МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

Кафедра СТ

Отчет

по лабораторной работе № 3

по дисциплине: «Программирование(системное)»

Выполнил: Проверил:

ст. гр. АКТСИу-17-2 Коваленко А. И.

Черкашин В.А.

Харьков 2018

1 Изучение языка Ассемблер

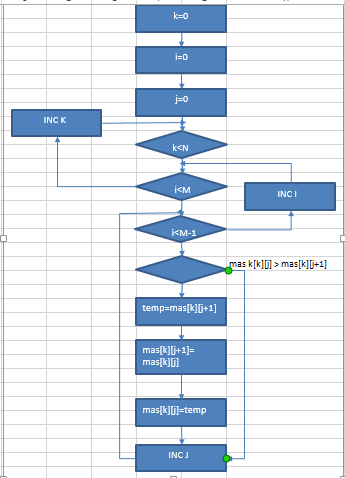
* 1. Цель работы

Изучение основных команд для работы с условными и безусловными переходами. Организация циклов.

* 1. Задание

В двухмерном массиве отсортировать элементы каждого столбцы по возрастанию. Найти значение минимального и максимального элемента в массиве.

1.3 Алгоритм сортировки.



* 1. Листинг программы

;Черкашин АКТСИу 17-2 Лаба 3

.486

.model flat, STDCALL

option casemap:none

include \masm32\include\windows.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

include \masm32\include\msvcrt.inc

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\msvcrt.lib

;=========================================

.data

N dw ?

M dw ?

Ns1 dw ?

M2 dw ?

Matrix db 128 dup (?)

MAX dw ?

MIN dw ?

iterator dw ?

iterator2 dw ?

iterator3 dw ?

outHandle dd ?

inHandle dd ?

namberW dd ?

namberR dd ?

ConsoleTitle db "Лаба 3",0

StartText db "Лабараторная работа №3 Черкашин В.А. АКТСИу 17-2 Вариант 11",0

Task db "В двухмерном массиве отсортировать элементы каждого стобца по возрастанию",0dh,0ah,

"найти значение макисмально и минимального элемента",0

MatrixText db "Введенная матрица",0

SortText db "Отсортированная матрица",0

NewLine db 0dh,0ah

TextN db "Введите N",0

TextM db "Введите M", 0

NumberBuf db 5 dup (?)

TextBuf db 60 dup (?)

TitleMB db "Лаб 3",0

formatMatrix db "%d ",0

MaxFormat db "Максимальное значение элемента в матрице = %d",0

MinFormat db "Минимальное значение элемента в матрице = %d",0

.code

start:

;======Получение консоли, установка Titile, получение Handle====

invoke AllocConsole

invoke GetStdHandle, STD\_OUTPUT\_HANDLE

MOV outHandle, EAX

invoke GetStdHandle, STD\_INPUT\_HANDLE

MOV inHandle, EAX

invoke CharToOem, ADDR ConsoleTitle, ADDR ConsoleTitle

invoke SetConsoleTitle, ADDR ConsoleTitle

;=========Перекодирование==================================

invoke CharToOem, ADDR StartText, ADDR StartText

invoke CharToOem, ADDR Task, ADDR Task

invoke CharToOem, ADDR TextN, ADDR TextN

invoke CharToOem, ADDR TextM, ADDR TextM

;=========Запись и чтение с консоли =========================

invoke WriteConsoleA, outHandle, ADDR StartText, SIZEOF StartText, ADDR namberW, NULL

invoke WriteConsoleA, outHandle, ADDR NewLine, SIZEOF NewLine, ADDR namberW, NULL

invoke WriteConsoleA, outHandle, ADDR Task, SIZEOF Task, ADDR namberW, NULL

invoke WriteConsoleA, outHandle, ADDR NewLine, SIZEOF NewLine, ADDR namberW, NULL

invoke WriteConsoleA, outHandle, ADDR TextN, SIZEOF TextN, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR NumberBuf, SIZEOF NumberBuf, ADDR namberR, NULL

invoke crt\_atoi, addr NumberBuf

MOV N, AX

invoke WriteConsoleA, outHandle, ADDR TextM, SIZEOF TextM, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR NumberBuf, SIZEOF NumberBuf, ADDR namberR, NULL

invoke crt\_atoi, addr NumberBuf

MOV M, AX

;=====Ввод Чисел с клавиатуры ==========

MOV AX, N

MUL M

MOV iterator , AX

mov esi,offset Matrix

@Input:

MOV AX, iterator

CMP AX, 0

JE @InputEnd

invoke ReadConsole, inHandle, ADDR NumberBuf, SIZEOF NumberBuf, ADDR namberR, NULL

invoke crt\_atoi, addr NumberBuf

MOV [ESI], AX

ADD ESI, 2

MOV AX, iterator

SUB AX, 1

MOV iterator, AX

JMP @Input

@InputEnd:

;========Вывод массива на экран=====================

invoke CharToOem, ADDR MatrixText, ADDR MatrixText

invoke WriteConsoleA, outHandle, ADDR MatrixText, SIZEOF MatrixText, ADDR namberW, NULL

invoke WriteConsoleA, outHandle, ADDR NewLine, SIZEOF NewLine, ADDR namberW, NULL

LEA ESI, Matrix

MOV AX, N

MUL M

MOV iterator , AX

Output:

MOV AX, iterator

CMP AX, 0

JE OutputEnd

MOV BX, word ptr [ESI]

invoke wsprintf, addr NumberBuf,addr formatMatrix, BX

invoke CharToOem, ADDR NumberBuf, ADDR NumberBuf

invoke WriteConsoleA, outHandle, ADDR NumberBuf, SIZEOF NumberBuf, ADDR namberW, NULL

ADD ESI, 2

MOV AX, iterator

SUB AX, 1

MOV iterator, AX

MOV AX, iterator

div byte ptr N

CMP AH, 0

JE newline

JMP Output

newline:

invoke WriteConsoleA, outHandle, ADDR NewLine, SIZEOF NewLine, ADDR namberW, NULL

JMP Output

OutputEnd:

;=============Поиск максимального значения и его вывод==============================

LEA ESI, Matrix

MOV AX, N

MUL M

MOV iterator , AX

MOV AX, [ESI]

MOV MAX,AX

ADD ESI,2

Mmax:

MOV AX, iterator

CMP AX, 1

JE MaxEND

MOV AX, [ESI]

CMP AX, MAX

JA MaxProm

MOV AX, iterator

SUB AX, 1

MOV iterator, AX

ADD ESI,2

JMP Mmax

MaxProm:

MOV MAX, AX

MOV AX, iterator

SUB AX, 1

MOV iterator, AX

ADD ESI,2

JMP Mmax

MaxEND:

invoke wsprintf, addr TextBuf,addr MaxFormat, MAX

invoke CharToOem, ADDR TextBuf, ADDR TextBuf

invoke WriteConsoleA, outHandle, ADDR TextBuf, SIZEOF TextBuf, ADDR namberW, NULL

invoke WriteConsoleA, outHandle, ADDR NewLine, SIZEOF NewLine, ADDR namberW, NULL

;========Поиск минимального значения и его вывод=======================

LEA ESI, Matrix

MOV AX, N

MUL M

MOV iterator , AX

MOV AX, [ESI]

MOV MIN,AX

ADD ESI,2

Mmin:

MOV AX, iterator

CMP AX, 1

JE MinEND

MOV AX, [ESI]

CMP AX, MIN

JL MinProm

MOV AX, iterator

SUB AX, 1

MOV iterator, AX

ADD ESI,2

JMP Mmin

MinProm:

MOV MIN, AX

MOV AX, iterator

SUB AX, 1

MOV iterator, AX

ADD ESI,2

JMP Mmin

MinEND:

invoke wsprintf, addr TextBuf,addr MinFormat, MIN

invoke CharToOem, ADDR TextBuf, ADDR TextBuf

invoke WriteConsoleA, outHandle, ADDR TextBuf, SIZEOF TextBuf, ADDR namberW, NULL

invoke WriteConsoleA, outHandle, ADDR NewLine, SIZEOF NewLine, ADDR namberW, NULL

;==========Сортировка элементов в матрице =====================================

MOV EBP, 0

MOV ESI, 0

MOV AX, 0

MOV iterator, AX

MOV AX, N

SUB AX, 1

MOV Ns1, AX

MOV AX, 0

MOV iterator2, AX

MOV iterator3, AX

MOV AX, M

MOV BX, 2

MUL BX

MOV M2, AX

Sort2:

MOV AX, iterator3

CMP AX, M2

JE Sort2end

Sort1:

MOV AX, iterator2

CMP AX, M

JE Sort1end

Sort:

MOV AX, iterator

CMP AX, Ns1

JE SortInnerEnd

MUL N

MOV BX, 2

MUL BX

MOV BP, AX

LEA ECX, Matrix[EBP][ESI]

INC iterator

MOV AX, iterator

MUL N

MOV BX, 2

MUL BX

MOV BP, AX

LEA EBX, Matrix[EBP][ESI]

MOV AX,[EBX]

CMP [ECX], AX

JA SortProm

JMP Sort

SortProm:

MOV AX, [ECX]

MOV DX, [EBX]

MOV [ECX],DX

MOV [EBX],AX

JMP Sort

SortInnerEnd:

INC iterator2

MOV AX, 0

MOV iterator,AX

JMP Sort1

Sort1end:

INC iterator3

INC iterator3

MOV AX, iterator3

MOV ESI, EAX

MOV AX, 0

MOV iterator, AX

MOV iterator2, AX

JMP Sort2

Sort2end:

;========Вывод сортированного массива на экран =============

invoke CharToOem, ADDR SortText, ADDR SortText

invoke WriteConsoleA, outHandle, ADDR SortText, SIZEOF SortText, ADDR namberW, NULL

invoke WriteConsoleA, outHandle, ADDR NewLine, SIZEOF NewLine, ADDR namberW, NULL

LEA ESI, Matrix

MOV AX, N

MUL M

MOV iterator , AX

Output1:

MOV AX, iterator

CMP AX, 0

JE OutputEnd1

MOV BX, word ptr [ESI]

invoke wsprintf, addr NumberBuf,addr formatMatrix, BX

invoke CharToOem, ADDR NumberBuf, ADDR NumberBuf

invoke WriteConsoleA, outHandle, ADDR NumberBuf, SIZEOF NumberBuf, ADDR namberW, NULL

ADD ESI, 2

MOV AX, iterator

SUB AX, 1

MOV iterator, AX

MOV AX, iterator

div byte ptr N

CMP AH, 0

JE newline1

JMP Output1

newline1:

invoke WriteConsoleA, outHandle, ADDR NewLine, SIZEOF NewLine, ADDR namberW, NULL

JMP Output1

OutputEnd1:

;======Что б консоль не закрылась ==========================

invoke ReadConsole, inHandle, ADDR NumberBuf, SIZEOF NumberBuf, ADDR namberR, NULL

invoke ExitProcess, 0

end start

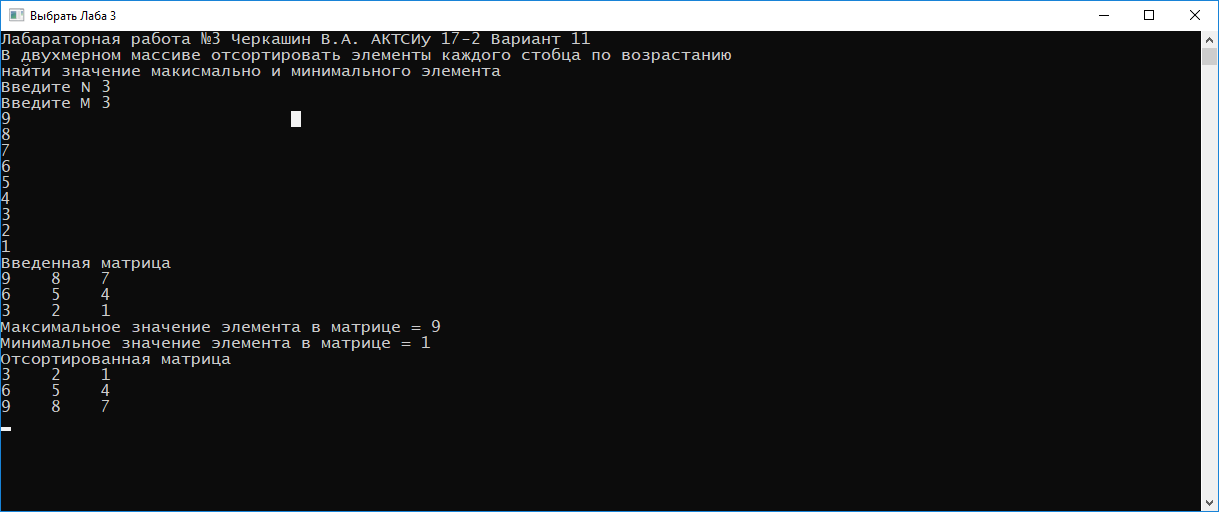


Рис 1 - Результат выполнения программы.

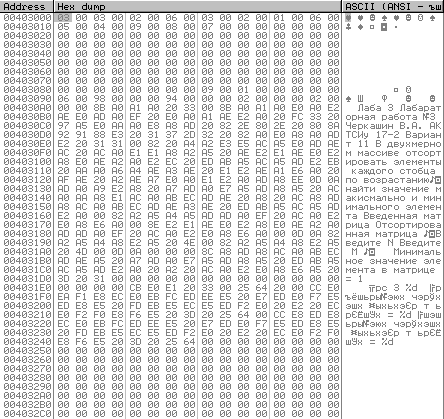


Рис 2 – Дамп памяти

Выводы: В ходе лабораторной работы были изучены команды для организации циклов, так же команды для сравнения и условного перехода, по флагам. Так же были рассмотрены основные флаги на которые влияет команда CMP. Была разработана консольная программа для сортировки массива.