

Лабораторная работа 4

Требуется разработать программу, вызываемую из DOS, которая за одно свое выполнение выводит на экран двоичное содержимое двух заданных регистров. Для выбора регистров используйте таблицу 5.

<i>K</i>	1	2	3	4	5	6	7	8	9	10
<i>Рег.1</i>	<i>AX</i>	<i>BX</i>	<i>CX</i>	<i>DX</i>	<i>BP</i>	<i>SI</i>	<i>DI</i>	<i>BX</i>	<i>CX</i>	<i>DX</i>
<i>Рег.2</i>	<i>BX</i>	<i>CX</i>	<i>AX</i>	<i>BX</i>	<i>AX</i>	<i>BX</i>	<i>CX</i>	<i>DX</i>	<i>BP</i>	<i>BP</i>
<i>K</i>	11	12	13	14	15	16	17	18	19	20
<i>Рег.1</i>	<i>BP</i>	<i>SI</i>	<i>DI</i>	<i>AX</i>	<i>BX</i>	<i>CX</i>	<i>DX</i>	<i>BP</i>	<i>SI</i>	<i>DI</i>
<i>Рег.2</i>	<i>DI</i>	<i>AX</i>	<i>DX</i>	<i>DX</i>	<i>DI</i>	<i>DX</i>	<i>SI</i>	<i>BX</i>	<i>CX</i>	<i>BP</i>

Примечание 1. Так как программа вызывается не из Debug, а из DOS, то она сама должна задавать первоначальное содержимое регистров, выводимых на экран. Для этого рекомендуется использовать команды mov.

Примечание 2. Если регистр, содержимое которого выводится на экран, используется в вашей программе и для других целей (например, CX – счетчик цикла), то первоначальное содержимое этого регистра необходимо переписать с 61 помощью команды mov в какой-то другой, свободный регистр данных и выполнять вывод на экран содержимого этого регистр.

```

0B0C:0103 xor cx, cx
0B0C:0105 mov bx, 2
0B0C:0108 xor dx, dx
0B0C:010A div bx
0B0C:010C push dx
0B0C:010D inc cx
0B0C:010E test ax, ax
0B0C:0110 jnz 108
0B0C:0112 mov ah, 02
0B0C:0114 pop dx
0B0C:0115 add dl, 30
0B0C:0118 int 21
0B0C:011A loop 114
0B0C:011C mov dl, 20
0B0C:011E int 21
0B0C:0120 mov ax, D892
0B0C:0123 xor cx, cx
0B0C:0125 mov bx, 2
0B0C:0128 xor dx, dx
0B0C:012A div bx
0B0C:012C push dx
0B0C:012D inc cx
0B0C:012E test ax, ax
0B0C:0130 jnz 128
0B0C:0132 mov ah

```

```

0B0C:0118 int 21
0B0C:011A loop 114
0B0C:011C mov dl, 20
0B0C:011E int 21
0B0C:0120 mov ax, D892
0B0C:0123 xor cx, cx
0B0C:0125 mov bx, 2
0B0C:0128 xor dx, dx
0B0C:012A div bx
0B0C:012C push dx
0B0C:012D inc cx
0B0C:012E test ax, ax
0B0C:0130 jnz 128
0B0C:0132 mov ah, 02
0B0C:0134 pop dx
0B0C:0135 add dl, 30
0B0C:0138 int 21
0B0C:013A loop 134
0B0C:013C mov dl, 20
0B0C:013E int 20
0B0C:0140

```

~g

```

101101101110100 1101100010010010
Программа завершилась нормально

```