

**Задание**

**Задание 1.** Разработайте и выполните программу, в которой реализуйте при помощи ассемблерной вставки

1 | Сложение целых чисел без знака |

После команды сложения/вычитания проанализируйте соответствующий флаг состояния процессора, и, в зависимости от его значения, выведите результат и строку «Результат верный» (если не было переполнения) или «Результат неверный» (при переполнении).

**Код**

```
#include <stdio.h>
int main(void)
{
    unsigned int x = 1914700000;
    unsigned int y = 9000000000;
    bool flag = false;

    asm ( "add %2, %0\n"
        : "=r" (x)
        : "a" (y), "d" (x) );

    asm ( "jno nooverflow;\n"
        "mov $0x1,%0;\n"
        "nooverflow : "\n"
        : "=r" (flag) : "0" (flag) );
    printf("x+y = %u\n", x);
    if (flag==true)
    {
        printf("bad\n");
    }
    else
    {
        printf("good\n");
    }
    return 0;
}
```

**Вывод**

```
x+y = 2324765408
bad
```

Получили переполнение

```
x+y = 2004700000
good
```

Без переполнения при

```
unsigned int x = 1914700000;  
unsigned int y = 90000000;
```

---

### Задание

**Задание 2.** Реализуйте задание лабораторной работы Л3 (см. раздел Л3.1, задание 1) как ассемблерную вставку в программу на C++.

1	$y(x) = 1 + x/2$
---	------------------

### Код

```
#include <iostream>  
  
using namespace std;  
//extern "C" int fun(int x);  
  
int main()  
{  
    int x = 10;  
    int y;  
    asm ( "shr %0" : "=r" (x) : "0" (x) );  
    //cout << " new x = " << x << endl;  
    asm ( "add $1,%0" : "=r" (x) : "0" (x) );  
  
    cout << " new x = " << x << endl;  
    return 0;  
}
```

### Вывод

```
new x = 6
```

---

### Задание

**Задание 3.** Реализуйте задание 2, не используя в тексте вставки конкретных имён регистров.

### Код

```

#include <iostream>

using namespace std;
//extern "C" int fun(int x);

int main()
{
    int x = 10;
    int y;
    asm ( "shr %0" : "=r" (x) : "0" (x) );
    //cout << " new x = " << x << endl;
    asm ( "add $1,%0" : "=r" (x) : "0" (x) );

    cout << " new x = " << x << endl;
    return 0;
}

```

#### Вывод

```
new x = 6
```

Второй номер сразу был решен без использования имен регистров

---

#### Задание(+2 балла)

##### Задание 4. Бонус +2 балла для пар, обязательное для троек.

Придумайте и запишите в отчёте:

- маску (т.е. второй операнд бинарной побитовой операции) для преобразования кодов ASCII [58] цифр '0'...'9' в их двоичные эквиваленты (однобайтовые числа 0...9) с помощью операции AND. При этом ASCII-код нуля (шестнадцатеричное число 0x30, что также иногда записывается как 30<sub>16</sub>, однако наиболее предпочтительное и самодокументирующееся представление ASCII-кода нуля — '0') должен быть преобразован в однобайтовое число 00, ASCII-код единицы ('1', шестнадцатеричное число 0x31 или 31<sub>16</sub>) — в 01, и т. д.;
- необходимую операцию и маску для преобразования однобайтовых чисел 0...9 в их коды ASCII;

- маску, которая преобразовывает строчные английские буквы в прописные и наоборот с помощью операции XOR.

Коды ASCII приведены в приложении Б.

Реализуйте преобразования ASCII-кодов цифр в значения и обратно как ассемблерную вставку в программу на C++.

## Код

```
#include <iostream>
using namespace std;
int main()
{
    int mask,number;
    for(int i = 0x30;i<=0x39;i++)
    {
        mask = 15;
        number = i;
        cout<<hex<<number<<" ";

        asm ("and %[mask],%[a]\n\t"
             :[a] "+r"(number) :[mask] "m"(mask) );

        cout<<hex<<number<<" ";
        mask=48;

        asm ("or %[mask],%[a]\n\t"
             :[a] "+r"(number) :[mask] "m"(mask) );
        //m-хранение в памяти, r-в регистре
        cout<<hex<<number<<endl;
    }
    for(int i = 0x41;i<=0x5a;i++)
    {
        mask = 32;
        number = i;
        cout<<(char)number<<" ";

        asm ("xor %[mask],%[a]\n\t"
             :[a] "+r"(number) :[mask] "m"(mask) );

        cout<<(char)number<<" ";
        mask=32;

        asm ("xor %[mask],%[a]\n\t"
             :[a] "+r"(number) :[mask] "m"(mask) );
        //+r - доступ для записи и чтения, =r - перезапись
        cout<<(char)number<<endl;
    }

    return 0;
}
```

## Вывод

30 0 30  
31 1 31  
32 2 32  
33 3 33  
34 4 34  
35 5 35  
36 6 36  
37 7 37  
38 8 38  
39 9 39  
A a A  
B b B  
C c C  
D d D  
E e E  
F f F  
G g G  
H h H  
I i I  
J j J  
K k K  
L l L  
M m M  
N n N  
O o O  
P p P  
Q q Q  
R r R  
S s S  
T t T  
U u U  
V v V  
W w W  
X x X  
Y y Y  
Z z Z