МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського

«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп’ютерних систем та інфокомунікацій

Кафедра комп’ютерних систем, мереж і кібербезпеки (503)

Лабораторна робота № *7*

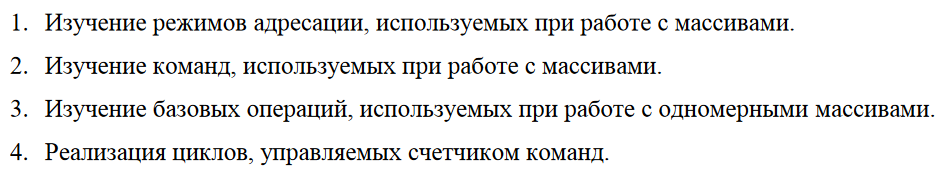
|  |  |
| --- | --- |
|  | *Вивчення одномірних масивів в МП х86* |
|  | (назва лабораторної роботи) |
| з дисципліни | *Архітектура комп'ютерів* |
|  | (шифр)  ХАІ**.**503**.**525a**.**03О**.**123-Комп'ютерна інженерія**,** ПЗ №9629619 |

|  |  |  |
| --- | --- | --- |
| Виконав студент гр. | 525а | *Литвиненко А.В.* |
| 17.12.2022 | (№ групи) | (П.І.Б.) |
| (підпис, дата) |  |  |
| Перевірив | канд. техн. наук, доцент | |
|  |  | *В. І. Дужий* |
| (підпис, дата) |  | (П.І.Б.) |

Харків – 2022

**Тема роботи:** изучение одномерных массивов в МП х86

**Мета роботи**:

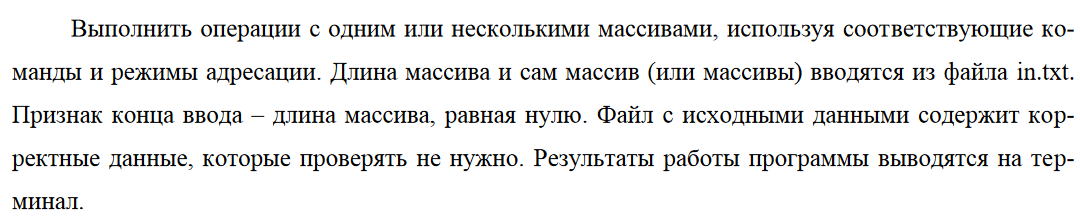


**Варіант 5**

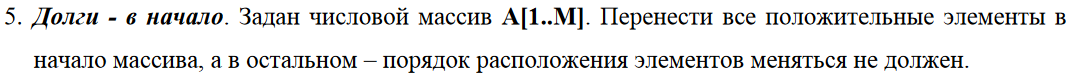
**Задача 1**

**Частина 1**. Постановка завдання

**Умова:**



**Умова з додатка:**



**Вхідні дані:**

in.txt – текстовий файл з вхідними даними;

MAX\_LENGTH – максимальна довжина масиву – константа

arr\_c, arr\_asm – вхіжний масив довгих цілих чисел

**Необхідний результат:**

arr\_c, arr\_asm – результуючий масив довгих цілих чисел

**Частина 2**. Схема алгоритму

**Опис на псевдокоді**

Відкрити файл з ім’ям in.txt;

J = 0;

// 10 – кількість тестів у файлі

Поки J < 10:

Прочитати довжину масиву – N;

Якщо N > максимальної довжини:

Пропустити тест;

Прочитати N елементів в масив arr\_c;

Зробити копію для мови асемблера – arr\_asm;

Вивести масиви;

Перемістити усі додатні числа на початок в arr\_c;

Перемістити усі додатні числа на початок в arr\_asm;

Вивести масиви;

J++

Закрити файл з ім’ям in.txt;

На основі постановки завдання розроблений алгоритм, представлений на рисунку 1.

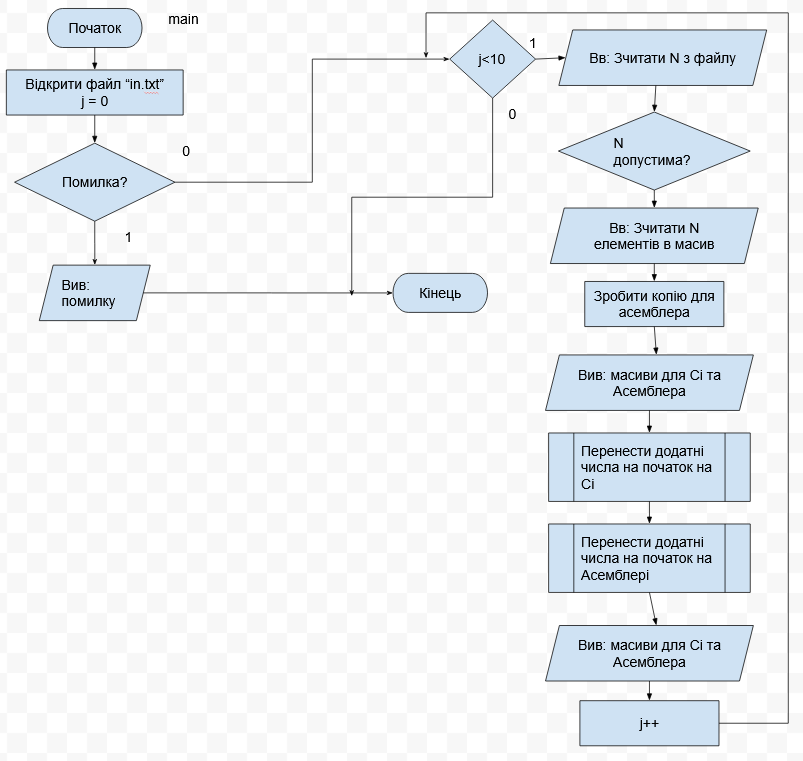


Рисунок 1 – Алгоритм основної програми

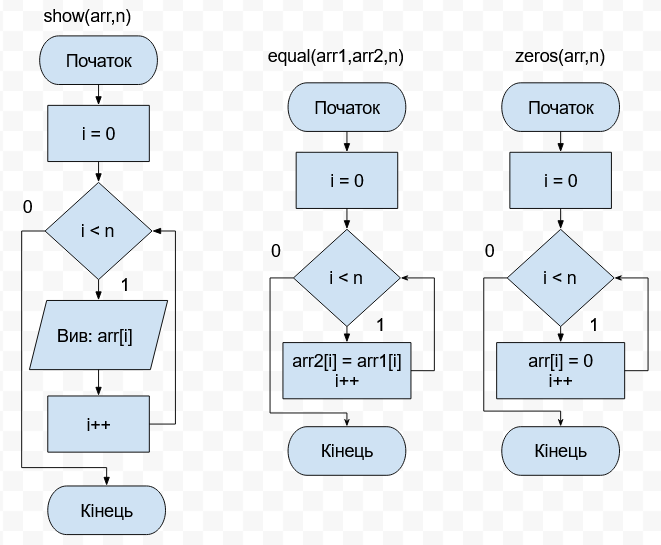


Рисунок 2 - Алгоритми додаткових функцій

**Частина 3**. Розробка тестів

Таблиця 1 – Тестові набори

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Довжина | Вхідний масив | Очікуваний масив | Коментар |
| 1 | 0 |  |  | Пустий масив |
| 2 | 1025 |  |  | Довжина більше максимальної |
| 3 | 10 | 123 -30 -60 70 0 -1000 40000 80 90 0 | 40000 123 90 80 70 0 0 -30 -60 -1000 | Різні числа (+-) |
| 4 | 10 | -1000 -60 -30 0 0 70 80 90 123 40000 | 70 80 90 123 40000 -1000 -60 -30 0 0 | Числа за зростанням |
| 5 | 10 | 123 -30 -60 70 0 -1000 40000 80 90 0 | 123 70 40000 80 90 -1000 -60 -30 0 0 | Числа за спаданням |
| 6 | 10 | 5 6 -2 -6 4 6 4 -3 1 2 | 5 6 4 6 4 1 2 -3 -6 -2 | Різні числа (+-)і |
| 7 | 1 | 5555 | 5555 | Один елемент |
| 8 | 10 | 1 2 3 4 5 6 7 8 9 10 | 1 2 3 4 5 6 7 8 9 10 | Лише додатні числа |
| 9 | 10 | -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 | -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 | Лише від’ємні числа |
| 10 | 100 | 54 -23 47 -67 10 -62 -1 22 -45 5 96 -18 -2 8 -37 -95 54 -18 19 55 28 28 24 66 52 -88 -39 -54 69 91 22 -32 -39 -45 -98 47 9 96 61 33 31 -91 18 -82 -99 -100 13 -22 -15 100 73 -58 -69 -23 30 23 -14 -52 68 89 7 -11 -30 -38 -7 -3 -22 23 -41 -48 91 -96 63 -4 -27 -76 8 33 65 22 63 76 35 91 96 13 38 44 36 -22 -33 25 73 50 6 27 9 -12 67 59 | 54 47 10 22 5 96 8 54 19 55 28 28 24 66 52 69 91 22 47 9 96 61 33 31 18 13 100 73 30 23 68 89 7 23 91 63 8 33 65 22 63 76 35 91 96 13 38 44 36 25 73 50 6 27 9 67 59 -52 -18 -32 -39 -11 -30 -38 -7 -3 -22 -45 -41 -48 -98 -96 -45 -4 -27 -76 -23 -62 -18 -2 -1 -91 -37 -82 -99 -100 -88 -22 -15 -22 -33 -39 -54 -58 -69 -23 -95 -12 -67 -14 | Сто елементів |

**Частина 4**. Текст програми

Відповідно до розробленого алгоритму в середовищі Microsoft Visual Studio була написана програма, яка наведена нижче.

**laba\_7.cpp**

/\*

\* File: laba\_7.cpp

\* Долги - в начало. Задан числовой массив А[1..M]. Перенести все положительные элементы в

\* начало массива, а в остальном – порядок расположения элементов меняться не должен.

\*

\* (C) Lytvynenko A.V., 2022

\*/

#include <iostream>

#include <iomanip>

using namespace std;

#define MAX\_LENGTH 1024

long arr\_c[1024] = { 0, };

long arr\_asm[1024] = { 0, };

void show(long arr[], long n) {

/// <summary>

/// Виводить масив елементів через пробіл

/// </summary>

/// <param name="arr">Масив елементів</param>

/// <param name="n">Довжина масиву</param>

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

printf("%ld ", arr[i]);

}

printf("\n");

}

void equal(long arr1[], long arr2[], long n) {

/// <summary>

/// Прирівнює перший масив до другого. Масив2 = Масив1

/// </summary>

/// <param name="arr1">Масив 1</param>

/// <param name="arr2">Масив 2</param>

/// <param name="n">Довжина масиву</param>

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

arr2[i] = arr1[i];

}

}

void zeros(long arr[], long n) {

/// <summary>

/// Обнуляє повністю масив

/// </summary>

/// <param name="arr">Масив</param>

/// <param name="n">Кількість елементів</param>

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

arr[i] = 0;

}

}

int main() {

// Довжина масиву

long n;

// Індекс для встановлення наступного позитивного числа (можливий індекс)

long allowed\_index = 0;

// Тимчасово змінна

long tmp;

// Ітераторний індекс

long i;

// Файловий вказівник

FILE\* file;

// Якщо не вдається відкрити файл - завершення програми

if ((file = fopen("in.txt", "r")) == NULL) {

printf("[ERR] Can't open file!\n");

return 1;

}

int j = 0;

while (j < 10) {

fscanf(file, "%ld\n", &n);

if (n <= 0 || n > MAX\_LENGTH) {

printf("[ERR] Invalid paramaters!\n\n");

j++;

continue;

}

zeros(arr\_c, n);

zeros(arr\_asm, n);

allowed\_index = 0;

for (i = 0;i < n;i++) {

fscanf(file, "%ld", &arr\_c[i]);

}

equal(arr\_c, arr\_asm, n);

// Виведення масивів до операцій

printf("[BEFORE] [C] :\t\t");

show(arr\_c, n);

printf("[BEFORE] [ASM] :\t");

show(arr\_asm, n);

// Частина на Сі

for (i = 0;i < n;i++) {

// Якщо число позитивне, то обмінятися з числом на можливому індексі

if (arr\_c[i] > 0) {

tmp = arr\_c[allowed\_index];

arr\_c[allowed\_index] = arr\_c[i];

arr\_c[i] = tmp;

allowed\_index++;

}

}

/\*

eax - allowed\_index

ecx - arr\_asm[eax]

edx - arr\_asm[esi]

ebx - tmp

esp

ebp

esi - I

edi

\*/

// Частина на асемблері

\_\_asm {

// for(i = 0;i < n;i++)

mov esi, 0 // i

mov eax, 0 // allowed\_index

For1:

cmp esi, n

jge EndFor1

cmp arr\_asm[esi \* 4], 0

jg Exchange

jmp Go

Exchange :

mov ebx, arr\_asm[eax \* 4]

mov edx, arr\_asm[esi \* 4]

mov arr\_asm[eax \* 4], edx

mov arr\_asm[esi \* 4], ebx

inc eax

jmp Go

Go :

inc esi

jmp For1

EndFor1 :

}

// Виведення масивів після виконання операцій

printf("[AFTER] [C] :\t\t");

show(arr\_c, n);

printf("[AFTER] [ASM] :\t\t");

show(arr\_asm, n);

printf("\n");

j++;

}

return 0;

}

**in.txt**

0

1025

10

40000 123 90 80 70 0 0 -30 -60 -1000

10

-1000 -60 -30 0 0 70 80 90 123 40000

10

123 -30 -60 70 0 -1000 40000 80 90 0

10

5 6 -2 -6 4 6 4 -3 1 2

1

5555

10

1 2 3 4 5 6 7 8 9 10

10

-1 -2 -3 -4 -5 -6 -7 -8 -9 -10

100

54 -23 47 -67 10 -62 -1 22 -45 5 96 -18 -2 8 -37 -95 54 -18 19 55 28 28 24 66 52 -88 -39 -54 69 91 22 -32 -39 -45 -98 47 9 96 61 33 31 -91 18 -82 -99 -100 13 -22 -15 100 73 -58 -69 -23 30 23 -14 -52 68 89 7 -11 -30 -38 -7 -3 -22 23 -41 -48 91 -96 63 -4 -27 -76 8 33 65 22 63 76 35 91 96 13 38 44 36 -22 -33 25 73 50 6 27 9 -12 67 59

**Частина 5**. Тестування

Результати тестування наведені в таблиці 2.

Таблиця 2 – Результати тестування на Сі

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Довжина | Вхідні дані | Очікуваний результат | Отриманий результат | Результат тестування |
| 1 | 0 |  |  |  | OK |
| 2 | 1025 |  |  |  | OK |
| 3 | 10 | 123 -30 -60 70 0 -1000 40000 80 90 0 | 40000 123 90 80 70 0 0 -30 -60 -1000 | 40000 123 90 80 70 0 0 -30 -60 -1000 | OK |
| 4 | 10 | -1000 -60 -30 0 0 70 80 90 123 40000 | 70 80 90 123 40000 -1000 -60 -30 0 0 | 70 80 90 123 40000 -1000 -60 -30 0 0 | OK |
| 5 | 10 | 123 -30 -60 70 0 -1000 40000 80 90 0 | 123 70 40000 80 90 -1000 -60 -30 0 0 | 123 70 40000 80 90 -1000 -60 -30 0 0 | OK |
| 6 | 10 | 5 6 -2 -6 4 6 4 -3 1 2 | 5 6 4 6 4 1 2 -3 -6 -2 | 5 6 4 6 4 1 2 -3 -6 -2 | OK |
| 7 | 1 | 5555 | 5555 | 5555 | OK |
| 8 | 10 | 1 2 3 4 5 6 7 8 9 10 | 1 2 3 4 5 6 7 8 9 10 | 1 2 3 4 5 6 7 8 9 10 | OK |
| 9 | 10 | -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 | -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 | -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 | OK |
| 10 | 100 | 54 -23 47 -67 10 -62 -1 22 -45 5 96 -18 -2 8 -37 -95 54 -18 19 55 28 28 24 66 52 -88 -39 -54 69 91 22 -32 -39 -45 -98 47 9 96 61 33 31 -91 18 -82 -99 -100 13 -22 -15 100 73 -58 -69 -23 30 23 -14 -52 68 89 7 -11 -30 -38 -7 -3 -22 23 -41 -48 91 -96 63 -4 -27 -76 8 33 65 22 63 76 35 91 96 13 38 44 36 -22 -33 25 73 50 6 27 9 -12 67 59 | 54 47 10 22 5 96 8 54 19 55 28 28 24 66 52 69 91 22 47 9 96 61 33 31 18 13 100 73 30 23 68 89 7 23 91 63 8 33 65 22 63 76 35 91 96 13 38 44 36 25 73 50 6 27 9 67 59 -52 -18 -32 -39 -11 -30 -38 -7 -3 -22 -45 -41 -48 -98 -96 -45 -4 -27 -76 -23 -62 -18 -2 -1 -91 -37 -82 -99 -100 -88 -22 -15 -22 -33 -39 -54 -58 -69 -23 -95 -12 -67 -14 | 54 47 10 22 5 96 8 54 19 55 28 28 24 66 52 69 91 22 47 9 96 61 33 31 18 13 100 73 30 23 68 89 7 23 91 63 8 33 65 22 63 76 35 91 96 13 38 44 36 25 73 50 6 27 9 67 59 -52 -18 -32 -39 -11 -30 -38 -7 -3 -22 -45 -41 -48 -98 -96 -45 -4 -27 -76 -23 -62 -18 -2 -1 -91 -37 -82 -99 -100 -88 -22 -15 -22 -33 -39 -54 -58 -69 -23 -95 -12 -67 -14 | OK |

**Скриншот тестування:**

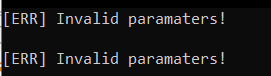


Рисунок 3 – скришот тестування тест №1-2

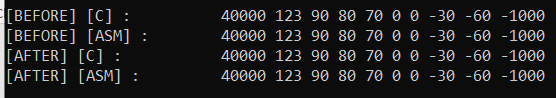


Рисунок 4 – тест №3

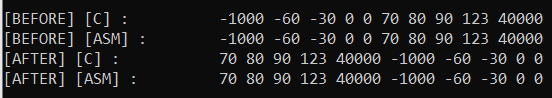


Рисунок 5 - тест №4

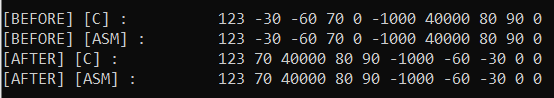


Рисунок 6 - тест №5

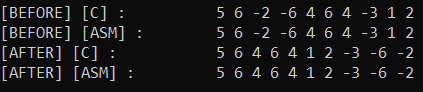


Рисунок 7 - тест №6

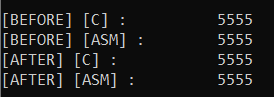


Рисунок 8 - тест №7

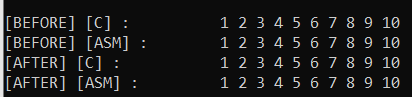


Рисунок 9 - тест №8

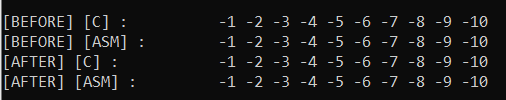


Рисунок 10 - тест №9

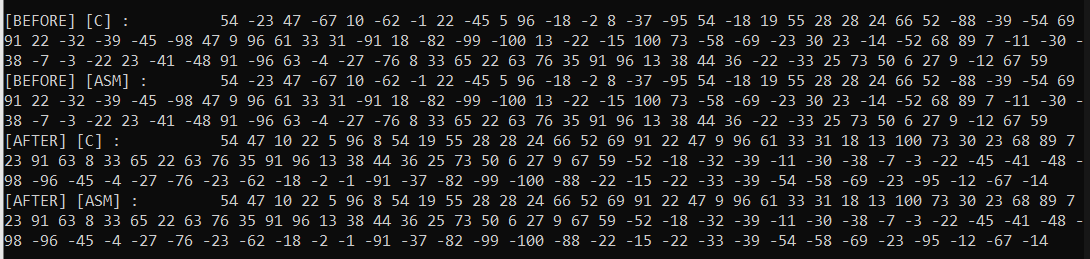


Рисунок 11 - тест №10

**Висновки**

Під час цієї лабораторної роботи я вивчив адресацію та її використання під час програмування на мові асемблера з масивами. У моєму завданні було необхідно було пробігтися по масиву і перемістити усі додатні числа в початок масиву, що не створило дуже великих проблем: спочатку я проаналізував задачу і умову до неї, після чого я написав псевдокод і графічно зобразив їх за допомогою блок-схем – це дуже спрощує проектування.

Приступивши до технічної реалізації проєкту, було важливо написати дуже простий код на мові Сі, щоб як наслідок, було просто реалізувати на мові асемблері і допустити якомога менше помилок.

Після закінчення основної частини проєкту і базового тестування на функціональність я приступив до включення «автоматизації» у процес тестування – додав можливість працювати з файлами, звідки зчитувалася уся необхідна інформація, а саме N – кількість елементів та Arr[N] – масив елементів довжини N.

На мою думку, я повністю засвоїв цю лабораторну роботу і зможу реалізувати подібний функціонал у майбутньому без додаткових труднощів.