

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

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

Студентка гр. 1303

Королева П.А

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

## Цель работы.

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

## Задание.

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

- а) значения функций  $i1 = f1(a,b,i)$  и  $i2 = f2(a,b,i)$ ;
- б) 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  $b$ .

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

$$f4 = \begin{cases} / -(6*i - 4), & \text{при } a > b \\ \backslash 3*(i+2), & \text{при } a \leq b \end{cases}$$

$$f7 = \begin{cases} / |i1| + |i2|, & \text{при } k < 0 \\ \backslash \max(6, |i1|), & \text{при } k \geq 0 \end{cases}$$

## Выполнение работы.

1. Первые две функции имеют одинаковые условия, поэтому их можно совместить. В одном блоке высчитываются значения если  $a > b$ , в другом SecondSituation если  $a \leq b$ .
2. Т.к в обоих случаях потребуется значение  $i*4$ , высчитываем его сразу.

3. Находим значения  $i1$   $i2$  в зависимости от  $a$ ,  $b$ .
  4. Для вычисления значения третьей функции, находим модуль  $i1$ , тк он потребуется при любом исходе сравнения.
  5. В зависимости от результата сравнения проводим нужные расчеты.
- Текст исходного файла программы `lr3.asm` представлен в приложении А.

### **Выводы.**

Была разработана программа на языке Ассемблера, которая по заданным целочисленным значениям параметров вычисляет значения функций.

## ПРИЛОЖЕНИЕ А

Название файла: lr3.asm

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

```
AStack    SEGMENT    STACK
           DW 12 DUP(?)
AStack    ENDS
```

```
DATA      SEGMENT
```

```
a        DW    0
b        DW    0
i        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 ax, a
           mov dx, i
```

```
           sal dx, 1
           add i1, dx
           add i1, dx
```

```
           cmp ax, b
           jle SecondSituation
```

```
           add i1, 3; i*4 + 3
           neg i1
           mov bx, i1
           mov i2, bx
           sub i2, dx
           add i2, 7
           jmp EndOfFunction1_2
```

```
SecondSituation:
```

```
           add i1, dx
           sub i1, 10
```

```
           mov bx, i1
           mov i2, bx
           sar i2, 1
           add i2, 11
```

```

EndOfFunction1_2:
mov    ax,i1
getabs1:
neg ax
js  getabs1

mov cx, k
cmp cx, 0
jge ElsePart

mov    bx,i2
getabs2:
neg bx
js  getabs2

add bx, ax
mov res, bx
jmp EndOfFunc3

ElsePart:
cmp ax, 6
jl  secondMax
mov res, ax
jmp EndOfFunc3
secondMax:
mov res, 6

EndOfFunc3:
int 20h

Main      ENDP
CODE      ENDS
          END Main

```