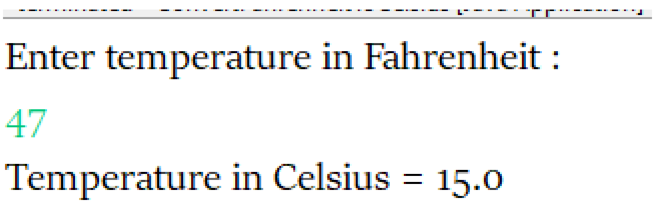
Que 1. WAP to convert Fahrenheit to Celsius in Java using formula given below

°C = (°F – 32) / (9/5)

```
package Lab_Assignment;
import java.util.Scanner;
public class ConvertFahrenheitToCelsius {
        public static void main(String[] args) {
                float fahrenheit;
                float temperature;
                Scanner sc = new Scanner(System.in)  ;
                System.out.println("Enter temperature in Fahrenheit : ");
                fahrenheit = sc.nextFloat();
                temperature = (fahrenheit - 32)/(9/5);
                System.out.println("Temperature in Celsius = "+temperature);
  }
}
```

OUTPUT :



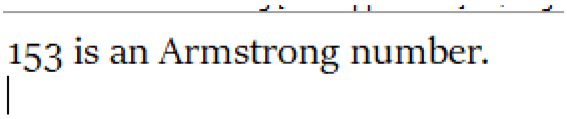
```
Enter temperature in Fahrenheit :
47
Temperature in Celsius = 15.0
```

Que 2. WAP to check a given number is Armstrong or not i.e. (153 = 1*1*1 + 5*5*5+3*3*3)

```
package Lab_Assignment;
public class Armstrong {
        public static void main(String[] args) {
                //First, given number (number)'s value is stored in another integer
variable, originalNumber.
                //This is because, we need to compare the values of final number and
original number at the end.
                int number = 153;
                int originalNumber, remainder;
                int result = 0;
                originalNumber = number;
                //Then, a while loop is used to loop through originalNumber until it is
equal to 0
```

```
                while (originalNumber != 0)
                {
                        //On each iteration, the last digit of num is stored in remainder.
                        remainder = originalNumber % 10;
                        //Then, remainder is powered by 3 (number of digits) using
Math.pow() function and added to result.
//      Then, the last digit is removed from originalNumber after division by 10.
          result += Math.pow(remainder, 3);
          //Then, the last digit is removed from originalNumber after division by 10.
          originalNumber /= 10;
        }
                //Finally, result and number are compared. If equal, it is an Armstrong
number. If not, it isn't.
        if(result == number)
          System.out.println(number + " is an Armstrong number.");
        else
          System.out.println(number + " is not an Armstrong number.");
         }
}
```

OUTPUT :



```
153 is an Armstrong number.
```

Que 3. Rajan went to a movie with his friends in a multiplex theatre and during break time he bought pizzas, puffs and cool drinks. Consider the following prices :

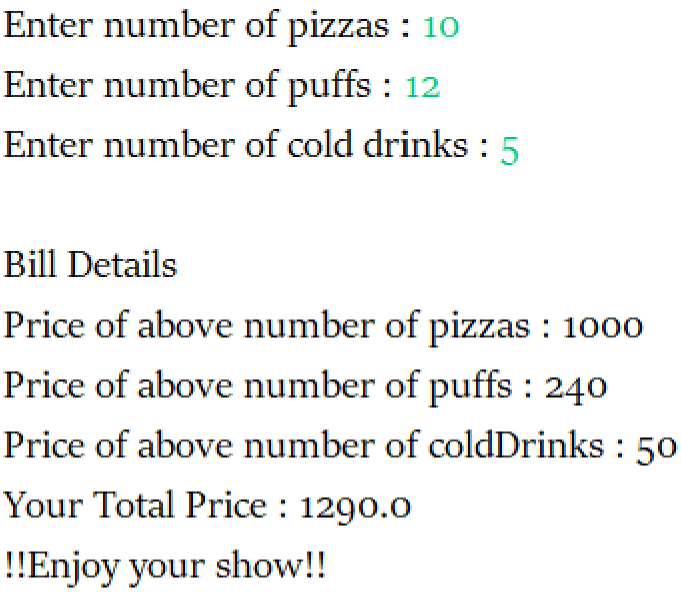
Rs.100/pizza
Rs.20/puffs
Rs.10/cooldrink

Generate a bill for What Rajan has bought.

```
package Lab_Assignment;

import java.util.Scanner;
public class Bill {
        public static void main(String[] args) {
          float totalPrice;
          Scanner sc = new Scanner(System.in);
          System.out.print("Enter number of pizzas : ");
```

```
            int pizza = sc.nextInt();
            System.out.print("Enter number of puffs : ");
            int puffs = sc.nextInt();
            System.out.print("Enter number of cold drinks : ");
            int coldDrinks = sc.nextInt();
            pizza = pizza *100;
            puffs = puffs*20;
            coldDrinks = coldDrinks*10;
            System.out.println();
            System.out.println("Bill Details");
            System.out.println("Price of above number of pizzas : " +pizza);
            System.out.println("Price of above number of puffs : " +puffs);
            System.out.println("Price of above number of coldDrinks : " +coldDrinks);
            totalPrice = pizza+puffs+coldDrinks;
            System.out.println("Your Total Price : " +totalPrice);
            System.out.println("!!Enjoy your show!!");
        }
}
```

OUTPUT:



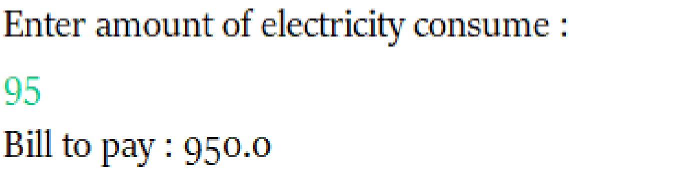
```
Enter number of pizzas : 10
Enter number of puffs : 12
Enter number of cold drinks : 5

Bill Details
Price of above number of pizzas : 1000
Price of above number of puffs : 240
Price of above number of coldDrinks : 50
Your Total Price : 1290.0
!!Enjoy your show!!
```

--

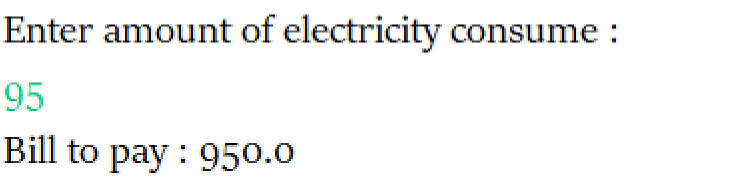
Que 4. Given an integer U denoting the amount of KWh units of electricity consumed, the task is to calculate the electricity bill with the help of the below charges:
1 to 100 units – Rs. 10/unit
100 to 200 units – Rs. 15/unit
200 to 300 units – Rs. 20/unit
above 300 units – Rs. 25/unit

```
package Lab_Assignment;
import java.util.Scanner;
public class ElectricityBill {
	public static void main(String[] args) {
	long units;
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter amount of electricity consume : ");
	units = sc.nextLong();
	double billPay = 0;
	if (units <= 100)
   {
     billPay =  units * 10;
   }
   else if (units <= 200)
   {
        billPay = (100 * 10) +(units - 100) * 15;
   }
   else if (units <= 300)
   {
        billPay = (100 * 10) +(100 * 15) +(units - 200) * 20;
   }
   else if (units > 300)
   {
        billPay = (100 * 10) + (100 * 15) + (100 * 20) + (units - 300) * 25;
   }
	System.out.println("Bill to pay : " + billPay);
	}
}
```

OUTPUT:



```
Enter amount of electricity consume :
95
Bill to pay : 950.0
```

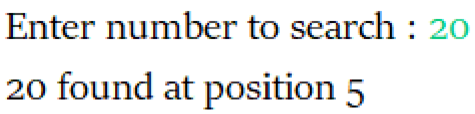


```
Enter amount of electricity consume :
95
Bill to pay : 950.0
```

Que 5. Write a java program that define a sorted array of size N and an integer K, find the position at which K is present in the array using binary search.

```
 package Lab_Assignment;
import java.util.Scanner;
public class BinarySearch {
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                int arr[] = {5,7,9,11,15,20,30,45,89,90};
                System.out.print("Enter number to search : ");
                int n = sc.nextInt();
                int l =0;
                int h = arr.length -1;
                int index = -1;
                while(l<=h) {
                        int m = (l+h)/2;
                        if(arr[m]<n)
                                l = m+1;
                        else if(arr[m]>n)
                                h = m-1;
                        else {
                                index = m;
                                break;
                        }
                }
                if(index==-1) {
                        System.out.println("Search element is not found");
                }
                else {
                        System.out.println(n + " found at position " + index);
                }
        }
}
```

OUTPUT:

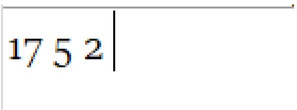


```
Enter number to search : 20
20 found at position 5
```

Que 6. Write a java program and define an array, print all the elements which are leaders. A Leader is an element that is greater than all of the elements on its right side in the array.

```java
package Lab_Assignment;
public class LeaderInArray {
         void printLeaders(int arr[], int size)
    {
        for (int i = 0; i < size; i++)
        {
            int j;
            for (j = i + 1; j < size; j++)
            {
                if (arr[i] <=arr[j])
                    break;
            }
            if (j == size) // the loop didn't break
                System.out.print(arr[i] + " ");
        }
    }
    /* program to test above functions */
    public static void main(String[] args)
    {
        LeaderInArray lead = new LeaderInArray();
        int arr[] = new int[]{16, 17, 4, 3, 5, 2};
        int n = arr.length;
        lead.printLeaders(arr, n);
    }
}
```

OUTPUT:



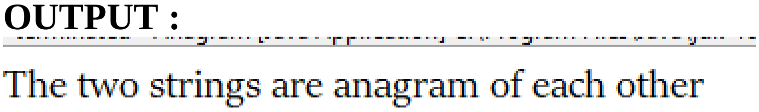
```
17 5 2
```

Que 7. Given two strings a and b consisting of lowercase characters. The task is to check whether two given strings are an anagram of each other or not. An anagram of a string is another string that contains the same characters, only the order of characters can be different. For example, abc and bca are an anagram of each other.

```java
package Lab_Assignment;
import java.util.Arrays;
```

```
public class Anagram {
        /* function to check whether two strings are
    anagram of each other */
    static boolean areAnagram(char[] str1, char[] str2)
    {
        // Get lengths of both strings
        int n1 = str1.length;
        int n2 = str2.length;
        // If length of both strings is not same,
        // then they cannot be anagram
        if (n1 != n2)
            return false;
        // Sort both strings
        Arrays.sort(str1);
        Arrays.sort(str2);
        // Compare sorted strings
        for (int i = 0; i < n1; i++)
            if (str1[i] != str2[i])
                return false;
  return true;
    }
    /* Driver Code*/
    public static void main(String args[])
    {
        char str1[] = { 't', 'e', 's', 't' };
        char str2[] = { 't', 't', 'e', 's' };
        // Function Call
        if (areAnagram(str1, str2))
            System.out.println("The two strings are" + " anagram of each other");
        else
            System.out.println("The two strings are not"+ " anagram of each other");
    }
}
```

OUTPUT :



```
The two strings are anagram of each other
```