

ЛАБОРАТОРНА РОБОТА №7

ОРГАНІЗАЦІЯ ЦИКЛІВ І РОБОТА З ЦІЛОЧИСЛЕНИМИ МАСИВАМИ

Мета: ознайомитися з основними командами мови Assembler для організації циклів і роботи з цілочисельними масивами; набути практичних навичок в написанні програм з використанням циклів та масивів на мові Assembler

Хід роботи:

Завдання 1. Написати програму для обробки одномірного масиву для початкових даних(табл.7.5) в знаковому форматі. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні. Відмітити нормальні та аномальні результати, зробити аналіз результатів.

Знайти, скільки елементів масиву $A=\{a[i]\}$ задовольняють умові: $c \leq a[i] \leq d$. Тип даних **BYTE**.

Лістинг програми:

```
#include "pch.h"
#include "stdio.h"
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <conio.h>
#include <time.h>
#define N 10
int main()
{
    setlocale(LC_ALL, "rus");
    short int a[N];
    short int c = 0, d = 0;
    printf("c<=a[i]<=d\n\n");
    do
    {
        printf("Введите значения из диапазона [-32768...32767]:\n");
        printf("c = "); scanf_s("%d", &c);
        printf("d = "); scanf_s("%d", &d);
        printf("\n");
        if (c >= d)
            printf("C не может быть больше или равно D! Попробуйте еще раз.\n");
    } while (c >= d);
    int n = N;
    short int res = 0;
    for (int i = 0; i < N; i++)
    {
        a[i] = rand() % 10;
        printf("A[%d] = %d\n", i, a[i]);
    }
}
```

					ЖДТУ.18.121.01.000 – Лр7					
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.	Абанін К.А				Звіт з лабораторної роботи №7			Літ.	Арк.	Аркушів
Перевір.	Байлюк Є.М								1	6
Керівник								ФІКТ Гр. ПІ-60 [1]		
Н. контр.										
Зав. каф.										

```

asm
{
    mov ax, c; // <ax> = c
    mov bx, d; // <bx> = d
    mov ecx, n; // <ecx> = n
    mov dx, 0; // <dx> = 0
    dec ecx; // зменшуємо значення в регістрі <ecx> на 1 (n=9)
    //---------------------
cycle: // цикл cycle
    shl ecx, 1; // зсув вліво на 1 розряд
    mov si, a[ecx]; // <si> = a[ecx]
    cmp si, 0; // порівнюємо значення регістра <si> з 0
    jnl exit1; // а[ecx]<0 якщо менше - перейти до циклу exit1
    cmp si, ax; // порівнюємо значення регістра <si> з значенням в регістрі
    <ax>
    jnl exit1; // а[ecx]<c якщо менше - перейти до циклу exit1
    cmp si, bx; // порівнюємо значення регістра <si> з значенням в
    регістрі <bx>
    jg exit1; // а[ecx]>d якщо більше - перейти до циклу exit1
    inc dx; // dx=dx+1 додаємо до регістра <dx> значення регістра <si>
    //---------------------
exit1: // цикл exit1
    shr ecx, 1; // зсув вправо на 1 розряд
    dec cx; // зменшуємо значення в регістрі <ecx> на 1
    cmp cx, 0; // порівнюємо значення регістра <ecx> з 0
    jnl cycle; // поки не менше - перейти до циклу cycle
    mov res, dx; // res = <dx>
}
if (res > 32767 || res < -32768)
    printf("Переполнение!\n");
else
    printf("Количество элементов = %d\n", res);
_getch(); return 0;
}

```

Результат виконання програми:

```

C:\Users\aweso\source\repos\ConsoleApplication77\Debug\ConsoleAppli
c<=a[i]<=d
Введите значения из диапазона [-32768...32767]:
c = 1251
d = 1
С не может быть больше или равно D! Попробуйте еще раз.
Введите значения из диапазона [-32768...32767]:
c = 12
d = 12
С не может быть больше или равно D! Попробуйте еще раз.
Введите значения из диапазона [-32768...32767]:
c = 7
d = 16
A[0] = 1
A[1] = 7
A[2] = 4
A[3] = 0
A[4] = 9
A[5] = 4
A[6] = 8
A[7] = 8
A[8] = 2
A[9] = 4
Количество элементов = 4

```

Рис. 1. Результат виконання програми

Значення регістрів при покроковому виконанні:

Команда	Значення регістра				EFLAGS/FLAGS(CF,OF)
	ax	bx	ecx	dx	
mov bx, c	н/в	7	н/в	н/в	---
mov dx, d	н/в	н/в	н/в	16	---
mov ecx, n	н/в	н/в	10	н/в	---
lea si, a	Завантаження зміщення масиву в регістр SI				
cycle:	Мітка циклу cycle(код буде виконуватись в цій мітці, якщо умова – істина)				
loadsw	Завантажуємо слово з пам'яті, на який вказує регістр SI				
cmp ax, bx	Порівнюємо значення ax та bx				
jl next	Перехід до циклу next ,якщо значення менше				
cmp ax, bx	Порівнюємо значення ax та bx				
jg next	Перехід до циклу next ,якщо значення більше				
inc suma	Збільшення результату на 1				
next:	Мітка циклу next				
dec ecx	н/в	н/в	9	н/в	---
cmp ecx, 0	Порівнюємо значення ecx з нулем				
jnl cycle	Якщо не менше - перехід до циклу cycle				
mov res, dx	н/в	н/в	н/в	4	---

Завдання 2. Написати програму для обробки двовимірного масиву для початкових даних(табл.7.6) в знаковому форматі. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні. Відмітити нормальні та аномальні результати, зробити аналіз результатів.

Знайти, скільки елементів масиву $A=\{a[i][j]\}$ задовольняють умові: $c \leq a[i][j] \leq d$. Тип даних BYTE.

Лістинг програми:

```
#include "pch.h"
#include "stdio.h"
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <conio.h>
#include <time.h>
#define N 5
int main()
{
    setlocale(LC_ALL, "rus");
    short int a[N][N];
    short int c = 0, d = 0;
    do {
        printf("c<=a[i]<=d\n\n");
        printf("Введите значения из диапазона: [-32768...32767]\n");
        printf("c = ");
        scanf_s("%d", &c);
        printf("d = ");
        scanf_s("%d", &d);
        printf("\n");
        if (c >= d)
            printf("C не может быть больше или равно D! Попробуйте еще раз.\n");
    } while (c >= d);
    int n = N * N; // загальна кількість елементів масиву
    short int suma = 0;
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            a[i][j] = rand() % 10;
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    __asm
    {
        mov bx, c; // <bx> = c
        mov dx, d; // <dx> = d
        mov ecx, n; // <ecx> = n
        lea si, a; // завантаження зміщення масиву в регістр SI(замінює shr ecx, 1 та shl ecx, 1)
        //-----
cycle: // цикл cycle
        lodsw; // завантажуюємо слово з пам'яті, на який вказує регістр SI
        cmp ax, bx; // порівнюємо з нижньою границею - c
        jl next; // якщо менше - перейти до циклу next
        cmp ax, dx; // порівнюємо з верхньою границею - d
        jg next; // якщо більше - перейти до циклу next
        inc suma; // збільшити результат на 1
        //-----
next: // цикл next
        dec ecx; // зменшуємо к-ть чисел на 1
        cmp ecx, 0; // порівнюємо к-ть чисел з 0
        jnl cycle; // якщо не менше - перейти до циклу cycle
    }
    if (suma > 32767 || suma < -32768)
        printf("Переполнение!\n");
    else
        printf("\n"); printf("Количество элементов = %d\n", suma);
    _getch(); return 0;
}
```

		Абанін К.А			ЖДТУ.18.121.01.000 – Лр7	Арк.
		Байлюк Є.М				
Змн.	Арк.	№ докум.	Підпис	Дата		4

Результат виконання програми:

```
C:\Users\aweso\source\repos\ConsoleApplication77\Debug\ConsoleA
c<=a[i]<=d
Введите значения из диапазона:[-32768...32767]
c = 1212312
d = 11
С не может быть больше или равно D! Попробуйте еще раз.
c<=a[i]<=d
Введите значения из диапазона:[-32768...32767]
c = 12
d = 17
1 7 4 0 9
4 8 8 2 4
5 5 1 7 1
1 5 2 7 6
1 4 2 3 2
Количество элементов = 0
```

Рис. 2. Результат виконання програми

Значення регістрів при покроковому виконанні:

Команда	Значення регістра				EFLAGS/FLAGS(CF,OF)
	ax	bx	ecx	dx	
mov bx, c	н/в	12	н/в	н/в	---
mov dx, d	н/в	н/в	н/в	17	---
mov ecx, n	н/в	н/в	10	н/в	---
lea si, a	Завантаження зміщення масиву в регістр SI				
cycle:	Мітка циклу cycle(код буде виконуватись в цій мітці, якщо умова – істина)				
loadsw	Завантажуємо слово з пам'яті, на який вказує регістр SI				
cmp ax, bx	Порівнюємо значення ax та bx				
jl next	Перехід до циклу next ,якщо значення менше				
cmp ax, bx	Порівнюємо значення ax та bx				
jg next	Перехід до циклу next ,якщо значення більше				
inc suma	Збільшення результату на 1				

next:	Мітка циклу next				
dec esx	н/в	н/в	9	н/в	---
cmp esx, 0	Порівнюємо значення esx з нулем				
jnl cycle	Якщо не менше - перехід до циклу cycle				

Висновок: Ми ознайомилися з основними командами мови Assembler для організації циклів і роботи з цілочисельними масивами; набули практичних навичок в написанні програм з використанням циклів та масивів на мові Assembler

		Абанін К.А			ЖДТУ.18.121.01.000 – Лр7	Арк.
		Байлюк Є.М				
Змн.	Арк.	№ докум.	Підпис	Дата		6