1. WAP to sorting string without using string Methods?.
2. WAP to show object cloning in java using cloneable and copy constructor both.
3. Convert the following code so that it uses nested while statements instead of for statements:

```
int s = 0;
int t = 1;
for (int i = 0; i < 10; i++)
{
s = s + i;
for (int j = i; j > 0; j−−)
{
t = t * (j - i);
}
s = s * t;
System.out.println("T is " + t);
}
System.out.println("S is " + s);
```

4.What will be the output on new Child(); ?

```
class Parent extends Grandparent {

    {
    System.out.println("instance - parent");
    }

    public Parent() {
    System.out.println("constructor - parent");
    }

    static {
    System.out.println("static - parent");
    }
}

class Grandparent {

    static {
    System.out.println("static - grandparent");
    }

    {
    System.out.println("instance - grandparent");
    }

    public Grandparent() {
    System.out.println("constructor -
grandparent");
```

```
    }
}

class Child extends Parent {

    public Child() {
    System.out.println("constructor - child");
    }

    static {
    System.out.println("static - child");
    }

    {
    System.out.println("instance - child");
    }
}
```

4. In this challenge, you must read an integer, a double, and a String from stdin, then print the values according to the instructions in the Output Format section below. To make the problem a little easier, a portion of the code is provided for you in the editor.
Note: We recommend completing Java Stdin and Stdout I before attempting this challenge.
Input Format
There are three lines of input:
1. The first line contains an integer.
2. The second line contains a double.
3. The third line contains a String.
Output Format
There are three lines of output:
1. On the first line, print String: followed by the unaltered String read from stdin.
2. On the second line, print Double: followed by the unaltered double read from stdin.
3. On the third line, print Int: followed by the unaltered integer read from stdin.
To make the problem easier, a portion of the code is already provided in the editor.
Note: If you use the nextLine() method immediately following the nextInt() method, recall that nextInt() reads integer tokens; because of this, the last newline character for that line of integer input is still queued in the input buffer and the next nextLine() will be reading the remainder of the integer line (which is empty).
Sample Input
42
3.1415
Welcome to Java tutorials!
Sample Output
String: Welcome to Java tutorials!
Double: 3.1415
Int: 42

5. Java's System.out.printf function can be used to print formatted output. The purpose of this exercise is to test your understanding of formatting output using printf.

To get you started, a portion of the solution is provided for you in the editor; you must format and print the input to complete the solution.
Input Format
Every line of input will contain a String followed by an integer.
Each String will have a maximum of 10 alphabetic characters, and each integer will be in the inclusive range from 0 to 999.
Output Format
In each line of output there should be two columns:
The first column contains the String and is left justified using exactly 15 characters.
The second column contains the integer, expressed in exactly 3 digits; if the original input has less than three digits, you must pad your output's leading digits with zeroes.
Sample Input
java 100
cpp 65
python 50
Sample Output
```
==============================
java           100
cpp            065
python          050
==============================
```
Explanation
Each String is left-justified with trailing whitespace through the first
15 characters. The leading digit of the integer is the 16th character, and each integer that was less than 3 digits now has leading zeroes.

6.Given a string, s , and two indices, start and end , print a substring consisting of all characters in the inclusive range from start to end -1 . You'll find the String class' substring method helpful in completing this challenge.

Input Format

The first line contains a single string denoting .

The second line contains two space-separated integers denoting the respective values of start and end.

Constraints

- 1 <= |s| <= 100
- 0 <= start < end <=n
- String s consists of English alphabetic letters (i.e.,[a-z A - Z]) only.

Output Format

Print the substring in the inclusive range from start to end -1 .

Sample Input

Helloworld

3 7

Sample Output

lowo

Explanation

In the diagram below, the substring is highlighted in green: