РАСЧЕТ СЛЕДА КВАДРАТНОЙ МАТРИЦЫ

Голубков Никита, БПИ195

НИУ ВШЭ ФКН
ОП «Программная инженерия»

1. Текст задания

Разработать программу вычисления сле́да квадратной матрицы порядка N≤5 при условии размещения элементов матрицы в линейном массиве по строкам.

2. Методы расчета

Программа сохраняет массив как одномерный и шагает по нему с шагом (N+1), складывая соответствующие элементы для расчета следа матрицы. N – порядок квадратной матрицы.

Приложение

bigHW.asm

```
1.  ; БПИ-195 Голубков Никита Юрьевич
2.  ; Вариант 3
3.  ;
4.  ; Разработать программу вычисления следа квадратной матрицы
5.  ; порядка N<=5 при условии размещения элементов матрицы
6.  ; в линейном массиве по строкам
7.  ;
8.  ;-----------------first act-------------------------------------------------------
9.  format PE console
10. entry start
11.
12. include 'win32a.inc'
13. include 'macro.inc'
14.
15. section '.data' data readable writable
16.
17.         strVecSize        db 'Enter the size of array A in range from 1 to 5: ', 0
18.         strVecInput       db 'Enter the elements of array:', 10, 0
19.         strVecElInput     db '[%d, %d] = ', 0
20.         strSomethingWrong db 'Something went wrong!', 10, 0
21.         strScanInt        db '%d', 0
22.         strVecElemOut     db 'trA = %d', 10, 0
23.
24.         vec_size      dd 0
25.         trVal         dd 0
26.         step          dd ?
27.         i             dd ?
28.         tmp           dd ?
29.         vec           rd 25
30.
31. ;---------------------second act-----------------------------------------------
32. section '.code' code readable executable
33. start:
34.         ;input the size
35.         invoke printf, strVecSize
36.         invoke scanf, strScanInt, vec_size
37.
38.         ;check size of vector
39.         cmp [vec_size], 5
40.         jg extra_finish
41.         cmp [vec_size], 0
42.         jle extra_finish
43.
44.         invoke printf, strVecInput
45.
46.         ;calling input and calculate macros
47.         VInputMacro vec, vec_size
48.
49.         CalcVecTrMacro vec, vec_size
50.
51.         mov [trVal], eax
52.         invoke printf, strVecElemOut, [trVal]
53.
54.         jmp finish
55.
56. ;extra finish
57. extra_finish:
58.         invoke printf, strSomethingWrong
59.
60. ;finish
61. finish:
62.         invoke getch
63.         invoke ExitProcess, 0
64.
```

```
;------------------------------third act-------------------------
section '.idata' import data readable
    library kernel, 'kernel32.dll',\
            msvcrt, 'msvcrt.dll',\
            user32,'USER32.DLL'

include 'api\user32.inc'
include 'api\kernel32.inc'
    import kernel,\
           ExitProcess, 'ExitProcess'
  include 'api\kernel32.inc'
    import msvcrt,\
           printf, 'printf',\
           scanf, 'scanf',\
           getch, '_getch'
```

macro.inc

```
;vector input macro
macro VInputMacro vecAddr, len {
        xor ecx, ecx      ;ecx = 0
        mov ebx, vecAddr  ;ebx = vecAddr

;loop label
  getVecLoop:
        mov [tmp], ebx

        ;count len * len for size of array
        mov edx, [len]
        imul edx, [len]

        ;check if index greater than size of array
        cmp ecx, edx
        jge endInputVector

        ;counts index div len and index mod len and asks for new item
        mov [i], ecx
        mov eax, [i]
        cdq
        idiv [len]
        invoke printf, strVecElInput, eax, edx

        invoke scanf, strScanInt, ebx

        ;makes step in memory
        mov ecx, [i]
        inc ecx
        mov ebx, [tmp]
        add ebx, 4
        jmp getVecLoop

;end label
  endInputVector:
}

;tr calculate macro
macro CalcVecTrMacro vecAddr, len {
        xor ecx, ecx      ;ecx = 0
        mov ebx, vecAddr  ;ebx = vacAddr
        mov eax, [len]    ;eax = len
        inc eax           ;eax++
        mov [step], eax   ;step = eax
        xor eax, eax      ;eax = 0

;loop label
  trVecLoop:
        mov [tmp], ebx

```

```
51.         ;count len * len for size of array
52.         mov edx, [len]
53.         imul edx, [len]
54.
55.         ;check if index greater than size of array
56.         cmp ecx, edx
57.         jge endTrVector
58.
59.         ;counts tr in eax
60.         add eax, [ebx]
61.
62.         ;makes step in memory
63.         add ecx, [step]
64.         mov ebx, [tmp]
65.         mov edx, [step]
66.         imul edx, 4
67.         add ebx, edx
68.         jmp trVecLoop
69.
70. ;end label
71.   endTrVector:
72. }
```