МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3
по дисциплине «Организация систем и ЭВМ»
Тема «Представление и обработка целых чисел. Организация ветвящихся процессов»

Студентка гр. 1303 _____ Чернуха В.В.

Преподаватель _____ Ефремов М.А

Санкт-Петербург
2022

Цель работы.

Разработать на языке Ассемблера программу, которая вычисляет значения функций по заданным целочисленным значениям.

Задание.

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров a, b, i, k вычисляет:

a) значения функций i1 = f1(a,b,i) и i2 = f2(a,b,i);

b) значения результирующей функции res = f3(i1,i2,k),

где вид функций f1 и f2 определяется из табл. 2, а функции f3 - из табл.3 по цифрам шифра индивидуального задания (n1,n2,n3), приведенным в табл.4. Значения a, b, i, k являются исходными данными, которые должны выбираться студентом самостоятельно и задаваться в процессе исполнения программы в режиме отладки. При этом следует рассмотреть всевозможные комбинации параметров a, b и k, позволяющие проверить различные маршруты выполнения программы, а также различные знаки параметров a и b.

Согласно 1 варианту взяты следующие функции f1, f2, f3:

$$f1 = \begin{cases} 15-2*i, & \text{при } a>b \\ 3*i+4, & \text{при } a<=b \end{cases}$$

$$f2 = \begin{cases} -(4*i+3), & \text{при } a>b \\ 6*i-10, & \text{при } a<=b \end{cases}$$

$$f1 = \begin{cases} \min(i1,i2), & \text{при } k=0 \\ \max(i1,i2), & \text{при } k/=0 \end{cases}$$

Выполнение работы. Протокол работы на компьютере.

Ход работы:

1. Получен набор функций f1, f2, f3 согласно варианту.

2. Программа протранслирована и запущена в режиме отладчика в пошаговом режиме

В сегменте данных задаются метки для переменных i, a, b, k, i1, i2, res и в них записываются нули. Так как функции использовать запрещено, некоторые фрагменты кода размечены метками для перехода на них. Для реализации ветвления в программе используются команды условного перехода вида J** <метка>. Поведение программы зависит от переменных a, b, k. Сначала сравниваются значения a и b и в зависимости от результата сравнения функции f1 и f2 вычисляются по разному (если a>b то выполняется переход на метку part2). Далее вычисляется f3 в зависимости от значения k. Для нахождения f3 нам нужно найти либо max(i1,i2), либо min(i1,i2), для чего нам снова придётся сравнивать значения и использовать ветвления. В конце выполнения программы в res записывается ответ и программа завершается.

Вывод.

В ходе выполнения лабораторной работы было на практике изучено ветвление на языке Ассемблера, разработана программа вычисляющая значение функции по заданным целочисленным параметрам.

ПРИЛОЖЕНИЕ А

Текст исходного файла программы lab3.asm

```
ASSUME CS:CODE, SS:AStack, DS:DATA

AStack    SEGMENT  STACK
          DW 32 DUP(0)
AStack    ENDS

DATA      SEGMENT

i     DW   0
a     DW   0
b     DW   0
k     DW   0
i1    DW   0
i2    DW   0
res   DW   0

DATA      ENDS

CODE SEGMENT

Main      PROC  FAR
    mov   AX,DATA
    mov   DS,AX
    mov cx, i
    mov AX, a
    cmp AX, b
    JG part2      ; if a>b go to 2
    sal cx, 1
    add cx, i
    add cx, 4
    mov i1, cx

    sal cx, 1
    sub cx, 18
    mov i2, cx
    jmp f3

part2:
    sal cx, 1
    mov ax, 15
    sub ax, cx
    mov i1, ax

    sal ax, 1
    sub ax, 33
    mov i2, ax

f3:
    cmp k, 0
    JE min ;if k==0 go to min
```

```
    mov ax, i1
    cmp ax, i2
    jge res1      ; if i1 >= i2

res2:
    mov ax, i2
    mov res, ax
    jmp endprog

min:
    mov ax, i1
    cmp ax, i2
    jge res2       ; if i1 >= i2 go to i2

res1:
    mov ax, i1
    mov res, ax

endprog:
    int 20h

    Main      ENDP
    CODE      ENDS
          END Main
```

ПРИЛОЖЕНИЕ Б

Листинг lab3.lst

```
     #Microsoft      (R)      Macro      Assembler      Version      5.10
11/3/22 00:45:29
                                                                      Page
1-1


                        ASSUME CS:CODE, SS:AStack, DS:DATA

 0000                   AStack    SEGMENT  STACK
 0000  0020[                            DW 32 DUP(0)
        0000
                ]

 0040                   AStack    ENDS

 0000                   DATA      SEGMENT

 0000  0000                  i     DW    0
 0002  0000                  a     DW    0
 0004  0000                  b     DW    0
 0006  0000                  k     DW    0
 0008  0000                  i1    DW    0
 000A  0000                  i2    DW    0
 000C  0000                  res   DW    0

 000E                   DATA      ENDS

 0000                   CODE SEGMENT

 0000                   Main      PROC  FAR
 0000  B8 ---- R             mov   AX,DATA
 0003  8E D8                  mov   DS,AX
 0005  8B 0E 0000 R          mov cx, i
 0009  A1 0002 R         mov AX, a
 000C  3B 06 0004 R           cmp AX, b
 0010  7F 19                  JG part2      ; if a>b go to 2
 0012  D1 E1                  sal cx, 1
 0014  03 0E 0000 R           add cx, i
 0018  83 C1 04               add cx, 4
 001B  89 0E 0008 R           mov i1, cx

 001F  D1 E1                  sal cx, 1
 0021  83 E9 12               sub cx, 18
 0024  89 0E 000A R           mov i2, cx
 0028  EB 13 90               jmp f3

 002B                   part2:
 002B  D1 E1                  sal cx, 1
 002D  B8 000F                mov ax, 15
 0030  2B C1                  sub ax, cx
```

```
 0032  A3 0008 R          mov i1, ax

 0035  D1 E0                  sal ax, 1
 0037  2D 0021                sub ax, 33
 003A  A3 000A R          mov i2, ax

 003D                 f3:
 003D  83 3E 0006 R 00              cmp k, 0
 0042  74 12                  JE min ;if k==0 go to min
 0044  A1 0008 R          mov ax, i1
#Microsoft (R) Macro Assembler Version 5.10                    11/3/22
00:45:29
                                                                 Page
1-2


 0047  3B 06 000A R            cmp ax, i2
 004B  7D 12                   jge res1      ; if i1 >= i2

 004D                 res2:
 004D  A1 000A R          mov ax, i2
 0050  A3 000C R          mov res, ax
 0053  EB 10 90                jmp endprog

 0056                 min:
 0056  A1 0008 R          mov ax, i1
 0059  3B 06 000A R            cmp ax, i2
 005D  7D EE                   jge res2       ; if i1 >= i2 go to i2

 005F                 res1:
 005F  A1 0008 R          mov ax, i1
 0062  A3 000C R          mov res, ax

 0065                 endprog:
 0065  CD 20                   int 20h

 0067                      Main      ENDP
 0067                      CODE      ENDS
                                END Main
#Microsoft (R) Macro Assembler Version 5.10                    11/3/22
00:45:29
                                                              Symbols-1


Segments and Groups:

                N a m e         Length     Align     Combine Class

ASTACK . . . . . .              0040  PARA  STACK
CODE . . . . . .                0067  PARA  NONE
DATA . . . . . .                000E  PARA  NONE

Symbols:
```

```
                N a m e         Type  Value     Attr

A  . . . . . .                  L WORD    0002  DATA

B  . . . . . .                  L WORD    0004  DATA

ENDPROG  . . . . . .            L NEAR    0065  CODE

F3 . . . . . .                  L NEAR    003D  CODE

I  . . . . . .                  L WORD    0000  DATA
I1 . . . . . .                  L WORD    0008  DATA
I2 . . . . . .                  L WORD    000A  DATA

K  . . . . . .                  L WORD    0006  DATA

MAIN . . . . . .                F PROC    0000  CODE  Length     =
0067
MIN  . . . . . .                L NEAR    0056  CODE

PART2  . . . . . .              L NEAR    002B  CODE

RES  . . . . . .                L WORD    000C  DATA
RES1 . . . . . .                L NEAR    005F  CODE
RES2 . . . . . .                L NEAR    004D  CODE

@CPU . . . . . .                TEXT  0101h
@FILENAME  . . . . . .          TEXT  lab3
@VERSION . . . . . .            TEXT  510


     74 Source  Lines
     74 Total   Lines
     22 Symbols

  48056 + 461251 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```