

```

format PE console
entry start

include 'win32a.inc'

; Кузнецов Михаил Александрович ВПИ199
; 13 вариант
; Массив из элементов не равных минимуму
;-----
section '.data' data readable writable

    strVecSize      db 'Enter size of vector A: ', 0
    strIncorSize    db 'Incorrect size of vector = %d', 10, 0
    strVecElemI     db ' [%d]? ', 0
    strScanInt      db '%d', 0
    strVecElemOut    db ' [%d] = %d', 10, 0
    strNewVector     db 'Vector B:', 10, 0
    strZeroNewVector db 'Vector B is empty', 0

    vec_size        dd 0
    new_vec_size     dd 0
    max_vec_size     dd 100
    i               dd ?
    tmp             dd ?
    tmpStack        dd ?
    min             dd 0
    vec             rd 100
    new_vec         rd 100

;-----
section '.code' code readable executable
start:
; 1) Ввод массива A
    call VectorInput
; 2) Поиск минимального элемента
    call VectorMin
; 3) Заполнение массива B
    call VectorBGenerator
; 4) Вывод нового массива
    push strNewVector
    call [printf]
    call VectorOut

finish:                ; Завершение работы программы
    call [getch]
    push 0
    call [ExitProcess]

;-----Ввод массива и его длины-----
VectorInput:
    push strVecSize
    call [printf]
    add esp, 4

    push vec_size
    push strScanInt      ; считывание длины массива A
    call [scanf]
    add esp, 8

    mov eax, [vec_size]
    cmp eax, [max_vec_size] ; проверка, что введенная длина больше 0 и меньше или равна 100
    jg failSize
    cmp eax, 0
    jg getVector
; fail size
failSize:                ; Вывод сообщения о некорректной длине
    push [vec_size]
    push strIncorSize
    call [printf]
    jmp finish
; else continue...
getVector:                ; Ввод элементов массива
    xor ecx, ecx          ; ecx = 0
    mov ebx, vec          ; ebx = &vec
getVecLoop:
    mov [tmp], ebx
    cmp ecx, [vec_size]
    jge endInputVector    ; to end of loop

    mov [i], ecx
    push ecx
    push strVecElemI      ; Отображение сообщения
    call [printf]
    add esp, 8

    push ebx
    push strScanInt        ; Считывание введенного элемента
    call [scanf]
    add esp, 8

    mov ecx, [i]
    inc ecx
    mov ebx, [tmp]
    add ebx, 4
    jmp getVecLoop
endInputVector:
    ret
;----- Поиск минимального элемента-----
VectorMin:
    xor ecx, ecx          ; ecx = 0
    mov ebx, vec          ; ebx = &vec
    mov ebp, vec          ; min

minVecLoop:
    cmp ecx, [vec_size]

```

```

        je endMinVector          ; to end of loop

        cmp ebp,[ebx]
        jle nextMinVecLoop       ; Если элемент не меньше минимума то переходим
        mov ebp,[ebx]           ; сохранение нового минимального элемента
        mov [min], ebp

nextMinVecLoop:
    inc ecx
    add ebx, 4
    jmp minVecLoop
endMinVector:
    ret
;-----Создание нового массив-----
VectorBGenerator:
    mov [tmpStack], esp
    xor ecx, ecx                ; ecx = 0
    mov ebx, vec                ; ebx = &vec
    mov edx,new_vec             ; edx = &new_vec

newVecLoop:
    cmp ecx, [vec_size]
    je endNewSumVector          ; to end of loop
    mov eax,[min]
    cmp eax,[ebx]
    je nextNewVecLoop           ; Пропуск элементов равных минимальному

    inc [new_vec_size]
    mov ebp, [ebx]              ; Добавление элемента в новый массив
    mov [edx], ebp
    add edx,4

nextNewVecLoop:
    inc ecx
    add ebx, 4
    jmp newVecLoop
endNewSumVector:
    mov esp, [tmpStack]
    ret
;-----Вывод нового массива-----
VectorOut:
    mov [tmpStack], esp
    xor ecx, ecx                ; ecx = 0
    mov ebx, new_vec            ; ebx = &new_vec

    cmp [new_vec_size],0        ; Проверка длины нового массива
    je zeroElements             ; Если длина равна 0 значит в массиве нет элементов и отображается соответствующее сообщение

putVecLoop:
    mov [tmp], ebx
    cmp ecx, [new_vec_size]
    je endOutputVector          ; to end of loop
    mov [i], ecx

    push dword [ebx]
    push ecx                    ; output element
    push strVecElemOut
    call [printf]

    mov ecx, [i]
    inc ecx
    mov ebx, [tmp]
    add ebx, 4
    jmp putVecLoop

zeroElements:                   ; Отображение сообщения о пустом массиве B
    push strZeroNewVector
    call [printf]
endOutputVector:
    mov esp, [tmpStack]
    ret
;-----
section '.idata' import data readable
    library kernel, 'kernel32.dll',\
        msvcrt, 'msvcrt.dll'

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

```

```

Enter size of vector A: 4
[0]? 2
[1]? -5
[2]? 22
[3]? -7
Vector B:
[0] = 2
[1] = -5
[2] = 22

```

```

Enter size of vector A: 0
Incorrect size of vector = 0

```

```
Enter size of vector A: 101  
Incorrect size of vector = 101
```

```
Enter size of vector A: 4  
[0]? 1  
[1]? 2  
[2]? 3  
[3]? 2  
Vector B:  
[0] = 2  
[1] = 3  
[2] = 2
```

```
Enter size of vector A: 3  
[0]? 1  
[1]? 1  
[2]? 1  
Vector B:  
Vector B is empty
```