Государственный Университет Молдовы  
Факультет Математики и Информатики  
Департамент Информатики

**Самостоятельная работа №10**

По курсу “Архитектура компьютера и язык ассемблера”

Тема: Макросы. Числовые массивы

Выполнил студент группы I1902:  
Чобану Артём

Кишинэу, 2019

**Самостоятельная работа №9**

**Цель работы:** ознакомление с принципами написания макросов и их использования в программах, а также с технологиями обработки числовых массивов.

**Задание**

**Вариант 3**: Дана числовая матрица размера mx. Создать вектор размера m, содержащий максимальные элементы каждого столбца матрицы.

**Алгоритм:**

1. Для вычисления адреса номер строки умножается на максимальное их количество. К этому добавляется номер столбца. То есть я делю матрицу в памяти по строкам. Для этого я сделал макрос GetAdr.
2. Макросы Get и Set используют GetAdr для получения адреса, а после возвращают/устанавливают значение.
3. При нахождении элемента, который больше текущего максимального, его значение присваивается переменной максимального элемента

**Код программы:**

|  |  |
| --- | --- |
| .model small  NL macro ;макрос перехода  ;на новую строку      mov dx, 0      mov ax, 0      mov ah, 9      lea dx, newLine      int 21h      mov dx, 0      mov ax, 0  endm  GetAdr macro row, col, adr      mov ax, 0      mov bx, 0      mov dx, 0      mov al, row      mul ra;умножаем строку на  ;максимальное их количество  Get macro row, col, val      GetAdr row, col, b;берём адрес      mov ax, 0      mov bx, 0      mov bl, b;Берём значение по адресу      mov bh, 0      mov al, mtx[bx]      mov val, al      mov ax, 0      mov bx, 0  endm  .stack  .data      mtx db 1, 3, 2,9, 8, 3,2, 4, 5,"$"      vec db 3 dup (?), "$"      m db 0      ra db 3      ca db 3      ind db 0      a db 0      b db 0      i db 0      j db 1      h db 0      newLine db 10, 13, '$'  .code  mov ax, @data  mov ds, ax      mov ax, 0      mov dx, 0      mov bx, 0  l1:      Get 0, i, m ;Делаем максимальным  ;первый элемент стобца      mov j, 1;обнуляем индекс строки  l2:      Get j, i, h;получаем текущий  элемент      mov bh, h      cmp bh, m      jle less | mov bh, col      add al, bh;Добавляем количество    ;столбцов        mov adr, al;Возврат адреса      mov ax, 0      mov dx, 0      mov bx, 0  endm  Set macro row, col, val      mov ax, 0      mov bx, 0      GetAdr row, col, b;Берём адрес      mov al, val;устанавливаем  ;значение по адресу      mov bl, b      mov mtx[bx], al      mov ax, 0      mov bx, 0  endm      mov m, bh;Если больше, то  ;делаем максимумом      jmp less  rjump: jmp l1;промежуточный прыжок  less:      inc j      mov bl, ra      cmp j, bl;сравнение счётчика               ;с количеством строк      jl l2 ;Внутренний цикл      mov bl, ind ;Берём позицию  ;в векторе      mov bh, 0      mov al, m      mov vec[bx], al;переписываем в  ;вектор      inc ind;Увеличиваем индекс  ;элемента вектора      mov al, ca      inc i      cmp i, al      jl rjump ;Внешний цикл  mov bx, 0  output:;Вывод вектора      inc bx      mov ah, 2      mov dl, vec[bx-1]      add dx, 30h      int 21h      cmp bl, ind      jl output      NL  mov ah, 4Ch  int 21h  end |

Вывод программы:

