

1- Write a program to perform bitwise AND operation between two numbers, 7 and 3.

اكتب برنامجًا لإجراء عملية bitwise AND بين رقمين 7 و3.

Output

```
Result of 7 & 3: 3
```

Solution

```
// www.gammal.tech
#include <iostream>

int main() {
    int result = 7 & 3; // Bitwise AND operation
    std::cout << "Result of 7 & 3: " << result << "\n";
    return 0;
}
```

2- Write a program to perform bitwise OR operation between two numbers, 5 and 3.

اكتب برنامجًا لإجراء عملية bitwise OR بين رقمين 5 و3.

Output

```
Result of 5 | 3: 7
```

Solution

```
// www.gammal.tech
#include <iostream>

int main() {
    int result = 5 | 3; // Bitwise OR operation
    std::cout << "Result of 5 | 3: " << result << "\n";
    return 0;
}
```

3- Write a program that performs the following operations:

Apply bitwise NOT operation on the decimal number 5.

Add 10 to the result obtained from the first operation.

اكتب برنامجاً يقوم بالعمليات التالية:

تطبيق عملية NOT على الرقم 5 decimal.

أضف 10 إلى النتيجة التي تم الحصول عليها من العملية الأولى.

Output

```
Result of ~5: -6
Result of (~5) + 10: 4
```

Solution

```
// www.gammal.tech
#include <iostream>

int main() {
    // Step 1: Bitwise NOT operation on 5
    int x = ~5;

    // Step 2: Add 10 to the result
    int y = x + 10;

    // Display results
    std::cout << "Result of ~5: " << x << "\n";
    std::cout << "Result of (~5) + 10: " << y << "\n";

    return 0;
}
```

4- Write a program that performs the following operations:

Apply bitwise NOT operation on the decimal number 5.

Add 10 to the result obtained from the first operation.

Left shift the result from the first operation by 2 bits.

Update the variable y based on the new value of x.

Display the results after each operation.

اكتب برنامجاً يقوم بالعمليات التالية:

تطبيق عملية NOT على الرقم العشري 5.

أضف 10 إلى النتيجة التي تم الحصول عليها من العملية الأولى.

قم بإزاحة النتيجة إلى اليسار من العملية الأولى بمقدار 2 بت.

قم بتحديث المتغير y بناءً على القيمة الجديدة لـ x.

عرض النتائج بعد كل عملية.

Output

```
Result of ~5: -6
Result of (~5) + 10: 4
Result of (~5) << 2: -24
Result of (~5) << 2 + 5: -19
```

Solution

```
// www.gammal.tech
#include <iostream>

int main() {
    // Step 1: Bitwise NOT operation on 5
    int x = ~5;
    std::cout << "Result of ~5: " << x << "\n";

    // Step 2: Add 10 to the result
    int y = x + 10;
    std::cout << "Result of (~5) + 10: " << y << "\n";

    // Step 3: Left shift x by 2 bits
    x = x << 2;
    std::cout << "Result of (~5) << 2: " << x << "\n";

    // Step 4: Update y based on the new x value
    y = x + 5;
    std::cout << "Result of (~5) << 2 + 5: " << y << "\n";

    return 0;
}
```

5- Write a program that performs the following operations:

Apply bitwise NOT operation on the decimal number 5.

Add 10 to the result obtained from the first operation.

Left shift the result from the first operation by 2 bits.

Update the variable y based on the new value of x.

Perform a bitwise AND operation with 7 on the final result.

Display the results after each operation.

اكتب برنامجاً يقوم بالعمليات التالية:

تطبيق عملية NOT على الرقم العشري 5.

أضف 10 إلى النتيجة التي تم الحصول عليها من العملية الأولى.

قم بإزاحة النتيجة إلى اليسار من العملية الأولى بمقدار 2 بت.

قم بتحديث المتغير y بناءً على القيمة الجديدة لـ x.

قم بإجراء عملية AND مع رقم 7 في النتيجة النهائية.

عرض النتائج بعد كل عملية.

Output

```
Result of ~5: -6
Result of (~5) + 10: 4
Result of (~5) << 2: -24
Result of (~5) << 2 + 5: -19
Result of ((~5) << 2 + 5) & 7: 5
```

Solution

```
// www.gammal.tech

#include <iostream>

int main() {
    // Step 1: Bitwise NOT operation on 5
    int x = ~5;
    std::cout << "Result of ~5: " << x << "\n";

    // Step 2: Add 10 to the result
    int y = x + 10;
    std::cout << "Result of (~5) + 10: " << y << "\n";

    // Step 3: Left shift x by 2 bits
    x = x << 2;
    std::cout << "Result of (~5) << 2: " << x << "\n";

    // Step 4: Update y based on the new x value
    y = x + 5;
    std::cout << "Result of (~5) << 2 + 5: " << y << "\n";

    // Step 5: Perform a bitwise AND operation with 7
    y = y & 7;
    std::cout << "Result of ((~5) << 2 + 5) & 7: " << y << "\n";

    return 0;
}
```

6- Write a program that performs the following operations:

Apply bitwise NOT operation on the decimal number 8.

Add 15 to the result obtained from the first operation.

Left shift the new result by 3 bits.

Subtract 10 from the shifted result.

Perform a bitwise AND operation with 7 on the final result.

اكتب برنامجاً يقوم بالعمليات التالية:

تطبيق عملية NOT على الرقم العشري 8.

أضف 15 إلى النتيجة التي تم الحصول عليها من العملية الأولى.

قم بإزاحة النتيجة الجديدة إلى اليسار بمقدار 3 بت.

اطرح 10 من النتيجة المتغيرة.

قم بإجراء عملية AND مع رقم 7 في النتيجة النهائية.

Output

```
Result of ~8: -9
Result of (~8) + 15: 6
Result of ((~8) + 15) << 3: 48
Result of (((~8) + 15) << 3) - 10: -4
Result of (((~8) + 15) << 3) - 10) & 7: 6
```

Solution

```
// www.gammal.tech

#include <iostream>

int main() {
    int x = ~8;    // Bitwise NOT operation on 8
    int y = x + 15; // Add 15 to the result
    y = y << 3;     // Left shift by 3 bits
    y = y - 10;    // Subtract 10
    y = y & 7;     // Bitwise AND operation with 7

    std::cout << "Result of ~8: " << x << "\n";
    std::cout << "Result of (~8) + 15: " << y << "\n";
    std::cout << "Result of ((~8) + 15) << 3: " << (y << 3) << "\n";
    std::cout << "Result of (((~8) + 15) << 3) - 10: " << (y - 10) << "\n";
    std::cout << "Result of (((~8) + 15) << 3) - 10) & 7: " << (y & 7) << "\n";

    return 0;
}
```

7- Write a program that performs the following operations:

Apply bitwise NOT operation on the decimal number 3.

Add 5 to the result obtained from the first operation.

اكتب برنامجًا يقوم بالعمليات التالية:

تطبيق عملية NOT على الرقم 3 decimal.

أضف 5 إلى النتيجة التي تم الحصول عليها من العملية الأولى.

Output

```
Result of ~3: -4  
Result of (~3) + 5: 1
```

Solution

```

// www.gammal.tech

#include <iostream>

int main() {
    // Step 1: Bitwise NOT operation on 3
    int x = ~3;
    std::cout << "Result of ~3: " << x << "\n";

    // Step 2: Add 5 to the result
    int y = x + 5;
    std::cout << "Result of (~3) + 5: " << y << "\n";

    return 0;
}
```

8- Write a program that performs the following operations:

Apply bitwise NOT operation on the decimal number 7.

Subtract 3 from the result obtained from the first operation.

اكتب برنامجًا يقوم بالعمليات التالية:

تطبيق عملية NOT على الرقم 7 decimal.

اطرح 3 من النتيجة التي تم الحصول عليها من العملية الأولى.

Output

```
Result of ~7: -8  
Result of (~7) - 3: -11
```

Solution

```
// www.gammal.tech

#include <iostream>

int main() {
    // Step 1: Bitwise NOT operation on 7
    int x = ~7;
    std::cout << "Result of ~7: " << x << "\n";

    // Step 2: Subtract 3 from the result
    int y = x - 3;
    std::cout << "Result of (~7) - 3: " << y << "\n";

    return 0;
}
```

9- Write a simple program that performs the following operations:

Apply bitwise NOT operation on the decimal number 10.
Left shift the result obtained from the first operation by 2 bits.
Add 8 to the shifted result.

اكتب برنامجاً بسيطاً يقوم بالعمليات التالية:

تطبيق عملية NOT على الرقم العشري 10.
إزاحة اليسار النتيجة التي تم الحصول عليها من العملية الأولى بمقدار 2 بت.
أضف 8 إلى النتيجة المتغيرة.

Output

```
Result of ~10: -11
Result of (~10) << 2: -44
Result of ((~10) << 2) + 8: -36
```


Solution

```
// www.gammal.tech

#include <iostream>

int main() {
    // Step 1: Bitwise NOT operation on 10
    int x = ~10;
    std::cout << "Result of ~10: " << x << "\n";

    // Step 2: Left shift the result by 2 bits
    x = x << 2;
    std::cout << "Result of (~10) << 2: " << x << "\n";

    // Step 3: Add 8 to the shifted result
    int y = x + 8;
    std::cout << "Result of ((~10) << 2) + 8: " << y << "\n";

    return 0;
}
```

10- Write a program that performs the following operations:

Apply bitwise NOT operation on the decimal number 4.
Right shift the result obtained from the first operation by 1 bit.
Subtract 2 from the shifted result.

اكتب برنامجاً يقوم بالعمليات التالية:


تطبيق عملية NOT على الرقم العشري 4.
قم بإزاحة النتيجة التي تم الحصول عليها من العملية الأولى بمقدار 1 بت.
اطرح 2 من النتيجة المتغيرة.

Output

```
Result of ~4: -5
Result of (~4) >> 1: -3
Result of ((~4) >> 1) - 2: -5
```

Solution

```


// www.gammal.tech

#include <iostream>

int main() {
    // Step 1: Bitwise NOT operation on 4
    int x = ~4;
    std::cout << "Result of ~4: " << x << "\n";

    // Step 2: Right shift the result by 1 bit
    x = x >> 1;
    std::cout << "Result of (~4) >> 1: " << x << "\n";

    // Step 3: Subtract 2 from the shifted result
    int y = x - 2;
    std::cout << "Result of ((~4) >> 1) - 2: " << y << "\n";

    return 0;
}
```
