i != j && strcmp(temp2string7[i], temp2string7[j]) == 0)

{

for (int k = i; k <= count2\_words; k++)

{

strcpy(temp2string7[k], temp2string7[k + 1]);

}

count2\_words--;

}

}

}

}

void task8(char string8[]) {

char tempstring8[100][100] = { 0 }, temp2string8[100][100] = {}, copy\_string[1000] = { 0 }, \* string8p;

int raw = 0, count\_words = 0, count2\_words = 0;

strcpy(copy\_string, string8); // копіювання рядка. щоб не втратити значення при роботі

string8p = strtok(copy\_string, " \_-:;().,/!?"); // ділення рядок на слова

while (string8p != NULL) {

strcpy(tempstring8[raw], string8p);

raw++;

count\_words = raw;

string8p = strtok(NULL, " \_-:;().,/!?");

}

for (int i = 0; i < count\_words; i++) { // ще одне копіювання

strcpy(temp2string8[i], tempstring8[i]);

}

count2\_words = count\_words; // копіювання рядка. щоб не втратити значення при роботі

for (int i = 0; i < count2\_words; i++) // цикл на запис слів без повторювань

{

for (int j = i; j < count2\_words; j++) {

if (i != j && strcmp(temp2string8[i], temp2string8[j]) == 0)

{

for (int k = i; k <= count2\_words; k++)

{

strcpy(temp2string8[k], temp2string8[k + 1]);

}

count2\_words--;

}

}

}

}

int main() {

SetConsoleCP(1251); SetConsoleOutputCP(1251);

// Тема завдання

printf("Завдання 2:\nДано рядок зі стандартним набором символів-роздільників між словами. Необхідно:");

int indexstring = 1;

//1)

printf("\n1) перевірити, чи є у рядку слова паліндроми;");

char string1[100] = { 0 };

printf("\nВведіть текст(1):"); gets\_s(string1);

printf("Текст до змін(1):"); printstrings(string1, indexstring);

task1(string1);

indexstring++;

//2)

printf("\n2) визначити кількість повторень кожного слова;");

char string2[100] = { 0 };

printf("\nВведіть текст(2):"); gets\_s(string2);

printf("Текст до змін(2):"); printstrings(string2, indexstring);

task2(string2);

indexstring++;

//3)

printf("\n3) знайти слово, що повторюється в рядку максимальну кількість разів;");

char string3[100] = { 0 };

printf("\nВведіть текст(3):"); gets\_s(string3);

printf("Текст до змін(3):"); printstrings(string3, indexstring);

task3(string3);

indexstring++;

//4)

printf("\n4) видалити із рядка повторення слів, сформувати рядок із різних слів;");

char string4[100] = { 0 };

printf("\nВведіть текст(4):"); gets\_s(string4);

printf("Текст до змін(4):"); printstrings(string4, indexstring);

task4(string4, indexstring);

indexstring++;

//5)

printf("\n5) підрахувати у рядку кількість слів, що починаються з однаковій літери;");

char string5[100] = { 0 };

printf("\nВведіть текст(5):"); gets\_s(string5);

printf("Текст до змін(5):"); printstrings(string5, indexstring);

task5(string5);

indexstring++;

//6)

printf("\n6) переставити слова за спаданням кількості голосних букв;");

char string6[100] = { 0 };

printf("\nВведіть текст(6):"); gets\_s(string6);

printf("\nТекст до змін(6):"); printstrings(string6, indexstring);

task6(string6);

indexstring++;

//7)

printf("\n7) переставити слова в алфавітному порядку;");

char string7[100] = { 0 };

printf("\nВведіть текст(7):"); gets\_s(string7);

printf("Текст до змін(7):"); printstrings(string7, indexstring);

task7(string7);

indexstring++;

//8)

printf("\n8) знайти пари слів, довжини яких кратні.");

char string8[100] = { 0 };

printf("\nВведіть текст(8):"); gets\_s(string8);

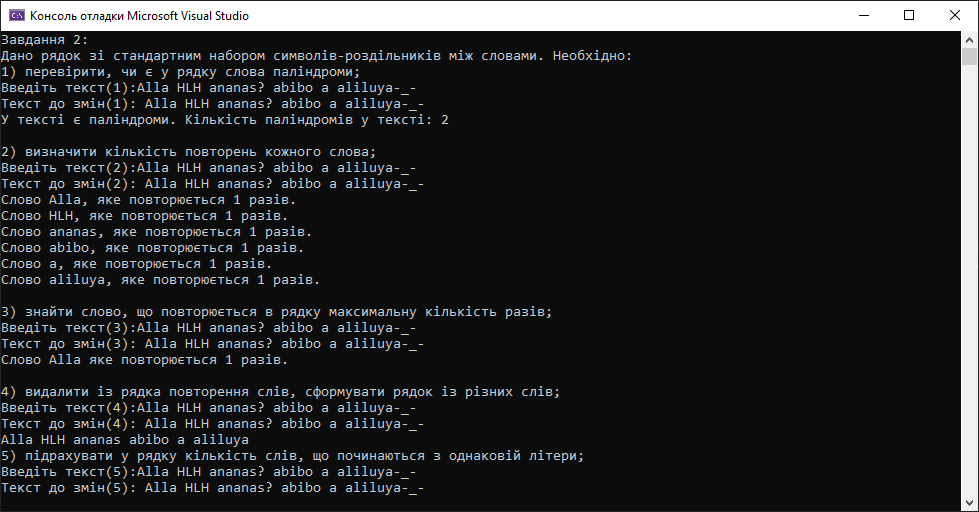
printf("Текст до змін(8):"); printstrings(string8, indexstring);

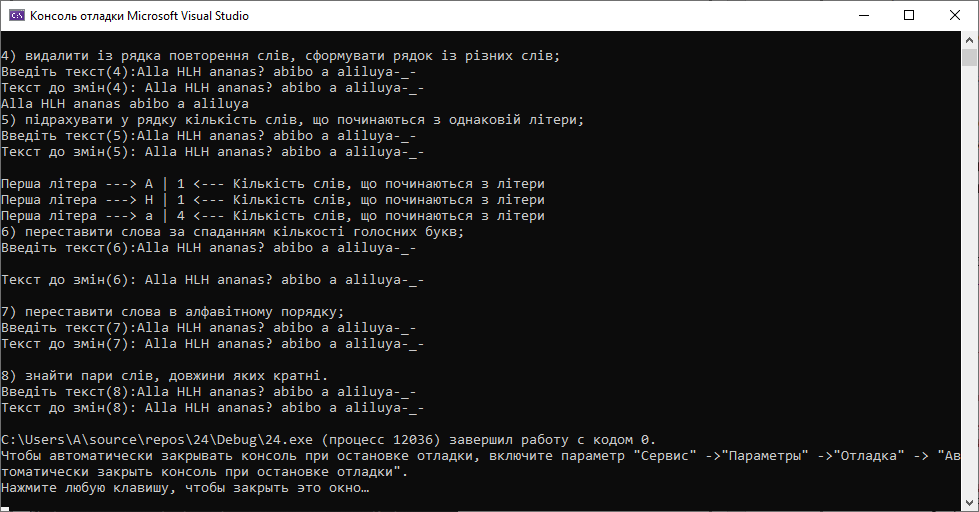
task8(string8);

indexstring++;

}

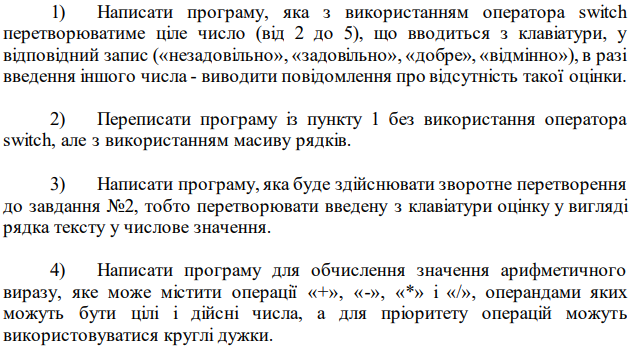
Результат програми:





**Самостійна робота**

**Завдання.**

****

**Завдання.**

Лістинг програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

#include <windows.h>

void task1() {

printf("Завдання 1: Написати програму, яка з використанням оператора switch перетворювати-ме ціле число(від 2 до 5),\nщо вводиться з клавіатури, увідповідний запис(«незадовільно», «за-довільно», «добре», «відмінно»),\nв разі введення іншого числа - виводити повідомлення про відсутність такої оцінки.\n");

int mark;

do {

printf("Введіть оцінку(від 2 - 5) - (0 - вихід з завдання) ---> "); scanf("%d", &mark);

switch (mark) {

case 2: {

printf("Не задовільно! // Вчись добре!\n");

break;

}

case 3: {

printf("Задовільно! // Бабушко ідьош на пятьорку.\n");

break;

}

case 4: {

printf("Добре! // Ідьош на 200% IQ\n");

break;

}

case 5: {

printf("Відмінно! // А чого не 6?\n");

break;

}

default: {

printf("Така оцінка відсутня! // Та за шо(((\n");

}

}

} while (mark != 0);

}

void task2() {

char mark\_string[10], mark\_arr[5][2] = { {"2"},{"3"},{"4"},{"5"} }, mark\_names\_marks[4][100] = { {"Не задовільно! // Вчись добре!"}, {"Задовільно! // Бабушко ідьош на пятьорку."}, {"Добре! // Ідьош на 200% IQ"}, {"Відмінно! // А чого не 6?"} };

printf("Завдання 2: Переписати програму із пункту 1 без використання оператора switch,\nале з використанням масиву рядків.");

do {

printf("Введіть оцінку(від 2 - 5) - (0 - вихід з завдання) ---> "); scanf("%s", mark\_string);

for (int i = 0; i < 5; i++) {

if (strcmp(mark\_arr[i], mark\_string) == 0) {

printf("%s\n", mark\_names\_marks[i]);

break;

}

if (i == 4) {

printf("Така оцінка відсутня! // Та за шо(((\n");

break;

}

}

} while (atoi(mark\_string) != 0);

}

void task3() {

char mark\_string[10], mark\_names\_marks[4][100] = { {"Не задовільно! // Вчись добре!"}, {"Задовільно! // Бабушко ідьош на пятьорку."}, {"Добре! // Ідьош на 200% IQ"}, {"Відмінно! // А чого не 6?"} };

int mark, mark\_arr[5] = { 2,3,4,5,0 };

printf("Завдання 3: Написати програму, яка буде здійснювати зворотне перетвореннядо зав-дання №2,\nтобто перетворювати введену з клавіатури оцінку у вигляді рядка тексту у числове значення.\n");

do {

printf("Введіть оцінку(від 2 - 5) - (0 - вихід з завдання) ---> "); scanf("%s", mark\_string);

mark = atoi(mark\_string);

if (mark == 0) {

break;

}

for (int i = 0; i < 5; i++) {

if (mark == mark\_arr[i]) {

printf("%s\n", mark\_names\_marks[i]);

break;

}

if (mark\_arr[i] == 0) {

printf("Така оцінка відсутня! // Та за шо(((\n");

break;

}

}

} while (mark);

}

int main() {

SetConsoleCP(1251); SetConsoleOutputCP(1251);

printf("Самостійна робота.\n");

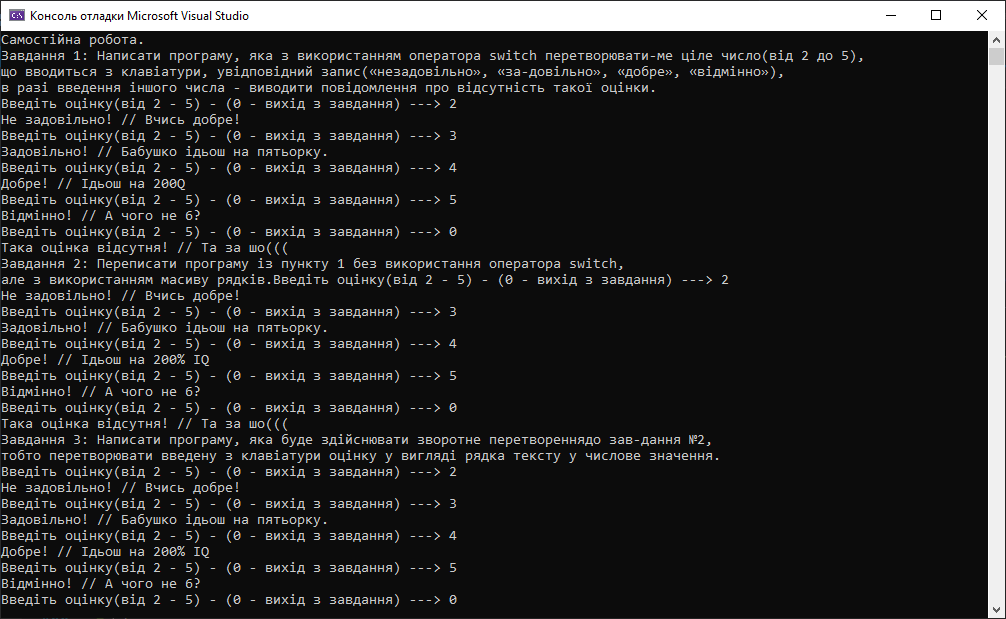
task1();

task2();

task3();

}

Результат програми:



***Висновки:*** в ході виконання лабораторної роботи було ознайомлено з середовищем MS Visual Studio. Досліджено та отримано практичні навики щодо створення програм.