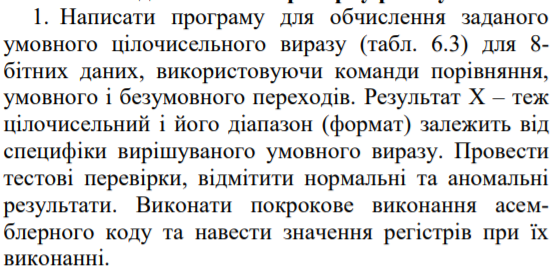
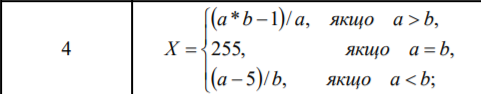
**ЛАБОРАТОРНА РОБОТА № 6**

***Мета:*** ознайомитися з основними ко- мандами мови Assembler дляорганізації умовних переходів; набути практичних навичок в написанні програм з організацією умовних переходів на мові.

**Хід роботи:**

Завдання 1:





#include <Windows.h>

#include <stdio.h>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

signed char a, b, res, dil = 0, per = 0, res\_asm;

printf("a[-128; 127] = "); scanf\_s("%hhi", &a);

printf("b[-128; 127] = "); scanf\_s("%hhi", &b);

if (a > b) {

if (a == 0) {

printf("Error:");

}

else {

res = (a \* b - 1)/a;

}

}

else if (a == b) {

res = 255;

}

else if (a < b) {

if (b == 0) {

printf("Error:");

}

else {

res = (a - 5) / b;

}

}

\_\_asm {

mov al, a;

mov bl, b;

cmp al, bl;

jg mark1; //a > b

je mark2; //a = b

jl mark3; //a < b

mark1: //a > b

cmp al, 0;

je error1; //ділення на 0

imul al;

sub al, 1;

cbw;

mov bl, a;

idiv bl;

jo error2;

mov res\_asm, al;

jmp ext;

mark2: //a = b

mov res\_asm, 255;

jmp ext;

mark3://a < b

cmp bl, 0;

je error1; //ділення на 0

sub al,5;

cbw;

idiv bl;

jo error2;

mov res\_asm, al;

jmp ext;

//errors

error1: //ділення на 0

mov dil, 1;

jmp ext;

//exit

error2:

mov per, 1;

jmp ext;

ext:

}

if (dil > 0) {

printf("Ділення на 0!\n");

}

else if (per > 0) {

printf("Переповнення!\n");

}

else if (dil == 0 && per == 0) {

printf("res = %hhi,\nres\_asm = %hhi\n", res, res\_asm);

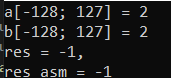
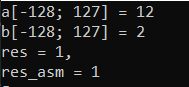
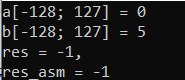
}

system("pause");

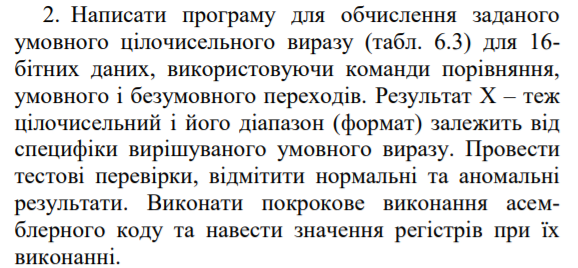
return 0;

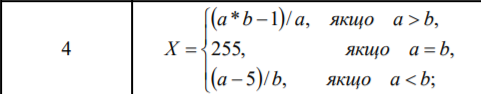
}

**Результат виконання:**



Видає 1,бо переповнений тип пам’яті





#include <Windows.h>

#include <stdio.h>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

short int a, b, res, dil = 0, per = 0, res\_asm;

printf("a[-128; 127] = "); scanf\_s("%hi", &a);

printf("b[-128; 127] = "); scanf\_s("%hi", &b);

if (a > b) {

if (a == 0) {

printf("Error:");

}

else {

res = (a \* b - 1)/a;

}

}

else if (a == b) {

res = 255;

}

else if (a < b) {

if (b == 0) {

printf("Error:");

}

else {

res = (a - 5) / b;

}

}

\_\_asm {

mov ax, a;

mov bx, b;

cmp ax, bx;

jg mark1; //a > b

je mark2; //a = b

jl mark3; //a < b

mark1: //a > b

cmp al, 0;

je error1; //ділення на 0

imul ax;

sub ax, 1;

cwd;

mov bx, a;

idiv ax;

jo error2;

mov res\_asm, ax;

jmp ext;

mark2: //a = b

mov res\_asm, 255;

jmp ext;

mark3://a < b

cmp bx, 0;

je error1; //ділення на 0

sub ax,5;

cwd;

idiv bx;

jo error2;

mov res\_asm, ax;

jmp ext;

//errors

error1: //ділення на 0

mov dil, 1;

jmp ext;

//exit

error2:

mov per, 1;

jmp ext;

ext:

}

if (dil > 0) {

printf("Ділення на 0!\n");

}

else if (per > 0) {

printf("Переповнення!\n");

}

else if (dil == 0 && per == 0) {

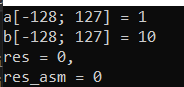
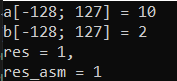
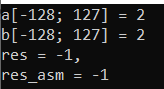
printf("res = %hhi,\nres\_asm = %hhi\n", res, res\_asm);

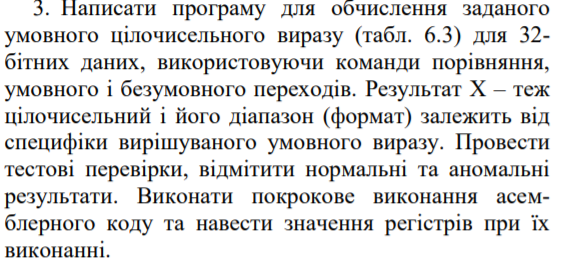
}

system("pause");

return 0;

}





Лістинг програми:

#include <Windows.h>

#include <stdio.h>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

short int a, b, res, dil = 0, per = 0, res\_asm;

printf("a[-128; 127] = "); scanf\_s("%hd", &a);

printf("b[-128; 127] = "); scanf\_s("%hd", &b);

if (a > b) {

if (a == 0) {

printf("Error:");

}

else {

res = (a \* b - 1)/a;

}

}

else if (a == b) {

res = 255;

}

else if (a < b) {

if (b == 0) {

printf("Error:");

}

else {

res = (a - 5) / b;

}

}

\_\_asm {

mov eax, a;

mov ebx, b;

cmp eax, ebx;

jg mark1; //a > b

je mark2; //a = b

jl mark3; //a < b

mark1: //a > b

cmp al, 0;

je error1; //ділення на 0

imul eax;

sub eax, 1;

cdq;

mov ebx, a;

idiv eax;

jo error2;

mov res\_asm, eax;

jmp ext;

mark2: //a = b

mov res\_asm, 255;

jmp ext;

mark3://a < b

cmp ebx, 0;

je error1; //ділення на 0

sub eax,5;

cdq;

idiv ebx;

jo error2;

mov res\_asm, eax;

jmp ext;

//errors

error1: //ділення на 0

mov dil, 1;

jmp ext;

//exit

error2:

mov per, 1;

jmp ext;

ext:

}

if (dil > 0) {

printf("Ділення на 0!\n");

}

else if (per > 0) {

printf("Переповнення!\n");

}

else if (dil == 0 && per == 0) {

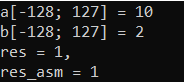
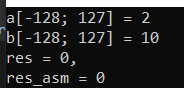
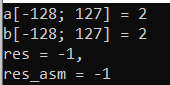
printf("res = %hhi,\nres\_asm = %hhi\n", res, res\_asm);

}

system("pause");

return 0;

}

******

***Висновки:*** Ми ознайомилися з основними ко- мандами мови Assembler для організації умовних переходів; набули практичних навичок в написанні програм з організацією умовних переходів на мові