

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Конструирование программ

ОТЧЁТ
к лабораторной работе
на тему

ЛОГИЧЕСКИЕ КОМАНДЫ. КОМАНДЫ РАБОТЫ С БИТОВЫМИ ПОЛЯМИ.
КОМАНДЫ СДВИГОВ

Выполнил: студент группы 153501
Тимофеев К.А.

Проверил: Туровец Н.О.

Минск 2022

Вариант 19

Ввести матрицу целых чисел размерностью 5x6 элементов. Найти номера столбцов с минимальной суммой элементов

Код программы

```
.model small
.stack 50h
.code
start:
    mov ax, data
    mov dx, 0
    mov ds, ax
    mov cx, 30
    mov ah, 01h
    mov di, offset array

loop1:
    call read_number
    cmp al, 1
    je _error
    add di, 2
    loop loop1

    mov cx, 6
    mov bx, offset array

loop3:
    call sumofcolumn
    push bx
    push ax

    mov bx, offset currentsum
    mov ah, [bx+2]
    mov bx, offset min
    mov al, [bx+2]
    cmp ah, al
    jl load_to_min
    mov ax, word ptr currentsum
    cmp ax, word ptr min
    jl load_to_min
    jmp end_of_iter

load_to_min:
    mov ax, word ptr currentsum
    mov word ptr min, ax
    mov bx, offset currentsum
    mov al, [bx+2]
    mov bx, offset min

    mov bx, offset min
    mov [bx+2], al
    mov di, 7
    sub di, cx

end_of_iter:
    mov word ptr currentsum, 0
    mov bx, offset currentsum
    mov [bx+2], 0
    pop ax
    pop bx
    add bx, 1
    loop loop3

    mov ah, 09h
    mov bx, offset answer
    add bx, 46
    add di, 30h
    mov [bx], di
    mov dx, offset answer
    int 21h
    jmp end_of_program

_error:
    mov ah, 9
    mov dx, offset error_message
    int 21h

end_of_program:
    mov ax, 4C00h
    int 21h

read_number proc near ;
    push cx
    mov ah, 01h
    mov cl, 0
    mov al, 0
    int 21h
    cmp al, '-'
    jne @1
    mov cl, 1
loop2:
    mov al, 0
    int 21h
    @1:
    cmp al, '0'
    jb exit2
    cmp al, '9'
    ja exit2
    sub al, 30h
    push ax
    mov ax, 0Ah
    mov bx, [di]
    mul word ptr [di]

no_of:
    mov bx, ax
    pop ax
    push ax
    mov ah, 0
    add bx, ax
    pop ax
    push cx
    add cx, 7FFFh
    cmp bx, cx
    pop cx
    ja check_return
    mov [di], bx
    jmp loop2

exit2:
    cmp cl, 1
```

```

        jne check_return
        push cx
        mov cx, [di]
        neg cx
        mov [di], cx
        pop cx
check_return:
        cmp al, 0Dh
        je enter
        cmp al, ' '
        je return
        mov al, 1
        jmp return
return:
        pop cx
        ret 0
enter:
        call newl
        jmp return
read_number endp

newl proc nier uses ax, dx
        mov dx, offset newline
        mov ah, 09h
        int 21h

        ret
newl endp

to3b proc nier ;in si: first byte
        push ax
        push bx

        mov ax, [bx+si]
        mov word ptr buff, ax

        mov bx, offset buff

        cmp ax, 0
        jns not_neg
        mov [bx,2], FFh
        jmp ex
not_neg:
        mov [bx+2], 0
ex:
        pop bx
        pop ax
        ret
to3b endp

.data
array dw 30 dup (0)
error_message db 0Ah, 0Dh, "Error", 0Ah, 0Dh, '$'
min db 0FFh, 0FFh, 7Fh
newline db 0Ah, 0Dh, '$'
currentsum db 0,0,0
buff db 0,0,0
answer db "The column with the lowest sum of elements is ", 0, 0Ah, 0Dh, '$'
end start

sumofcolumn proc nier uses ax; b:
        push ax
        push cx
        mov ax, 0
        mov si, 0
        mov cx, 5
loopS:
        call to3b ;in 3 byte format
        mov ax, word ptr currentsum
        add ax, word ptr buff; addti
        mov word ptr currentsum, ax;
        push bx ;save bx in stack
        mov al, 0; zero in al
        adc al, 0 ; cf in al
        mov bx, offset buff; adress i
        add bx, 2; to last byte
        add al, [bx]; addition
        mov bx, offset currentsum; a
        add bx, 2; to last byte
        add al, [bx]; addition
        mov [bx], al; to sum
        pop bx; load bx from stack
        add si, 12 ;
        loop loopS
        pop cx
        pop ax
        ret
sumofcolumn endp

```

Консольный ввод

```

1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
19 20 21 22 23 24
25 26 27 28 29 30

```

Значения в регистрах после выполнения программы

```
The column with the lowest sum of elements is 1
```