|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №5**

**«Программирование ветвлений»**

**ДИСЦИПЛИНА: «Машинно-зависимые языки программирования»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-32Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Амеличева К.А. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2021

**Цель:** практическое овладение навыками разработки программного кода на языке Ассемблер. Изучение команд условного и безусловного перехода. Исследование организации переходов.

**Задачи:** разработка простой программы, использующей операторы передачи управления и приемов программирования арифметических выражений, содержащих разветвления.

**Вариант 5**

**Часть 1**

**Задание:**

**Листинг:**

.model small

.stack 100h

.data

x db 0

a db 0

y db 0

y1 db 0

y2 db 0

.code

Start:

mov ax, @data

mov ds, ax

mov ax, 0

;проверка х на знак

cmp x, 0

JL Y\_1\_1

JMP Y\_1\_2

Y\_1\_1:

;вычисление модуля х

mov al, x

mov bl, -1

imul bl

mov y1, al

JMP Y\_1\_END

Y\_1\_2:

;вычисление разности х и а

mov al, x

sub al, a

mov y1, al

Y\_1\_END:

;проверка остатка от деления х на 3

mov al, x

mov bx, 3

idiv bl

cmp ah, 1

JE Y\_2\_1

JMP Y\_2\_2

Y\_2\_1:

;вычисление суммы а и х

mov al, x

add al, a

mov y2, al

JMP Y\_2\_END

Y\_2\_2:

;приравнивание у2 7

mov y2, 7

Y\_2\_END:

;вычисление разности у1 и у2

mov al, y1

sub al, y2

mov y, al

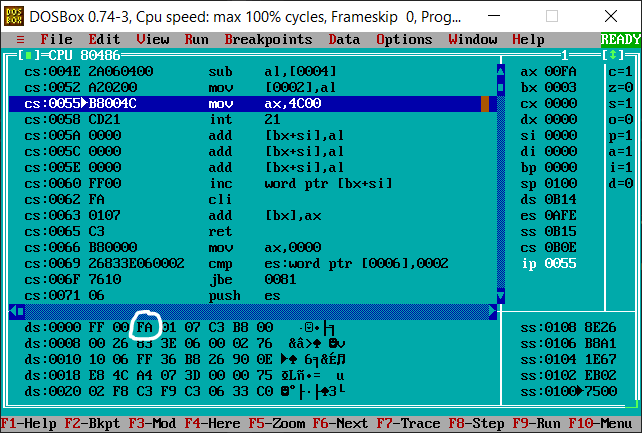
mov ax, 4C00h

int 21h

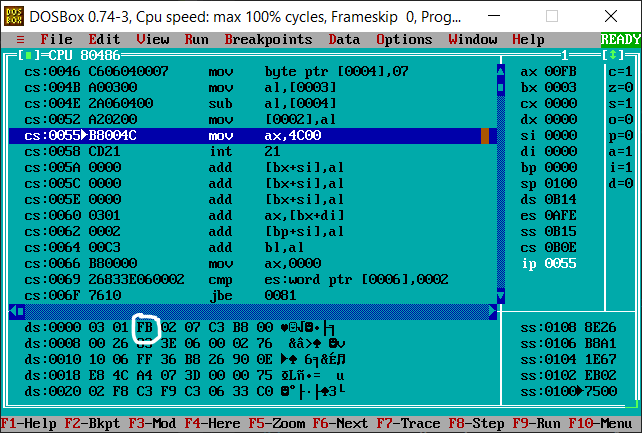
end start

END

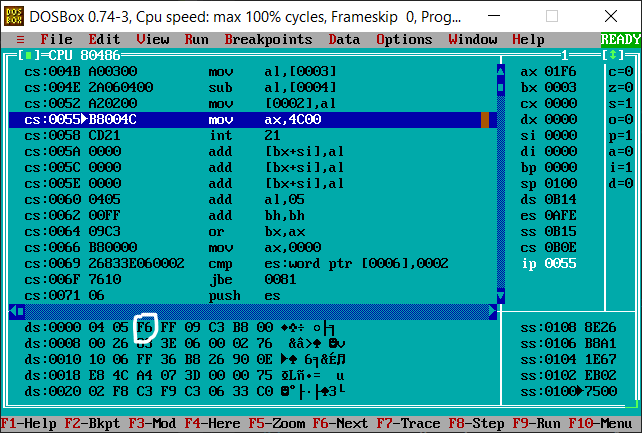
**Результаты работы:**



**Рисунок 1.** Результат (х = -1, а = 0, у = -6)



**Рисунок 2.** Результат (х = 3, а = 1, у = -5)



**Рисунок 3.** Результат (х = 4, а = 5, у = -10)

**Часть 2**

**Задание:**

1. В центре чистого экрана: сформировать меню отражающее суть задания и предлагающее пользователь ввести символ 1, 2 или 3 для расчета соответствующего выражения при заданных значениях переменных a и b.
2. Вывести по углам экрана:
   * Фамилию автора программы;
   * Номер группы;
   * Факультет;
   * Символ ‘!’, повторив его 5 раз.
3. Задать видео атрибуты, определяющие цвет символов и цвет фона при выводе символов на экран. Выбор цветового оформления зависит от вкуса.

**Листинг:**

.model small

.386

.stack 100h

.data

\_buffer db 3, 3 dup(0)

\_x db 4

\_a db 0

\_y db 0

\_y1 db 0

\_y2 db 0

\_command db 0

\_secDig db 0

\_name db 'Karelsky', '$'

\_number db 'IUK4-32.B', '$'

\_faculty db 'Informatics and control', '$'

\_symbols db '!!!!!', '$'

\_formula db ' { |x|, x < 0', 13, 10

db ' y1 = {', 13, 10

db ' { x - a, x >= 0', 13, 10, 13, 10

db ' { a + x, x mod 3 == 1', 13, 10

db ' y2 = {', 13, 10

db ' { 7, x mod 3 != 1', 13, 10, 13, 10

db ' y = y1 - y2', 13, 10, 13, 10

db ' x = 4', 13, 10

db ' a = 0', 13, 10

db ' y = 0', 13, 10, 13, 10

db ' 1: Change x', 13, 10

db ' 2: Change a', 13, 10

db ' 0: Exit', 13, 10

db ' >>> ', '$'

\_xInput db ' x = ', '$'

\_aInput db ' a = ', '$'

.code

Start:

mov ax, @data

mov ds, ax

sub ax, ax

;очищение окна

mov ax, 0600h

mov bh, 5Fh

mov cx, 0000

mov dx, 184Fh

int 10h

;отрисовка надписи в верхнем левом углу

;смещение курсора в верхний левый угол

mov ah, 2

mov dh, 0

mov dl, 0

mov bh, 0

int 10h

;вывод строки

mov ah, 09h

mov dx, offset \_name

int 21h

;отрисовка надписи в верхнем правом углу

;смещение курсора в верхний правый угол

mov ah, 2

mov dh, 0

mov dl, 47h

mov bh, 0

int 10h

;вывод строки

mov ah, 09h

mov dx, offset \_number

int 21h

;отрисовка надписи в нижнем левом углу

;смещение курсора в нижний левый угол

mov ah, 2

mov dh, 17h

mov dl, 0

mov bh, 0

int 10h

;вывод строки

mov ah, 09h

mov dx, offset \_faculty

int 21h

;отрисовка надписи в нижнем правом углу

;смещение курсора в нижний правый угол

mov ah, 2

mov dh, 17h

mov dl, 4Bh

mov bh, 0

int 10h

;вывод строки

mov ah, 09h

mov dx, offset \_symbols

int 21h

;смещение нижней строки до нижнего края

mov ax, 0701h

mov bh, 5Fh

mov cx, 1700h

mov dx, 184Fh

int 10h

;отрисовка меню

;смещение курсора

mov ah, 2

mov dh, 2

mov dl, 0

mov bh, 0

int 10h

;вывод меню

mov ah, 09h

mov dx, offset \_formula

int 21h

INPUT:

;смещение курсора к точке ввода

mov ah, 2

mov dh, 19

mov dl, 14

mov bh, 0

int 10h

;обработка нажатия клавиши

mov ah, 1

int 21h

mov \_command, al

;очищение ввода

;смещение курсора к точке ввода

mov ah, 2

mov dh, 19

mov dl, 14

mov bh, 0

int 10h

;вывод пробела в точке ввода

mov ah, 9

mov al, ' '

mov bl, 5Fh

mov cx, 1

int 10h

;обработка ввода команды меню

mov al, \_command

cmp al, '1'

je X\_INPUT

cmp al, '2'

je A\_INPUT

cmp al, '0'

je END\_PROG

JMP INPUT

X\_INPUT:

;отрисовка предложения сделать ввод числа

;смещение курсора к строке ввода числа

mov ah, 2

mov dh, 21

mov dl, 0

mov bh, 0

int 10h

;вывод строки

mov ah, 09h

mov dx, offset \_xInput

int 21h

;ввод числа

mov ah, 0Ah

lea dx, \_buffer

int 21h

;проверка числа на знак

mov al, \_buffer + 2

cmp al, '-'

;очистка предложения ввода

;смещение курсора к точке ввода числа

mov ah, 2

mov dh, 21

mov dl, 10

mov bh, 0

int 10h

;заполнение нужной части пробелами

mov ah, 9

mov al, ' '

mov bl, 5Fh

mov cx, 7

int 10h

;переходы, онованные на знаке числа

je MINUS\_X

jmp PLUS\_X

MINUS\_X:

;отрисовка символа минуса

;смещение курсора к нужной позиции меню

mov ah, 2

mov dh, 12

mov dl, 14

mov bh, 0

int 10h

;вывод минуса

mov ah, 9

mov al, '-'

mov bl, 5Fh

mov cx, 1

int 10h

;вывод числа

;смещение курсора к следующему символу

mov ah, 2

mov dh, 12

mov dl, 15

mov bh, 0

int 10h

;вывод цифры числа

mov ah, 9

mov al, \_buffer + 3

mov bl, 5Fh

mov cx, 1

int 10h

;перевод числа из символа в целочисленное, смена знака на плюс

sub al, 30h

mov bl, -1

imul bl

mov \_x, al

jmp CALCULATE

PLUS\_X:

;отрисовка символа первой цифры числа

;смещение курсора к нужной позиции

mov ah, 2

mov dh, 12

mov dl, 14

mov bh, 0

int 10h

;вывод первой цифры

mov ah, 9

mov al, \_buffer + 2

mov bl, 5Fh

mov cx, 1

int 10h

;проверка длины числа

mov al, \_buffer + 1

cmp al, 1

JE X\_ONE\_DIGIT

JMP X\_TWO\_DIGITS

X\_ONE\_DIGIT:

;очистка символа, идущего после первой цифры числа

;смещение курсора к позиции после первой цифры числа

mov ah, 2

mov dh, 12

mov dl, 15

mov bh, 0

int 10h

;вывод пробела

mov ah, 9

mov al, ' '

mov bl, 5Fh

mov cx, 1

int 10h

;перевод числа из символа в целочисленное

mov al, \_buffer + 2

sub al, 30h

mov \_x, al

jmp CALCULATE

X\_TWO\_DIGITS:

;отрисовка символа второй цифры числа

;смещение курсора к нужной позиции

mov ah, 2

mov dh, 12

mov dl, 15

mov bh, 0

int 10h

;вывод второй цифры числа

mov ah, 9

mov al, \_buffer + 3

mov bl, 5Fh

mov cx, 1

int 10h

;перевод числа из символа в целочисленное

mov al, \_buffer + 2

sub al, 30h

mov bl, 10

mul bl

add al, \_buffer + 3

sub al, 30h

mov \_x, al

jmp CALCULATE

A\_INPUT:

;отрисовка предложения сделать ввод числа

;смещение курсора к строке ввода числа

mov ah, 2

mov dh, 21

mov dl, 0

mov bh, 0

int 10h

;вывод строки

mov ah, 09h

mov dx, offset \_aInput

int 21h

;ввод числа

mov ah, 1

int 21h

;занесение числа в память

mov \_a, al

;очистка предложения ввода

;смещение курсора к точке ввода числа

mov ah, 2

mov dh, 21

mov dl, 10

mov bh, 0

int 10h

;заполнение нужной части пробелами

mov ah, 9

mov al, ' '

mov bl, 5Fh

mov cx, 7

int 10h

;отрисовка символа числа

;смещение курсора к нужной части

mov ah, 2

mov dh, 13

mov dl, 14

mov bh, 0

int 10h

;вывод числа

mov ah, 9

mov al, \_a

mov bl, 5Fh

mov cx, 1

int 10h

;перевод числа из символа в целочисленное

sub \_a, 30h

CALCULATE:

;проверка х на знак

cmp \_x, 0

JL Y\_1\_1

JMP Y\_1\_2

Y\_1\_1:

;вычисление модуля х

mov al, \_x

mov bl, -1

imul bl

mov \_y1, al

JMP Y\_1\_END

Y\_1\_2:

;вычисление разности х и а

mov al, \_x

sub al, \_a

mov \_y1, al

Y\_1\_END:

;проверка остатка от деления х на 3

mov al, \_x

mov bx, 3

mov ah, 0

idiv bl

cmp ah, 1

JE Y\_2\_1

JMP Y\_2\_2

Y\_2\_1:

;вычисление суммы а и х

mov al, \_x

add al, \_a

mov \_y2, al

JMP Y\_2\_END

Y\_2\_2:

;приравнивание у2 7

mov \_y2, 7

Y\_2\_END:

;вычисление разности у1 и у2

mov al, \_y1

sub al, \_y2

mov \_y, al

;расчистка места под вывод у

;смещение курсора к нужной позиции

mov ah, 2

mov dh, 14

mov dl, 14

mov bh, 0

int 10h

;вывод пробелов

mov ah, 9

mov al, ' '

mov bl, 5Fh

mov cx, 3

int 10h

;проверка у на знак

cmp \_y, 0

JL MINUS\_Y

PLUS\_Y:

;проверка длины у

cmp \_y, 9

JL PLUS\_Y\_ONE\_DIGIT

jmp PLUS\_Y\_TWO\_DIGITS

PLUS\_Y\_ONE\_DIGIT:

;отрисовка символа числа у

;смещение курсора к нужной позиции

mov ah, 2

mov dh, 14

mov dl, 14

mov bh, 0

int 10h

;вывод у

mov ah, 9

mov al, \_y

add al, 30h

mov bl, 5Fh

mov cx, 1

int 10h

JMP INPUT

PLUS\_Y\_TWO\_DIGITS:

;отрисовка числа у

;смещение курсора к нужной позиции

mov ah, 2

mov dh, 14

mov dl, 14

mov bh, 0

int 10h

;вывод первой цифры

mov ah, 0

mov al, \_y

mov bl, 10

div bl

mov \_secDig, ah

add al, 30h

mov bl, 5Fh

mov cx, 1

mov ah, 9

int 10h

;смещение курсора к следующему символу

mov ah, 2

mov dh, 14

mov dl, 15

mov bh, 0

int 10h

;вывод второй цифры

mov al, \_secDig

add al, 30h

mov bl, 5Fh

mov cx, 1

mov ah, 9

int 10h

JMP INPUT

MINUS\_Y:

;проверка длины у

cmp \_y, -10

;отрисовка символа минуса

;смещение курсора к нужной позиции

mov ah, 2

mov dh, 14

mov dl, 14

mov bh, 0

int 10h

;вывод минуса

mov ah, 9

mov al, '-'

mov bl, 5Fh

mov cx, 1

int 10h

;смещение курсора к следующему символу

mov ah, 2

mov dh, 14

mov dl, 15

mov bh, 0

int 10h

;смена знака у на плюс

mov al, \_y

mov bl, -1

imul bl

JG MINUS\_Y\_ONE\_DIGIT

jmp MINUS\_Y\_TWO\_DIGITS

MINUS\_Y\_ONE\_DIGIT:

;вывод числа

add al, 30h

mov bl, 5Fh

mov cx, 1

mov ah, 9

int 10h

JMP INPUT

MINUS\_Y\_TWO\_DIGITS:

;вывод числа

;вывод первой цифры

mov ah, 0

mov bl, 10

div bl

mov \_secDig, ah

add al, 30h

mov bl, 5Fh

mov cx, 1

mov ah, 9

int 10h

;смещение курсора к следующему символу

mov ah, 2

mov dh, 14

mov dl, 16

mov bh, 0

int 10h

;вывод второй цифры

mov al, \_secDig

add al, 30h

mov bl, 5Fh

mov cx, 1

mov ah, 9

int 10h

JMP INPUT

END\_PROG:

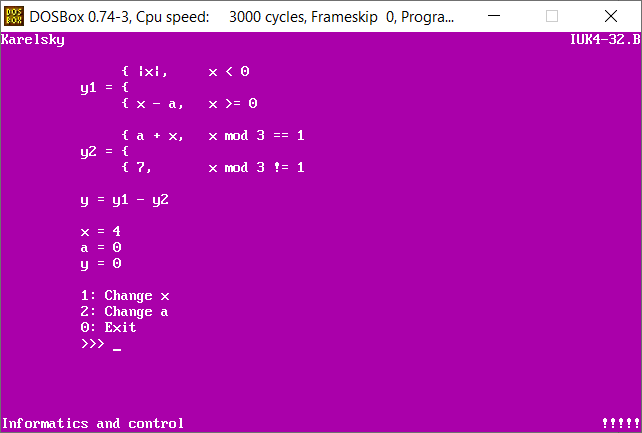
mov ax, 4C00h

int 21h

end start

END

**Результат:**



**Рисунок 4.** Результат

**Вывод:** в ходе выполнения лабораторной работы были получены навыки работы с командами безусловных и условных переходов, прерываниями BIOS.