

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

$(N - 1) \% 6 + 1$	Вариант
1	Сложение целых чисел без знака

Код

```
Help
1.cpp x
4 > 1.cpp > ...
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      unsigned p1 = 1;
7      unsigned p2 = 2147483600;
8      asm goto (
9          "add %[p1], %[p2]\n\t"
10         "jo  %l2          \n\t"
11         ":: [p1]"r"(p1),
12         |    [p2]"r"(p2)
13         : "cc"
14         : carry
15     );
16     cout<<"Result is right"<<endl;
17     return 0;
18
19     carry: cout<<"Result is wrong"<<endl;
20     return 1;
21 }
22 |
```

Вызов

```
File Edit View Bookmarks Settings Help
alex@alex-home ~/_my/lo1/lab_ass/4  master g++ 1.cpp -m32
alex@alex-home ~/_my/lo1/lab_ass/4  master ./a.out
Result is right
alex@alex-home ~/_my/lo1/lab_ass/4  master
```

(при переполнении).

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

Код

```
2.cpp x
4 > 2.cpp > tass(int)
1  #include <iostream>
2
3  using namespace std;
4
5  int tass(int x) {
6      int y;
7
8      asm volatile (R"(
9          mov %[x],    %%eax
10         shr $1,      %%eax
11         add $1,      %%eax
12         mov %%eax,   %[y]
13     )"
14         : [y] "=r"(y)
15         : [x] "r"(x)
16         : "cc", "%eax"
17     );
18
19     return y;
20 };
21
22
23 int main() {
24     int x = 10;
25     int y = tass(x);
26     cout << y << endl;
27     return 0;
28 }
29
```

Запуск

```
alex@alex-home ~/_my/lo/ab_ass/4$ g++ 2.cpp -m32
alex@alex-home ~/_my/lo/ab_ass/4$ ./a.out
6
alex@alex-home ~/_my/lo/ab_ass/4$
```

данные 1) как ассемблерную вставку в программу на C++.

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

Задание 4. Бонус (+3 балла)

Код

```
3.cpp x
4 > 3.cpp > tass(int)
1  #include <iostream>
2
3  using namespace std;
4
5  int tass(int x) {
6      int y;
7
8      asm volatile (R"(
9          mov %[x],    %[y]
10         shr $1,      %[y]
11         add $1,      %[y]
12     )"
13         : [y]"=r"(y)
14         : [x]"r"(x)
15         : "cc"
16     );
17
18     return y;
19 };
20
21
22 int main() {
23     int x = 10;
24     int y = tass(x);
25     cout << y << endl;
26     return 0;
27 }
```

Вызов

```
alex@alex-home ~/_my/lo1/lab_ass/4 $ g++ 3.cpp -m32
alex@alex-home ~/_my/lo1/lab_ass/4 $ ./a.out
6
alex@alex-home ~/_my/lo1/lab_ass/4 $
```

Задание 4. Бонус (+2 балла).

Придумайте и запишите в отчёте:

- маску (т.е. второй операнд бинарной побитовой операции) для преобразования кодов ASCII [59] 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 30₁₆, 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 31₁₆) — 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++.

```
4 > 4.cpp > ...
1  #include <stdio.h>
2
3  // 1
4  void ASCIIToNumber(unsigned int ascii_number) {
5      unsigned int result;
6      asm ( "and $0x0F, %[a]" : "=r" (result) : [a]"r"(ascii_number));
7      printf("0x%X ASCII is %u\n", ascii_number, result);
8  }
9
10 // 2
11 void NumberToASCII(unsigned int ascii_number) {
12     unsigned int result;
13     asm ( "xor $0x30, %[a]" : "=r" (result) : [a]"r"(ascii_number));
14     printf("%u is ASCII 0x%X \n",ascii_number, result);
15 }
16
17 // 3
18 void ToLower(unsigned int symbol) {
19     unsigned int result;
20     asm ( "xor $0x20, %[a]" : "=r" (result) : [a]"r"(symbol));
21     printf("%c to lower is: %c \n", symbol, result);
22 }
23 // 4
24 void ToUpper(unsigned int symbol) {
25     unsigned int result;
26     asm ( "xor $0x20, %[a]" : "=r" (result) : [a]"r"(symbol));
27     printf("%c to upper is: %c \n", symbol, result);
28 }
29
```

```

28 }
29
30 int main() {
31     ASCIIToNumber(0x30);
32     ASCIIToNumber(0x34);
33     ASCIIToNumber(0x39);
34
35     printf("\n");
36
37     NumberToASCII(0);
38     NumberToASCII(2);
39     NumberToASCII(9);
40
41     printf("\n");
42
43     ToLower('A');
44     ToLower('O');
45     ToLower('X');
46
47     printf("\n");
48
49     ToUpper('a');
50     ToUpper('o');
51     ToUpper('x');
52     return 0;
53 }
54

```

Зануль

```

alex@alex-home ~/_my/lo1/lab_ass/4 $ g++ 4.cpp -m32
alex@alex-home ~/_my/lo1/lab_ass/4 $ ./a.out
0x30 ASCII is 0
0x34 ASCII is 4
0x39 ASCII is 9

0 is ASCII 0x30
2 is ASCII 0x32
9 is ASCII 0x39

A to lower is: a
O to lower is: o
X to lower is: x

a to upper is: A
o to upper is: O
x to upper is: X
alex@alex-home ~/_my/lo1/lab_ass/4 $

```