

Правительство Российской Федерации
Федеральное государственное автономное образовательное учреждение
Высшего профессионального образования
Национальный исследовательский институт
«Высшая школа экономики»
Московский институт электроники и математики
Компьютерная безопасность

Отчет
по лабораторной работе №5
по курсу «Язык ассемблера»

Вариант №33

Ф.И.О. студента	Номер группы	Дата	Баллы
Николаев Александр Александрович	СКБ191	21.05.2022	

Задание А5

Задача

В массиве слов (байтов) выбранного размера (5 – 6 элементов) над каждым элементом выполнить операцию: если два старших бита и два младших бита совпадают с выбранным образцом (2 бита), то установить битовое поле 4:3. Вывод на экран исходного массива и массива результатов производить в двоичном и шестнадцатеричном представлениях.

Текст программы

```
#include <stdio.h>

void bin(int k, int size) {
    for(int i = size * 8; i > 0; --i) {
        printf("%d", (k & (1 << (i - 1))) ? 1 : 0);
    }
}

int main() {
    //char A[6] = {-57, -22, -106, -73, -127, -37};
    char sample = 3;
    short int A[6] = {-7325, -6730, -20782, -19609, -27223, -261};
    char isWord = 0;
    int size = 0;
    int volume = 0;

    if(sizeof(A[0]) == sizeof(short int)) {
        isWord = 1;
        size = sizeof(short int);
        volume = sizeof(A) / size;
    } else {
        size = sizeof(char);
        volume = sizeof(A) / size;
    }

    printf("
";
    printf("\n|   \t|\t Binary representation: \t|\t Heximal representation:\t|\n");
    printf("|_____|_____|\n");
    for (int i = 0; i < volume; ++i) {
        printf("|   %d\t|\t", i + 1);
        if (!isWord) {
            printf("\t");
        }
        bin(A[i], size);
        printf("\t\t|\t\t\tx\t\t|\n", A[i]);
    }

    printf("|_____|_____|\n\n\n");
```

<code>_asm {</code>	
<code> lea esi, A;</code>	<code>//занести адрес массива A в регистр esi</code>
<code> mov ecx, volume;</code>	<code>//занести в регистр ecx размер массива A</code>
 <code>CONDITION:</code>	
<code> cmp isWord, 1;</code>	<code>//сравнить значение флага isWord с нулём</code>
<code> je WORDBODY;</code>	<code>//если isWord == 1, то перейти к метке WORD</code>
	<code>//иначе перейти к обработке массива A как массива</code>
байтов	
<code> BYTEBODY:</code>	
<code> mov al, [esi];</code>	<code>//скопировать в регистр al значение по адресу esi</code>
для модификации	
<code> and al, 11000000b;</code>	<code>//обнулить все биты в регистре al кроме двух</code>
старших	
<code> shr al, 6;</code>	<code>//логический сдвиг вправо на 4 значения в</code>
регистре al	
 <code> mov bl, [esi];</code>	<code>//скопировать в регистр bl значение по адресу esi</code>
для модификации	
<code> and bl, 00000011b;</code>	<code>//обнулить все биты в регистре bl кроме двух</code>
младших	
 <code> mov dl, [esi];</code>	<code>//скопировать в регистр dl значение по адресу esi</code>
для модификации	
 <code> cmp al, bl;</code>	<code>//сравнить, совпадают ли значения в регистрах al</code>
и bl	
<code> jne NEXT;</code>	<code>//если не совпадают, то перейти к следующей</code>
итерации	
 <code> cmp al, sample;</code>	<code>//иначе сравнить значение регистра al с образцом</code>
sample	
<code> jne NEXT;</code>	<code>//если не совпадают, то перейти к следующей</code>
итерации	
 <code> or dl, 00011000b;</code>	<code>//иначе установить битовое поле 4:3</code>
 <code> mov [esi], dl;</code>	<code>//поместить обработанный элемент в массив A</code>
<code> inc esi;</code>	<code>//переместить указатель на следующий элемент</code>
массива A	
<code> jmp NEXT;</code>	<code>//перейти к следующей итерации</code>
 <code> WORDBODY:</code>	
<code> mov ax, [esi];</code>	<code>//скопировать в регистр ax значение по адресу esi</code>
для модификации	
<code> and ax, 1100000000000000b;</code>	<code>//обнулить все биты в регистре ax кроме двух</code>
старших	
<code> shr ax, 14;</code>	<code>//логический сдвиг вправо на 16 значения в</code>
регистре ax	
 <code> mov bx, [esi];</code>	<code>//скопировать в регистр bx значение по адресу esi</code>
для модификации	
<code> and bx, 0000000000000011b;</code>	<code>//обнулить все биты в регистре bx кроме двух</code>
младших	
 <code> mov dx, [esi];</code>	<code>//скопировать в регистр dx значение по адресу esi</code>
для модификации	

```

        cmp ax, bx;                                //сравнить, совпадают ли значения в регистрах ax
и bx
        jne NEXT;                                //если не совпадают, то перейти к следующей
итерации

        cmp bl, sample;                          //иначе сравнить значение регистра ax с образцом
sample
        jne NEXT;                                //если не совпадают, то перейти к следующей
итерации

        or dx, 0000000000011000b;                //иначе установить битовое поле 4:3 в регистре dx

        mov [esi], dx;                            //поместить обработанный элемент в массив A
        add esi, 2;                              //переместить указатель на следующий элемент
массива A
        jmp NEXT;                                //перейти к следующей итерации

NEXT:
        dec ecx;                                //уменьшить значение счётчика на 1
        cmp ecx, 0;                              //сравнить значение счётчика с нулём
        jne CONDITION;                          //если оно не равно 0, то перейти обработке
массива
        nop;

    }

    printf("
");
;
    printf("\n|   \t\t Binary representation: \t\t Heximal representation:\t|\n");

printf("|_____|_____|_____
_____| \n");
    for (int i = 0; i < volume; ++i) {
        printf("|   %d\t\t", i + 1);
        if(!isWord) {
            printf("\t");
        }
        bin(A[i], size);
        printf("\t\t|\t\t%x\t\t|\n", A[i]);
    }

printf("|_____|_____|_____
_____| \n\n\n");

    return 0;
}

```

Тестирование программы

В качестве образца выбрано число 3. Из 6 представленных однобайтовых чисел в массиве битовое поле 4:3 должно быть изменено только у чисел №1 и №6, так как 2 их старших и 2 младших бита совпадают с заданным образцом. Во второй таблице видно, что битовые поля 4:3 изменились только у чисел №1 и №6, как и ожидалось. Остальные числа не удовлетворяют условию задания, поэтому остаются неизменными.

	Binary representation:	Heximal representation:
1	11000111	ffffffc7
2	11101010	ffffffea
3	10010110	ffffff96
4	10110111	ffffffb7
5	10000001	ffffff81
6	11011011	ffffffdb

	Binary representation:	Heximal representation:
1	11011111	ffffffdf
2	11101010	ffffffea
3	10010110	ffffff96
4	10110111	ffffffb7
5	10000001	ffffff81
6	11011011	ffffffdb

В качестве образца выбрано число 3. Из 6 представленных двухбайтовых чисел в массиве битовое поле 4:3 должно быть изменено только у чисел №1 и №6, так как 2 их старших и 2 младших бита совпадают с заданным образцом. Во второй таблице видно, что битовые поля 4:3 изменились только у чисел №1 и №6, как и ожидалось. Остальные числа не удовлетворяют условию задания, поэтому остаются неизменными.

	Binary representation:	Heximal representation:
1	1110001101100011	ffffe363
2	1110010110110110	ffffe5b6
3	1010111011010010	ffffaed2
4	1011001101100111	ffffb367
5	1001010110101001	ffff95a9
6	1111111011111011	fffffefb

	Binary representation:	Heximal representation:
1	1110001101111011	ffffe37b
2	1110010110110110	ffffe5b6
3	1010111011010010	ffffaed2
4	1011001101100111	ffffb367
5	1001010110101001	ffff95a9
6	1111111011111011	fffffefb