Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Конструирование программ»

**ОТЧЕТ**

к лабораторной работе №5

на тему:

**«Целочисленные арифметические операции. Обработка массивов числовых данных»**

БГУИР 1-40-04-01

|  |
| --- |
| Выполнил студент группы 253504  ЛЯНГО Иван Геннадьевич |
|  |
| (дата, подпись студента) |
| Проверил ассистент кафедры информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2023

**Цель работы:** Вариант 18. Ввести матрицу целых чисел размерностью 5х6 элементов. Найти номера строк с максимальным произведением элементов.

**Ход работы:** на рисунке 1 представлены изначальные значения регистров, на рисунке 2 – изначальные значения переменных, на рисунках 3 и 4 – соответствующие значения после выполнения программы, на рисунке 5 – окно для работы с программой.

Листинг 1 – Исходный код программы задания 1

.model tiny

.code

org 100h

start:

mov di,0

read\_matrix:

cmp di,20 ;30 words // 10 words

je process\_matrix

call read\_number

mov [matrix+di],bx

add di,2

mov ah,9

lea dx,new\_line

int 21h

jmp read\_matrix

process\_matrix:

mov si,0 ;element index out of 60 bytes

mov di,0 ;column index

mov cx,1 ;mult

process\_loop:

cmp si,20

je handle\_row\_end

;add si,1

cmp di,10

;sub si,1

je handle\_row\_end

mov ax,[matrix+si] ;first byte of word

imul cx

mov cx,ax

jo error3

add si,2

add di,2

jmp process\_loop

after\_handle\_row\_end:

cmp si,20

je final

jmp process\_loop

handle\_row\_end:

cmp cx,max\_row\_value

jg change\_max\_values

inc row\_counter

jmp after\_handle\_row\_end

change\_max\_values:

mov max\_row\_value,cx

mov di,row\_counter

mov max\_row\_index,di

mov cx,1

mov di,0

inc row\_counter

jmp after\_handle\_row\_end

error1:

mov ah,9

lea dx,new\_line

int 21h

mov ah,9

lea dx,error\_msg1

int 21h

hlt

error2:

mov ah,9

lea dx,new\_line

int 21h

mov ah,9

lea dx,error\_msg2

int 21h

hlt

error3:

mov ah,9

lea dx,new\_line

int 21h

mov ah,9

lea dx,error\_msg3

int 21h

hlt

set\_negative:

mov bp,-1

jmp after\_set\_negative

final:

mov dx,max\_row\_value

mov number\_to\_print,dx

jmp push\_to\_stack\_number

after\_push\_to\_stack\_number:

mov ah,9

lea dx,new\_line

int 21h

mov ah,9

lea dx,msg2

int 21h

jmp print\_number\_from\_stack

after\_print\_number\_from\_stack:

mov ah,9

lea dx,new\_line

int 21h

mov ah,9

lea dx,msg3

int 21h

inc max\_row\_index

mov bx,max\_row\_index

add bx,48

mov ah,6

mov dx,bx

int 21h

hlt

read\_number:

mov ax,@data

mov ds,ax

mov ah,9

mov dx,offset msg

int 21h

mov ah,0Ah

mov dx,offset string

int 21h

call string2number

ret

proc string2number

mov bp,1

mov si,offset string + 1

cmp [si+1],'-'

je set\_negative

after\_set\_negative:

mov cl,[si]

mov ch,0

add si,cx

mov bx,0

mov dx,0

repeat:

mov al,[si]

cmp al,02dh

je if\_minus

sub al,48

cmp al,0

jl error1

cmp al,9

ja error1

mov ah,0

imul bp

add bx,ax

mov ax,bp

mov bp,10

imul bp

mov bp,ax

dec si

loop repeat

ret

if\_minus:

ret

endp

push\_to\_stack\_number:

mov di,0

cmp number\_to\_print,0

jl add\_minus

push\_to\_stack\_loop:

cmp number\_to\_print,0

je after\_push\_to\_stack\_number

mov ax,number\_to\_print

mov bl,10

mov dx,0

div bx

mov number\_to\_print,ax

add dx,48

push dx

jmp push\_to\_stack\_loop

add\_minus:

mov dx,'-'

mov ah,6

int 21h

neg number\_to\_print

jmp push\_to\_stack\_loop

print\_number\_from\_stack:

cmp sp,0

je after\_print\_number\_from\_stack

pop dx

mov ah,6

int 21h

jmp print\_number\_from\_stack

terminate:

hlt

.data

matrix dw 30 dup(?)

max\_row\_index dw 0

max\_row\_value dw 8000h

row\_counter dw 0

current\_row\_value dd 0h

number\_to\_print dw ?

number\_string db 5 dup('$')

new\_line db 0Ah,0Dh,'$'

error\_msg1 db 'Error: wrong input$'

error\_msg2 db 'Error: number out of bound$'

error\_msg3 db 'Error: multiplication overflow$'

msg db 'Enter a number: $'

msg2 db 'Biggest multiplication: $'

msg3 db 'Row number: $'

string db 5 ;MAX NUMBER OF CHARACTERS ALLOWED (4).

db ? ;NUMBER OF CHARACTERS ENTERED BY USER.

db 5 dup (?) ;CHARACTERS ENTERED BY USER.

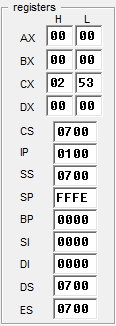


Рисунок 1 – Значения регистров программы перед выполнением

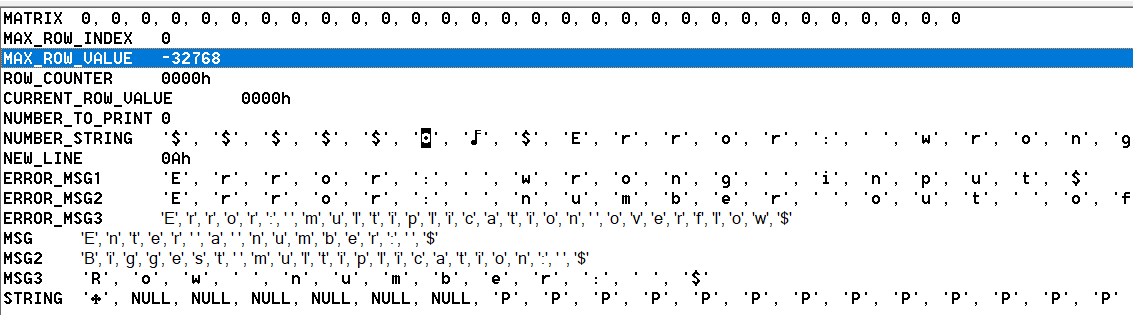


Рисунок 2 – Значение переменных программы перед выполнением

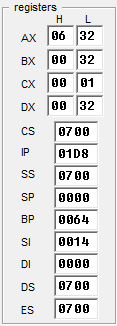


Рисунок 3 – Значения регистров программы после выполнения

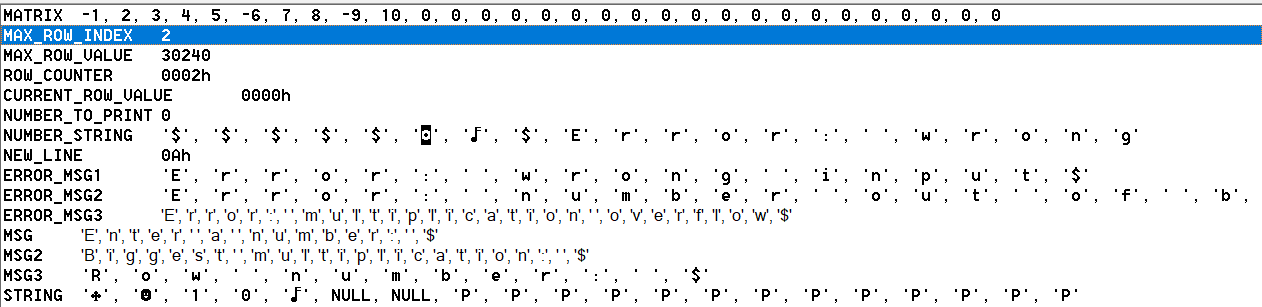


Рисунок 4 – значение переменных после выполнения программы после выполнения

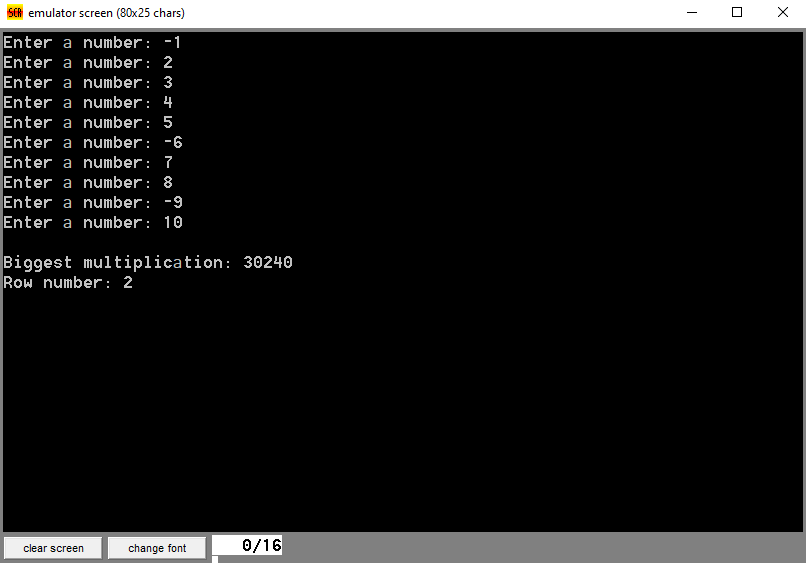


Рисунок 5 – окно для работы с программой

**Выводы:** в результате лабораторной работы была выполнена одна задача с использованием команд JMP для безусловного перехода к метке подпрограммы, CALL для вызова процедуры, LEA для загрузки адреса переменной в регистр, JO, JG, JL, JE для условных переходов, INT для вызова прерываний, INC для инкремента, DEC для декремента, MOV для сохранения значений, IDIV для деления знаковых чисел, DIV для деления беззнаковых чисел, CMP для сравнения значений, LOOP для работы с циклами, PUSH и POP для работы со стеком, HLT для завершения программы.