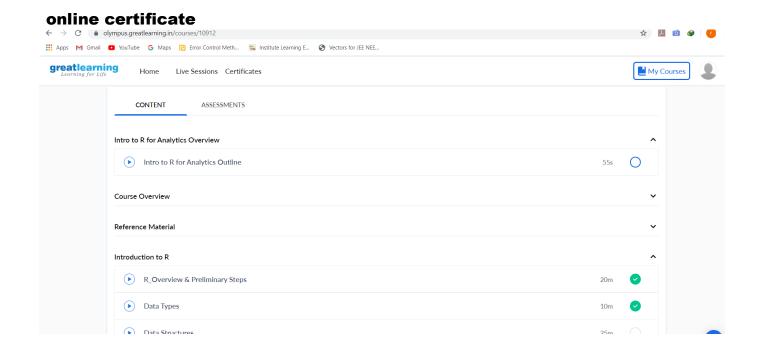
# **DAILY ONLINE ACTIVITIES SUMMARY**

Date:	08/06/2020	Name:	Raghavendra s	
Sem & Sec	8 sem B sec	USN:	4AL16CS071	
Online Test Summary				
Subject SMS				
Max. Marks	60	Score	48	
Certification Course Summary				
Course Introduction to R				
Certificate Provider greatlearning		Duration		3.00hrs
Coding Challenges				
Problem Statement:				
Status: Solved				
Uploaded the report in Github Uploaded				
If yes Repos	itory name	Raghavendra s		
Uploaded th	e report in slack	yes		

Online Test Details: (Attach the snapshot and briefly write the report for the same)

Certification Course Details: (Attach the snapshot and briefly write the report for the same)

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)



#### ONLINE CODDING

This challenge will let you learn about bitwise operators in C.

Inside the CPU, mathematical operations like addition, subtraction, multiplication and division are done in bit-level. To perform bit-level operations in C programming, bitwise operators are used which are explained below.

- Bitwise AND operator & The output of bitwise AND is 1 if the corresponding bits of two operands is 1. If either bit of an operand is 0, the result of corresponding bit is evaluated to 0. It is denoted by &.
- Bitwise OR operator | The output of bitwise OR is 1 if at least one corresponding bit of two operands is 1. It is
  denoted by |.
- Bitwise XOR (exclusive OR) operator ^ The result of bitwise XOR operator is 1 if the corresponding bits of two
  operands are opposite. It is denoted by .

For example, for integers 3 and 5,

```
3 = 00000011 (In Binary)
5 = 00000101 (In Binary)
```

#### Task

Given set, find:

- the maximum value of which is less than a given integer, where and (where) are two integers from set.
- the maximum value of which is less than a given integer, where and (where) are two integers from set.
- the maximum value of which is less than a given integer, where and (where) are two integers from set.

## **Input Format**

The only line contains space-separated integers, and, respectively.

### **Constraints**

### **Output Format**

- The first line of output contains the maximum possible value of .
- The second line of output contains the maximum possible value of .
- The second line of output contains the maximum possible value of .

#### Sample Input 0

5 4

## Sample Output 0

2

3

This challenge will let you learn about bitwise operators in C.

Inside the CPU, mathematical operations like addition, subtraction, multiplication and division are done in bit-level. To perform bit-level operations in C programming, bitwise operators are used which are explained below.

- Bitwise AND operator & The output of bitwise AND is 1 if the corresponding bits of two operands is 1. If either bit of an operand is 0, the result of corresponding bit is evaluated to 0. It is denoted by &.
- Bitwise OR operator | The output of bitwise OR is 1 if at least one corresponding bit of two operands is 1. It is
  denoted by |.
- Bitwise XOR (exclusive OR) operator ^ The result of bitwise XOR operator is 1 if the corresponding bits of two operands are opposite. It is denoted by .

For example, for integers 3 and 5,

```
3 = 00000011 (In Binary)
5 = 00000101 (In Binary)

AND operation OR operation XOR operation
00000011 00000011 00000011
& 00000101 | 00000101 ^ 00000101

00000001 = 1 00000111 = 7 00000110 = 6

Task
```

Given set, find:

- the maximum value of which is less than a given integer, where and (where) are two integers from set.
- the maximum value of which is less than a given integer, where and (where) are two integers from set.
- the maximum value of which is less than a given integer, where and (where ) are two integers from set.

# **Input Format**

The only line contains space-separated integers, and, respectively.

#### **Constraints**

•

•

# **Output Format**

- The first line of output contains the maximum possible value of .
- The second line of output contains the maximum possible value of .
- The second line of output contains the maximum possible value of .

### Sample Input 0

```
5 4
Sample Output 0
2
3
3
```

# **Program**

```
void calculate_the_maximum(int n, int k) {
  int maxAnd = 0;
  int maxOr = 0;
  int maxXor = 0;
  for (int i=1; i<=n; i++) {</pre>
```

```
for (int j=i+1; j<=n; j++) {
    if (((i&j) > maxAnd) && ((i&j) < k)) {
        maxAnd = i&j;
    }
    if (((i|j) > maxOr) && ((i|j) < k)) {
        maxOr = i|j;
    }
    if (((i^j) > maxXor) && ((i^j) < k)) {
        maxXor = i^j;
    }
    }
}
printf("%d\n%d\n%d\n", maxAnd, maxOr, maxXor);
}</pre>
```