МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**“УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”**

Факультет информационных систем и технологий

Кафедра “ Вычислительная техника”

Дисциплина “Машинно-ориентированное программирование”

**Лабораторная работа №3**

«Обработка чисел и структурированных данных»

Вариант 3

Выполнил:

студент гр. ИВТАПбд-31

Вершинин Д. В.

Проверила:

Лылова А. В.

Ульяновск, 2019

1. **Цель работы**

Изучение внутреннего и внешнего формата представления чисел, методов преобразования между этими представлениями. Приобретение умения программировать обработку числовой информации, массивов и структур на ассемблере.

1. **Техническое задание**

Программа вводит текст, задающий содержимое массива комплексных чисел через операторы вида «I.D,M», где I – десятичное значение индекса. D и M – десятичные значения действительной и мнимой части соответственно. Все значения лежат в диапазоне от 0 до 99. Неопределенные элементы массива заполняются нулевыми элементами обеих частей комплексного числа. Затем вводится индекс и выводится число , где d и m действительная и мнимая части комплексного числа, соответствующего введенному индексу.

1. **Ход работы**

Будем хранить комплексное число в виде двух массивов, где будут храниться вещественная и мнимая части каждого из чисел. По умолчанию заполним эти массивы нулями. Считаем строку, задающую последовательность комплексных чисел и обработаем ее. Найденное внутри строки число будем хранить внутри регистра. При нахождении символа «.» будем помещать число в переменную, хранящую индекс для записи числа, при нахождении символа «,» будем записывать число в массив, действительных частей, по индексу, хранящемуся в переменной. При нахождении пробела будем записывать число в массив мнимых частей. Если по введенному индексу в массиве уже будет лежать ненулевой элемент, он будет перезаписан более новым значением.

После этого остается только прочитать из консоли индекс и обратиться к элементу каждого из массивов по данному индексу. Если в них хранятся ненулевые значения возведем их в квадрат с помощью команды mul, суммируем и выведем на экран.

1. **Код программы на языке ассемблера**

|  |
| --- |
| .model small    .stack 100h    .data  CrLf db 0Dh, 0Ah, '$'  msgPressAnyKey db 0Dh, 0Ah, 'Press any key to exit...', '$'    BUF db 80  ENTERED db ?  STRING db 80 dup (?)        mas\_m dw 100 dup (0)  mas\_r dw 100 dup (0)  cnt dw 0  ind\_str db 13, 10, "Enter the index: $"  res\_str db 13, 10, "module = d^2 + m^2 = $"      .code    main proc  mov ax, @data  mov ds, ax    mov ah, 0Ah  lea dx, [BUF]  int 21h    mov ah, 09h  lea dx, [CrLf]  int 21h    xor cx, cx  mov cl, [ENTERED]  lea si, [STRING]  cld    mov di, 0  mov bx, 10  jcxz @@Exit  @@For:  lodsb  @@case1:  cmp al, '0'  jb @@case2  cmp al, '9'  ja @@case2    xor ah, ah  sub al, '0'  xchg ax, di  mul bx  add di, ax    cmp cx, 1  je @@ReadyNum  jmp @@Next  @@case2:    cmp al, '.'  jne @@mtk  mov [cnt], di  jmp @@Prepare    @@mtk:  cmp al, ','  jne @@mtk1  push bx  mov bx, [cnt]  mov mas\_r[bx], di  pop bx  jmp @@Prepare    @@mtk1:  cmp al, ' '  jne @@Next    @@ReadyNum:  push bx  mov bx, [cnt]  mov mas\_m[bx], di  pop bx  @@Prepare:  mov di, 0    @@Next:  loop @@For    mov ah, 09h  lea dx, [ind\_str]  int 21h    mov ah, 01h  int 21h    xor bx, bx  sub al, '0'  mov bl, al    mov ah, 09h  lea dx, [CrLf]  int 21h    mov ah, 09h  lea dx, [res\_str]  int 21h    mov ax, mas\_r[bx]  mul ax  xor cx, cx  add cx, ax  mov ax, mas\_m[bx]  mul ax  add cx, ax  mov ax, cx    call Show\_AX  mov ah, 09h  lea dx, [msgPressAnyKey]  int 21h    mov ah, 00h  int 16h    @@Exit:  mov ax, 4C00h  int 21h  main endp    Show\_AX proc  push ax  push bx  push cx  push dx  push di    mov cx, 10  xor di, di  or ax, ax  jns @@Conv  push ax  mov dx, '-'  mov ah, 2 ; ah - ôóíêöèÿ âûâîäà ñèìâîëà íà ýêðàí  int 21h  pop ax    neg ax    @@Conv:  xor dx, dx  div cx  add dl, '0'  inc di  push dx  or ax, ax  jnz @@Conv  @@Show:  pop dx  mov ah, 2  int 21h  dec di  jnz @@Show    pop di  pop dx  pop cx  pop bx  pop ax  ret  Show\_AX endp    end main |

1. **Контрольные примеры работы программы**

|  |  |
| --- | --- |
| № | Результат работы программы |
| 1 |  |
| 2 |  |
| 3 |  |