

Федеральное государственное автономное образовательное учреждение высшего  
образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

**САМОСТОЯТЕЛЬНАЯ РАБОТА**

Вычисление суммы положительных и отрицательных чисел в целочисленном массиве  
машинных слов длиной  $N=15$  элементов

по направлению подготовки Архитектура вычислительных систем  
образовательная программа «Программная инженерия»

Выполнил:

Студент группы БПИ196

Хожаахмедов Бобурбек

---

Преподаватель:

Легалов Александр Иванович

---

Москва 2020

## Задание

Разработать программу вычисления суммы положительных и отрицательных чисел в целочисленном массиве машинных слов длиной  $N=15$  элементов.

## Составление программы

Алгоритм решения задачи заключается в том, что сначала считываются входные параметры с проверкой на преобразование в строки в число, а затем выводится в консоль каждое прошлое число с последующим запросом ввода. Если же не получается преобразовать строку в число, то происходит запрос на ввод этого же элемента массива.

Далее для нахождения суммы положительных и отрицательных чисел описаны отдельные функции `sumpos` и `sumneg`. Принцип работы примерно одинаковый, первым делом создается кадр стека и сохраняются необходимые регистры. Следующим шагом является определение начала массива и создание переменной для хранения суммы, равной 0. Каждый элемент массива проверяется на положительность или отрицательность, если этот элемент не подходит, то пропускается этот элемент и происходит переход к следующему, если подходит, то прибавляется к сумме и переходит к следующему элементу.

Также написана функция `show` в которой происходит вывод всего массива.

## Текст программы:

```
format PE GUI
format PE CONSOLE ; 32-разрядка консольная программа
WINDOWS EXE
entry start ; точка входа

include 'include\win32a.inc'

section '.idata' import data readable ;секция
импортируемых функций

library kernel,'kernel32.dll',\
        msvcrt,'msvcrt.dll'
```

```

import kernel,\
        ExitProcess,'ExitProcess'

import msvcrt,\
        sscanf,'sscanf',\
        gets,'gets',\
        getch,'_getch',\
        printf,'printf'

section '.data' data readable writeable      ;секция
данных
n equ 15
A      dd 15 dup (0)
s      db 128 dup (0)
msg1   db 'Input array:',13,10,0
msg2   db 'Array:',13,10,0
msg3   db 13,10,'Sum of positives = %d',13,10,0
msg4   db 'Sum of negatives = %d',13,10,0
str0   db 'A[%d] = ',0 ;формируемая для входа строка
fmt     db '%d',0
fmt1   db '%5d',0

section '.code' code readable executable      ; секция
кода
start:                                       ; точка
входа в программу
        ccall [printf],msg1                ; вывод сообщения
        ccall input,A,n                    ; ввод массива
        ccall [printf],msg2                ; вывод сообщения
        ccall show,A,n                    ; ввод массива
        ccall sumpos,A,n                   ; сумма положительных
чисел
        ccall [printf],msg3,eax            ; вывод суммы
        ccall sumneg,A,n                   ; сумма отрицательных
чисел
        ccall [printf],msg4,eax            ; вывод суммы
        ccall [getch]                      ; ожидаем нажатие
любой клавиши
        stdcall [ExitProcess], 0;выход
;void input(int *a,int n)
;вывод массива
;соглашение вывода cdecl
input:
        push ebp                           ;создать кадр стека
        mov ebp,esp
        push ebx                           ;сохранить регистры по
соглашению stdcall

```

```

        push esi
; вывод массива
        mov ebx,[ebp+8]          ; начало массива
        mov esi,1                ; номер элемента
inpmas: ccall [printf],str0,esi  ; вывод сообщения
        ccall [gets],s           ; вывод строки
        ccall [sscanf],s,fmt,ebx; преобразование строки
в число
        test eax,eax             ; если ошибка
        jle inpmas               ; повторить ввод
        add ebx,4                 ; следующий элемент
        inc esi                  ; номер следующего
элемента
        cmp esi,[ebp+12]         ; сравниваем с размером
массива
        jbe inpmas               ; если меньше или
равно, продолжить
        pop esi                  ; восстановить регистры
        pop ebx
        pop ebp                  ; эпилог функции
        ret
; void show(int *a,int n)
; вывод массива
; соглашение вывода cdecl
show:
        push ebp                 ; создать кадр стека
        mov ebp,esp
        push ebx                 ; сохранить регистры по
соглашению stdcall
        push esi
; вывод массива
        mov esi,[ebp+12]         ; количество элементов
в массиве
        mov ebx,[ebp+8]          ; начало массива
mo6:    ccall [printf],fmt1,[ebx] ; вывод
элемента массива
        add ebx,4                 ; переход к следующему
элементу массива
        dec esi                  ; уменьшаем количество
оставшихся элементов

        jnz mo6                  ; продолжить пока не
ноль
        pop esi                  ; восстановить регистры
        pop ebx
        pop ebp                  ; эпилог функции
        ret
; int sumpos(int *a,int n)
; сумма положительных элементов массива

```

```

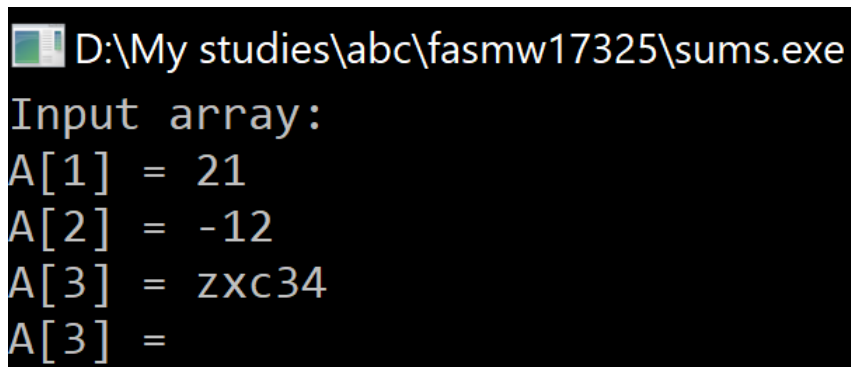
;соглашение вызова cdecl
sumpos:
    push ebp                                ;создать кадр стека
    mov ebp,esp
    push ebx                                ;сохранить регистры по
соглашению stdcall
    mov ecx,[ebp+12]                        ;количество элементов
в массиве
    mov ebx,[ebp+8]                        ;начало массива
    mov eax,0                              ;сумма положительных=0
s1:    cmp dword [ebx],0                   ;если число не
положительное
    jng s2                                  ;пропустить
    add eax,[ebx]                           ;прибавить к сумма
положительных
s2:    add ebx,4                            ;переходим к
следующему элементу массива
    loop s1                                ;цикл по массиву
    pop ebx                                 ;восстановить регистры
    pop ebp                                 ;эпилог функции
    ret

;int sumneg(int *a,int n)
;сумма отрицательных элементов массива
;соглашение вызова cdecl
sumneg:
    push ebp                                ;создать кадр стека
    mov ebp,esp
    push ebx                                ;сохранить регистры по
соглашению stdcall
    mov ecx,[ebp+12]                        ;количество элементов
в массиве,[ebp+8]
    mov ebx,[ebp+8]                        ;начало массива
    mov eax,0                              ;сумма отрицательных=0
s3:    cmp dword [ebx],0                   ;если число не
отрицательно
    jnl s4                                  ;пропустить
    add eax,[ebx]                           ;прибавить к сумме
отрицательных
s4:    add ebx,4                            ;переходим к
следующему элементу массива
    loop s3                                ;цикл по массиву
    pop ebx                                 ;восстановить регистры
    pop ebp                                 ;эпилог функции
    ret

```

## Тестирование программы

Поскольку программа в качестве входного параметра принимает целое число, для этого программа запрашивает ввод с клавиатуры каждый элемент массива



A screenshot of a Windows command prompt window. The title bar at the top reads "D:\My studies\abc\fasmw17325\sums.exe". The command prompt shows the text "Input array:" followed by four lines of input: "A[1] = 21", "A[2] = -12", "A[3] = zxc34", and "A[3] =".

```
D:\My studies\abc\fasmw17325\sums.exe
Input array:
A[1] = 21
A[2] = -12
A[3] = zxc34
A[3] =
```

Также предусмотрен ввод неверных значений, при вводе которых программа запрашивает ввод повторно.

В результате работы программа выведет

D:\My studies\abc\fasmw17325\sums.exe

Input array:

A[1] = 21

A[2] = -12

A[3] = zxc34

A[3] = 12

A[4] = 11

A[5] = -234

A[6] = 1232

A[7] = 23

A[8] = -568

A[9] = 45

A[10] = 34

A[11] = 234

A[12] = 12

A[13] = -34

A[14] = -55

A[15] = 34

Array:

21 -12 12 11 -234 1232 23 -568 45 34 234 12 -34 -55 34

Sum of positives = 1658

Sum of negatives = -903