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 5x6 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.
```

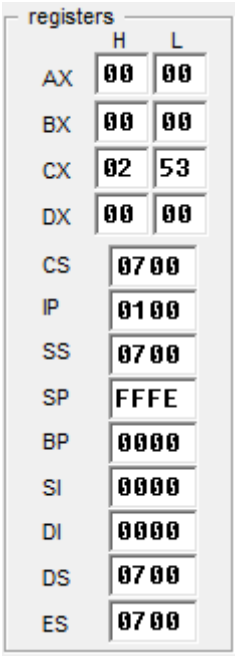

Рисунок 1 – Значения регистров программы перед выполнением

```
MATRIX  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
MAX_ROW_INDEX   0
MAX_ROW_VALUE   -32768
ROW_COUNTER     0000h
CURRENT_ROW_VALUE       0000h
NUMBER_TO_PRINT 0
NUMBER_STRING   '$', '$', '$', '$', '$', '◘', '♩', '$', 'E', 'r', 'r', 'o', 'r', ':', ' ', 'w', 'r', 'o', 'n', 'g
NEW_LINE        0Ah
ERROR_MSG1      'E', 'r', 'r', 'o', 'r', ':', ' ', 'w', 'r', 'o', 'n', 'g', ' ', 'i', 'n', 'p', 'u', 't', '$'
ERROR_MSG2      'E', 'r', 'r', 'o', 'r', ':', ' ', 'n', 'u', 'm', 'b', 'e', 'r', ' ', 'o', 'u', 't', ' ', 'o', 'f
ERROR_MSG3      'E', 'r', 'r', 'o', 'r', ':', ' ', 'm', 'u', 'l', 't', 'i', 'p', 'l', 'i', 'c', 'a', 't', 'i', 'o', 'n', ' ', 'o', 'v', 'e', 'r', 'f', 'l', 'o', 'w', '$'
MSG     'E', 'n', 't', 'e', 'r', ' ', 'a', ' ', 'n', 'u', 'm', 'b', 'e', 'r', ':', ' ', '$'
MSG2    'B', 'i', 'g', 'g', 'e', 's', 't', ' ', 'm', 'u', 'l', 't', 'i', 'p', 'l', 'i', 'c', 'a', 't', 'i', 'o', 'n', ':', ' ', '$'
MSG3    'R', 'o', 'w', ' ', 'n', 'u', 'm', 'b', 'e', 'r', ':', ' ', '$'
STRING  '♣', NULL, NULL, NULL, NULL, NULL, NULL, 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P'
```

Рисунок 2 – Значение переменных программы перед выполнением

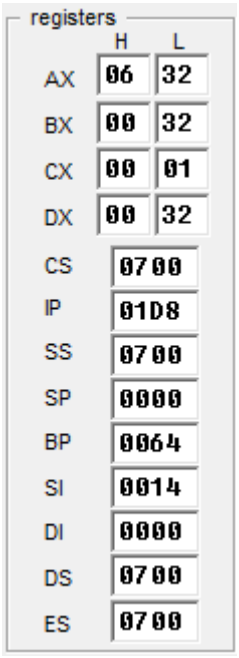

Рисунок 3 – Значения регистров программы после выполнения

```
MATRIX  -1, 2, 3, 4, 5, -6, 7, 8, -9, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
MAX_ROW_INDEX   2
MAX_ROW_VALUE   30240
ROW_COUNTER     0002h
CURRENT_ROW_VALUE       0000h
NUMBER_TO_PRINT 0
NUMBER_STRING   '$', '$', '$', '$', '$', '◘', '♩', '$', 'E', 'r', 'r', 'o', 'r', ':', ' ', 'w', 'r', 'o', 'n', 'g'
NEW_LINE        0Ah
ERROR_MSG1      'E', 'r', 'r', 'o', 'r', ':', ' ', 'w', 'r', 'o', 'n', 'g', ' ', 'i', 'n', 'p', 'u', 't', '$'
ERROR_MSG2      'E', 'r', 'r', 'o', 'r', ':', ' ', 'n', 'u', 'm', 'b', 'e', 'r', ' ', 'o', 'u', 't', ' ', 'o', 'f', ' ', 'b',
ERROR_MSG3      'E', 'r', 'r', 'o', 'r', ':', ' ', 'm', 'u', 'l', 't', 'i', 'p', 'l', 'i', 'c', 'a', 't', 'i', 'o', 'n', ' ', 'o', 'v', 'e', 'r', 'f', 'l', 'o', 'w', '$'
MSG     'E', 'n', 't', 'e', 'r', ' ', 'a', ' ', 'n', 'u', 'm', 'b', 'e', 'r', ':', ' ', '$'
MSG2    'B', 'i', 'g', 'g', 'e', 's', 't', ' ', 'm', 'u', 'l', 't', 'i', 'p', 'l', 'i', 'c', 'a', 't', 'i', 'o', 'n', ':', ' ', '$'
MSG3    'R', 'o', 'w', ' ', 'n', 'u', 'm', 'b', 'e', 'r', ':', ' ', '$'
STRING  '+', '☻', '1', '0', '♩', NULL, NULL, 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P'
```

Рисунок 4 – значение переменных после выполнения программы после выполнения

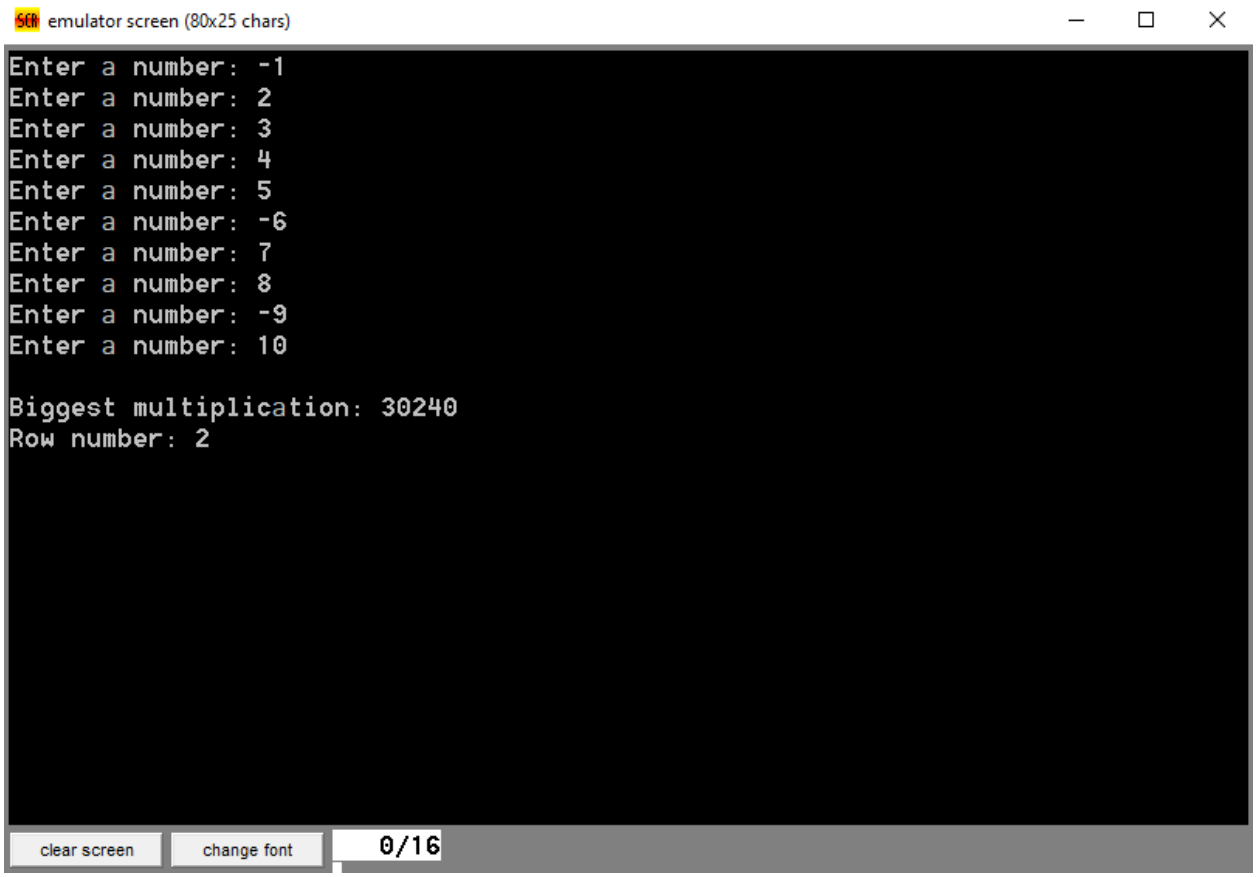

Рисунок 5 – окно для работы с программой

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.