

Написать программу на языке C++, читающую со стандартного ввода 4 целых десятичных числа **x**, **y**, **m**, **n**, причём **m**, **n** – неотрицательные. Все смещения будут указываться начиная с младшего бита, начиная с 0.

Если в числе **x** сброшен бит **m** и установлены биты **n** и **1**, то в числе **y** сбросить бит 6 и установить самый старший бит.

Программа должна вывести все исходные данные в десятичном виде, исходные и полученные значения **x**, **y** в двоичном виде в столбик (все разряды), причём справа от числа должна быть выведена строка, комментирующая данный вывод. Пример:

```
01010101010101010101010101010101 x_in
11111111111111111100000000000000 y_in
01010101010101010101010101010101 x_out
00111111111111111100000000000111 y_out
```

В программе не должно быть более одного цикла (в функции печати). Не следует использовать классы и библиотеки для работы с битовыми объектами. Рекомендуется не изменять исходные переменные, следует определить выходные и временные переменные наиболее удобного для данного условия типа. Программа должна быть корректной вне зависимости от количества бит в целом числе на данной архитектуре.

Для защиты лабораторной работы следует подготовить набор примеров, демонстрирующих правильность её работы. Необходимо знать, как работают и для чего применяются побитовые и сдвиговые операторы C++.

Написать программу на языке C++, читающую со стандартного ввода 4 целых десятичных числа **x**, **y**, **m**, **n**, причём **m**, **n** – неотрицательные. Все смещения будут указываться начиная с младшего бита, начиная с 0.

Если в числе **x** сброшены биты **n** и **0**, и установлен бит **m**, то в числе **y** инвертировать биты 1 и 7.

Программа должна вывести все исходные данные в десятичном виде, исходные и полученные значения **x**, **y** в двоичном виде в столбик (все разряды), причём справа от числа должна быть выведена строка, комментирующая данный вывод. Пример:

```
01010101010101010101010101010101 x_in
11111111111111111100000000000000 y_in
01010101010101010101010101010101 x_out
00111111111111111100000000000111 y_out
```

В программе не должно быть более одного цикла (в функции печати). Не следует использовать классы и библиотеки для работы с битовыми объектами. Рекомендуется не изменять исходные переменные, следует определить выходные и временные переменные наиболее удобного для данного условия типа. Программа должна быть корректной вне зависимости от количества бит в целом числе на данной архитектуре.

Для защиты лабораторной работы следует подготовить набор примеров, демонстрирующих правильность её работы. Необходимо знать, как работают и для чего применяются побитовые и сдвиговые операторы C++.

Написать программу на языке C++, читающую со стандартного ввода 4 целых десятичных числа **x**, **y**, **m**, **n**, причём **m**, **n** – неотрицательные. Все смещения будут указываться начиная с младшего бита, начиная с 0.

Если в числе **x** сброшен бит **m** и установлены биты **1** и **n**, то в числе **y** установить бит **4** и сбросить самый старший бит.

Программа должна вывести все исходные данные в десятичном виде, исходные и полученные значения **x**, **y** в двоичном виде в столбик (все разряды), причём справа от числа должна быть выведена строка, комментирующая данный вывод. Пример:

```
01010101010101010101010101010101 x_in
11111111111111111100000000000000 y_in
01010101010101010101010101010101 x_out
00111111111111111100000000000111 y_out
```

В программе не должно быть более одного цикла (в функции печати). Не следует использовать классы и библиотеки для работы с битовыми объектами. Рекомендуется не изменять исходные переменные, следует определить выходные и временные переменные наиболее удобного для данного условия типа. Программа должна быть корректной вне зависимости от количества бит в целом числе на данной архитектуре.

Для защиты лабораторной работы следует подготовить набор примеров, демонстрирующих правильность её работы. Необходимо знать, как работают и для чего применяются побитовые и сдвиговые операторы C++.

Написать программу на языке C++, читающую со стандартного ввода 4 целых десятичных числа **x**, **y**, **m**, **n**, причём **m**, **n** – неотрицательные. Все смещения будут указываться начиная с младшего бита, начиная с 0.

Если в числе **x** установлены биты **m** и **n** и сброшен бит **0**, то в числе **y** установить 2 младших бита и инвертировать бит 7.

Программа должна вывести все исходные данные в десятичном виде, исходные и полученные значения **x**, **y** в двоичном виде в столбик (все разряды), причём справа от числа должна быть выведена строка, комментирующая данный вывод. Пример:

```
01010101010101010101010101010101 x_in
11111111111111111100000000000000 y_in
01010101010101010101010101010101 x_out
00111111111111111100000000000111 y_out
```

В программе не должно быть более одного цикла (в функции печати). Не следует использовать классы и библиотеки для работы с битовыми объектами. Рекомендуется не изменять исходные переменные, следует определить выходные и временные переменные наиболее удобного для данного условия типа. Программа должна быть корректной вне зависимости от количества бит в целом числе на данной архитектуре.

Для защиты лабораторной работы следует подготовить набор примеров, демонстрирующих правильность её работы. Необходимо знать, как работают и для чего применяются побитовые и сдвиговые операторы C++.

Специальный вариант

Написать программу на языке C++, читающую со стандартного ввода 4 целых десятичных числа **x**, **y**, **m**, **n**, причём **m**, **n** – неотрицательные. Все смещения будут указываться начиная с младшего бита, начиная с 0.

Если в числе **x** установлены бит **n** и **8**, и сброшен бит **m**, то в числе **y** установить 3 младших бита, сбросить бит **b**, и инвертировать два старших бита.

Программа должна вывести все исходные данные в десятичном виде, исходные и полученные значения **x**, **y** в двоичном виде в столбик (все разряды), причём справа от числа должна быть выведена строка, комментирующая данный вывод. Пример:

```
0101010101010101010101010101010101 x_in  
11111111111111111100000000000000 y_in  
01010101010101010101010101010101 x_out  
00111111111111111100000000000111 y_out
```

В программе не должно быть более одного цикла (в функции печати). Не следует использовать классы и библиотеки для работы с битовыми объектами. Рекомендуется не изменять исходные переменные, следует определить выходные и временные переменные наиболее удобного для данного условия типа. Программа должна быть корректной вне зависимости от количества бит в целом числе на данной архитектуре.

Для защиты лабораторной работы следует продготовить набор примеров, демонстрирующих правильность её работы. Необходимо знать, как работают и для чего применяются побитовые и сдвиговые операторы C++.
