Lecture 7 : Operators & For Loop

Bitwise Output

Send Feedback

Choose the correct output for the following code:

```
public static void main (String[] args) {
   int a=10,b=50;
   a++;
   --b;
   int c=a--+b--;
   System.out.print(++c);
}
```

Options

This problem has only one correct answer

- ○ 60
- ◉ 61
- ○ 62
- ○ error

Hurray! Correct Answer

Solution Description

a++,a=11. --b,b=49.
c=a--+b--=11+49=60
a=10
b=48
print(++c)->print(61)

Find error

Send Feedback

Which line(s) of the following code would give an error:

```
public static void main (String[] args) {
   int a=10,b=20;
   System.out.println(a+++--b);//line 1
   System.out.println(a--+++b);//line 2
   System.out.println(a++-++b);//line 3
   System.out.println(a+++++b);//line 4
}
```

Options

This problem may have one or more correct answers

- ☐ line 1
- ☑ line 2 ✔
- ☐ line 3
- ☑ line 4 ✔

Hurray! Correct Answer

Solution Description

As + operator and ++ operator have equal precedence.
Similarly - and -- have equal precedence.
So in line 2 +++b and in line 4 +++b gives an error.
To solve this error we can use +(++b).

Bitwise operator

Send Feedback

Choose the correct output of the following code:

```
public static void main (String[] args) {
   int a=10,b=20;
   int c=a&b;
   System.out.print(c);
   int d=a|b;
   System.out.print(d);
   int e=a^b;
   System.out.print(e);
   int f=c+d+e;
   System.out.print(~f);
}
```

Options

This problem has only one correct answer

- 03030-60
- 03030-61
- 0303060
- 03030-59

Hurray! Correct Answer

Solution Description

c=10&20=0.
d=10|20=30.
e=10^20=30.
f=0+30+30=60.
~f=-61.

Correct Output

Send Feedback

Choose the correct output for the given code:

```
public static void main (String[] args) {
   int a=10,b=-20;
   System.out.print(a^b);
}
```

Options

This problem has only one correct answer

- (●) -26
- ( ) 26
- ( ) 35
- ( ) -35

Hurray! Correct Answer

Bitwise Operator

Send Feedback

What will be the output ?

```
public static void main(String args[])
  {
     int a = 42;
     int b = ~a; (there is tilde sign before a)
     System.out.print(a + " " + b);
  }
```

Options

This problem has only one correct answer

- ( ) 42 42
- (●) 42 -43
- ( ) 42 43
- ( ) 43 43

Hurray! Correct Answer

Guess the output

Send Feedback

```
class Output {
   public static void main(String args[])
   {
      int x , y = 1;
      x = 10;
      if (x != 10 && x / 0 == 0)
         System.out.println(y);
      else
         System.out.println(++y);
   }
}
```

Options

This problem has only one correct answer

- (•) 2
- ( ) 1
- ( ) Runtime error because of division by zero in if condition
- ( ) None of these

Hurray! Correct Answer

Guess the output

Send Feedback

```
class Solution {
   public static void main(String args[])
   {
   int x = 15;
   int y = x++;
   int  z = ++x;
   System.out.println(y +" " + z);
   }
}
```

Options

This problem has only one correct answer

- ( ) 15 16
- ( ) 16 17
- (•) 15 17
- ( ) 16 16

Hurray! Correct Answer

Guess the output

Send Feedback

```
class Solution {
   public static void main(String args[])
   {
      int g = 3;
      System.out.print(++g * 8);
   }
}
```

Options

This problem has only one correct answer

- ( ) 24
- (•) 32

Hurray! Correct Answer

Guess the output

Send Feedback

```
class Solution {
   public static void main(String args[])
   {
      int x =10;
      int y = 20;
      if(x++ > 10 && ++y > 20 ){
      System.out.print("Inside if  ");
      }else{
      System.out.print("Inside else  ");
      }
      System.out.println(x +" "+y);
    }
}
```

Options

This problem has only one correct answer

- ○ Inside if 11 21
- ○ Inside if 10 21
- ◉ Inside else 11 20
- ○ Inside else 11 21

✓ Hurray! Correct Answer

Guess the output

Send Feedback

```
class Solution {
   public static void main(String args[])
   {
       int x = 10;
       int y = 20;
       if(x++ > 10 || ++y > 20 ){
      System.out.print("Inside if  ");
       }else{
      System.out.print("Inside else  ");
       }
       System.out.println(x  + " " + y);
   }
}
```

Options

This problem has only one correct answer

- ◉ Inside if 11 21
- ○ Inside if 10 21
- ○ Inside else 11 20
- ○ Inside else 11 21

✓ Hurray! Correct Answer

Correct Output

Send Feedback

Select the correct output for the following code:

```
public static void main (String[] args) {
   int a=5;
   a+=5+(++a)+(a++);
   System.out.print(a);
}
```

Options

This problem has only one correct answer

- 20
- 22 (selected)
- 17
- 18

Hurray! Correct Answer

Solution Description

a=a+5+(++a)+(a++).
=5+5+6+6=22

Guess the Output

Send Feedback

```
public static void main (String[] args) {
   int a=10;
   a+=++a-5/3+6*a;
   System.out.print(a);
}
```

Options

This problem has only one correct answer

- 50
- 70
- 76
- 86 (selected)

Hurray! Correct Answer

Solution Description

a+=++a-5/3+6*a.
a+=11-1+6*11.
a=10+11-1+66=86.

Nth Fibonacci Number

Send Feedback

Nth term of Fibonacci series F(n), where F(n) is a function, is calculated using the following formula -

F(n) = F(n-1) + F(n-2),
Where, F(1) = 1,
F(2) = 1

Provided N you have to find out the Nth Fibonacci Number.

Input Format :

The first line of each test case contains a real number ‘N’.

Output Format :

For each test case, return its equivalent Fibonacci number.

Constraints:

1 <= N <= 10000

Where ‘N’ represents the number for which we have to find its equivalent Fibonacci number.

Time Limit: 1 second

Sample Input 1:

6

Sample Output 1:

8

Explanation of Sample Input 1:

Now the number is ‘6’ so we have to find the “6th” Fibonacci number

So by using the property of the Fibonacci series i.e

[1, 1, 2, 3, 5, 8, 13, 21]

So the “6th” element is “8” hence we get the output.

```java
import java.util.Scanner;

public class Solution {
   public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int fib = 0;
        int a = 0, b = 1;

        if (n == 1) {
            System.out.println(1);
            System.exit(0);
        }

        for (int i = 2; i <= n; i++) {
            fib = a + b;
            a = b;
            b = fib;
        }
        System.out.println(fib);
   }
}
```

For Loop

Send Feedback

How many times will the following loop run?

```
for(int i=1;i<10;i*=2)
{
    System.out.println(i);
}
```

Options

This problem has only one correct answer

- (•) 4
- ( ) 3
- ( ) 10
- ( ) infinite

Hurray! Correct Answer

Solution Description

The loop would run for i=1,2,4,8.

Guess the Output

Send Feedback

Which value(s) can be used to initialize i so that the loop is finite:

```
public static void main (String[] args) {
   for(int i=_;i>0;i=i%3)
   {
      System.out.print("*");
   }
}
```

Options

This problem may have one or more correct answers

- [ ] 1000
- [x] 729 ✓
- [ ] 386
- [x] 483 ✓

Hurray! Correct Answer

Solution Description

The value which is divisible by 3 would result in finite loop.

What is the output

Send Feedback

What will be the output the following code?

```
for(int i = 0; i < 5; i = i + 1){
   System.out.print(i + " ");
   i = i + 1;
}
```

Options

This problem has only one correct answer

- ( ) 0 1 2 3 4
- (•) 0 2 4
- ( ) 1 3
- ( ) 1 2 3 4 5

Hurray! Correct Answer

What is the output

Send Feedback

What will be the output the following code?

```
for(int i = 1; i < 5; i = i + 1){
   System.out.print(i +" ");
   i = i - 1;
}
```

Options

This problem has only one correct answer

- 1 2 3 4
- Infinite 1s
- Compilation error
- None of these

Hurray! Correct Answer

What is the output

Send Feedback

What will be the output ?

```
for(int i = 0; i < 2; i = i + 1) {
   for(int j = 0; j < 2; j = j + 1) {
      if (j == 1)
         break;
      System.out.print(j +" ");
   }
}
```

Options

This problem has only one correct answer

- 0 1 0 1
- 0 0 0 0
- 0 0
- 0 1

Hurray! Correct Answer

Guess the Output

Send Feedback

Guess the output for the following code:

```
public static void main (String[] args) {
   int i=0;
   for(;;)
   {
      if(i==5)
         break;
      System.out.print(i);
      i++;
   }
}
```

Options

This problem has only one correct answer

- 012345
- infinite loop
- 01234
- compilation error

Hurray! Correct Answer

Solution Description

The loop would execute for i=0,1,2,3,4

Correct Output

Send Feedback

Select the correct output for the following code:

```
public static void main (String[] args) {
   for(int i=7;i!=0;i--)
   {
      System.out.print(i--);
   }
}
```

Options

This problem has only one correct answer

- 7531
- 642
- infinite loop
- 753

Hurray! Correct Answer

Find the error

Send Feedback

Which code snippet would generate an error?

1.

```
for(int i=1;;i++)
   {  if(i==5)
         break;
      System.out.print(i);
   }
```

2.

```
for(int i=1;;i++)
   {  if(i<5)
         System.out.print(i);
      else
         break;
   }
```

3.

```
for(int i=1;;i++)
   {  if(i>5)
      {
         break;
         System.out.print("break statement reached");
      }
      System.out.print(i);
   }
```

Options

This problem has only one correct answer

- 1.
- 2.
- 3.
- None of them

Hurray! Correct Answer

Solution Description

In code snippet 3. the statement after break is never executed hence it gives an error.

Skip iteration

Send Feedback

Which of these jump statements can skip processing remainder of code in its body for a particular iteration ?

Options

Attempts

This problem has only one correct answer

- ○ break
- ○ return
- ◉ continue

Hurray! Correct Answer

Solution Description

####"break" is used to exit from the current loop.
####"return" statement is used to exit from the current function.
####"continue" is used to skip the current iteration of a loop and continue with the next iteration.

What is the output

Send Feedback

```
int i = 1;
while(i < 5) {
   if(i == 3) {
      break;
   }
   System.out.print(i + " ");
   i++;
}
```

Options

This problem has only one correct answer

- ○ 1 2 3 4
- ◉ 1 2
- ○ 1 2 4
- ○ [No output, just an infinite loop]

Hurray! Correct Answer

What is the output

Send Feedback

```
int i = 1;
while(i < 5) {
   if(i == 3) {
      continue;
   }
   System.out.print(i + " ");
   i++;
}
```

Options

This problem has only one correct answer

- ○ 1 2 3 4
- ○ 1 2 [Without infinite loop]
- ○ 1 2 4
- ◉ 1 2 [With infinite loop]

Hurray! Correct Answer

What is the output

Send Feedback

```
int i = 1;
while(i < 3) {
   int j = 1;
   while(j < 5) {
      if(j == 3) {
         break;
      }
      System.out.print(j + " ");
      j++;
   }
   i++;
}
```

Options

This problem has only one correct answer

- (●) 1 2 1 2
- ( ) 1 2
- ( ) 1 2 4 1 2 4
- ( ) Infinite loop

Hurray! Correct Answer

What is the output

Send Feedback

```
int i = 1;
while(i < 3) {
   int j = 0;
   while(j < 5) {
      j++;
      if(j == 3) {
         continue;
      }
      System.out.print(j + " ");
   }
   i++;
}
```

Options

This problem has only one correct answer

- ( ) 1 2 1 2
- ( ) 1 2 3 4 1 2 3 4
- (●) 1 2 4 5 1 2 4 5
- ( ) 1 2 4 1 2 4

Hurray! Correct Answer

All Prime Numbers

Send Feedback

Given an integer N, print all the prime numbers that lie in the range 2 to N (both inclusive).

Print the prime numbers in different lines.

Input Format :
Integer N
Output Format :
Prime numbers in different lines
Constraints :
1 <= N <= 100
Sample Input 1:

9

Sample Output 1:

2
3
5
7

Sample Input 2:

20

Sample Output 2:

2
3
5
7
11
13
17
19

```java
import java.util.Scanner;
public class Solution {
   public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        boolean flag;
        for (int i = 2; i <= n; i++) {
            flag = true;
            for (int j = 2; j <= i / 2; j++) {
                if (i == 2)
                    continue;
                if (i % j == 0) {
                    flag = false;
                    break;
                }
            }
            if (flag)
                System.out.println(i);
        }
    }
}
```

Sum or Product

Send Feedback

Write a program that asks the user for a number N and a choice C. And then give them the possibility to choose between computing the sum and computing the product of all integers in the range 1 to N (both inclusive).

If C is equal to -

1, then print the sum
2, then print the product
Any other number, then print '-1' (without the quotes)

Input format :

Line 1 : Integer N
Line 2 : Choice C

Output Format :

Sum or product according to user's choice

Constraints :

1 <= N <= 12

Sample Input 1 :

10
1

Sample Output 1 :

55

Sample Input 2 :

10
2

Sample Output 2 :

3628800

Sample Input 3 :

10
4

Sample Output 3 :

-1

```
import java.util.Scanner;
public class Main {
   public static void main(String[] args) {
       // Write your code here
       Scanner sc = new Scanner(System.in);
       int n = sc.nextInt();
       int c = sc.nextInt();
       int sum = 0, product = 1;

       if (c == 1) {
           for (int i = 0; i <= n; i++)
               sum += i;
           System.out.println(sum);
       }

       else if (c == 2) {
           for (int i = 1; i <= n; i++)
               product *= i;
```

```
            System.out.println(product);
        }
        else
            System.out.println("-1");
    }
}
```

Terms of AP

Send Feedback

Write a program to print first x terms of the series 3N + 2 which are not multiples of 4.

Input format :

Integer x

Output format :

Terms of series (separated by space)

Constraints :

1 <= x <= 1,000

Sample Input 1 :

10

Sample Output 1 :

5 11 14 17 23 26 29 35 38 41

Sample Input 2 :

4

Sample Output 2 :

5 11 14 17

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        // Write your code here
        Scanner sc = new Scanner(System.in);
        int x = sc.nextInt();
        int ap;

        for (int i = 1, j = 1; i <= x; j++) {
            if ((ap = 3 * j + 2) % 4 == 0)
                continue;
            System.out.print(ap + " ");
            i++;
        }
    }
}
```

Reverse of a number

Send Feedback

Write a program to generate the reverse of a given number N. Print the corresponding reverse number.

Note : If a number has trailing zeros, then its reverse will not include them. For e.g., reverse of 10400 will be 401 instead of 00401.

Input format :

Integer N

Output format :

Corresponding reverse number

Constraints:

0 <= N < 10^8

Sample Input 1 :

1234

Sample Output 1 :

4321

Sample Input 2 :

1980

Sample Output 2 :

891

```
import java.util.Scanner;
public class Main {

   public static void main(String[] args) {
       // Write your code here
       Scanner sc = new Scanner(System.in);
       int num = sc.nextInt();
       int rev_num = 0;

       while(num>0){
           rev_num = rev_num*10 + num%10;
           num /=10;
       }
       System.out.println(rev_num);
   }
}
```

Binary to decimal

Send Feedback

Given a binary number as an integer N, convert it into decimal and print.

Input format :

An integer N in the Binary Format

```
Output format :
Corresponding Decimal number (as integer)
Constraints :
0 <= N <= 10^9
Sample Input 1 :
1100
Sample Output 1 :
12
Sample Input 2 :
111
Sample Output 2 :
7
```

```java
import java.util.Scanner;
import java.lang.Math;
public class Main {
   public static void main(String[] args) {
       // Write your code here
       Scanner sc = new Scanner(System.in);
       int n = sc.nextInt();
       int bnum = 0;

       for(int i=0; n>0; i++){
           bnum += n%10*Math.pow(2,i);
           n=n/10;
       }
       System.out.print(bnum);
   }
}
```

Decimal to Binary

Send Feedback

Given a decimal number (integer N), convert it into binary and print.
The binary number should be in the form of an integer.
Note: The given input number could be large, so the corresponding binary number can exceed the integer range. So you may want to take the answer as long for CPP and Java.

Note for C++ coders: Do not use the inbuilt implementation of "pow" function. The implementation of that function is done for 'double's and it may fail when used for 'int's, 'long's, or 'long long's. Implement your own "pow" function to work for non-float data types.

Input format :

Integer N

Output format :

Corresponding Binary number (*long*)

Constraints :

0 <= *N* <= 10^5

Sample Input 1 :

12

Sample Output 1 :

1100

Sample Input 2 :

7

Sample Output 2 :

111

```
import java.util.Scanner;
public class Main {
   public static void main(String[] args) {
       // Write your code here
       Scanner sc = new Scanner(System.in);
       int dnum = sc.nextInt();
       long bnum = 0;

       for (int i = 0; dnum > 0; i++) {
           bnum += dnum % 2 * (long) Math.pow(10, i);
           dnum = dnum / 2;
       }
       System.out.print(bnum);
   }
}
```

Square Root (Integral)

Send Feedback

Given a number N, find its square root. You need to find and print only the integral part of square root of N.

For eg. if number given is 18, answer is 4.

Input format :

Integer N

Output Format :

Square root of N (integer part only)

Constraints :

0 <= *N* <= 10^8

Sample Input 1 :

10

```
Sample Output 1 :
3
Sample Input 2 :
4
Sample Output 2 :
2


import java.util.Scanner;
public class Main {
   public static void main(String[] args) {
       // Write your code here
       Scanner sc = new Scanner(System.in);
       int n = sc.nextInt();
       int root = 0;
       if (n == 1 || n == 0) {
           System.out.println(n);
           System.exit(0);
       }
       for (int i = 2; i * i <= n; i++) {
           if (i * i <= n) {
               root = i;
           }
       }
       System.out.println(root);
   }
}
```

Check Number sequence

You are given S, a sequence of n integers i.e. S = s1, s2, ..., sn. Compute if it is possible to split S into two parts : s1, s2, ..., si and si+1, si+2, …..., sn (0 <= i <= n) in such a way that the first part is strictly decreasing while the second is strictly increasing one.
Note : We say that x is strictly larger than y when x > y.
So, a strictly increasing sequence can be 1 4 8. However, 1 4 4 is NOT a strictly increasing sequence.

That is, in the sequence if numbers are decreasing, they can start increasing at one point. And once the sequence of numbers starts increasing, they cannot decrease at any point further.
Sequence made up of only increasing numbers or only decreasing numbers is a valid sequence. So, in both the cases, print true.

You just need to print true/false. No need to split the sequence.
Input format :
Line 1 : Integer 'n'
Line 2 and Onwards : 'n' integers on 'n' lines(single integer on each line)

Output Format :
"true" or "false" (without quotes)
Constraints :
1 <= n <= 10^7
Sample Input 1 :
5
9
8
4
5
6
Sample Output 1 :
true
Sample Input 2 :
3
1
2
3
Sample Output 2 :
true
Sample Input 3 :
3
8
7
7
Sample Output 3 :
false
Explanation for Sample Format 3 :
8 7 7 is not strictly decreasing, so output is false.
Sample Input 4 :
6
8
7
6
5
8
2
Sample Output 4 :
false
Explanation for Sample Input 4 :
The series is :
8 7 6 5 8 2
It is strictly decreasing first (8 7 6 5). Then it's strictly increasing (5 8). But then it starts strictly decreasing again (8 2). Therefore, the output for this test case is 'false'

```
import java.util.Scanner;

public class Main {
   public static void main(String[] args) {
       // Write your code here
       Scanner sc = new Scanner(System.in);
       int n = sc.nextInt();
       int num, prevnum = 0, count = 0;
```

```
        boolean ascending = false, decending = false, sequence = true;

        for (int i = 0; i < n; i++) {
            num = sc.nextInt();

            if (num > prevnum && !ascending) {
                ascending = true;
                decending = false;
                count++;
            }
            if (num < prevnum && !decending) {
                ascending = false;
                decending = true;
                count++;
            }
            if (count > 3 || num == prevnum) {
                sequence = false;
            }
            prevnum = num;
        }
        System.out.println(sequence);
    }
}
```