Strings

Basic string creation:

```
import java.util.*;
public class Stringcreation {
    public static void main(String args[]){
        Scanner sc= new Scanner(System.in);
        String name=sc.nextLine();
        System.out.println(name);
    }


}
```

```
public class Stringlength {
    //String concatenation
    public static void main(String args[]){
        String f_name="tony";
        String S_name="stark";
        String Fullname=f_name +" "+S_name;
        System.out.println(Fullname);           // in arrays we will not put
parenthesis because length is a propery -.length
        System.out.println(Fullname.length());// in Strings  length is a function
so we put parenthesis()-.length()

    }

}
Output:
tony stark
10
```

```
public class printingstrings {

    public static void printstring(String Fullname){
```

```
        for(int i=0;i<Fullname.length();i++){
            System.out.print(Fullname.charAt(i)+" ");
        }

    }
    public static void main(String args[]){
        String f_name="tony";
        String S_name="stark";
        String Fullname=f_name +" "+S_name;

        printstring(Fullname);
    }
}

Output:
t o n y   s t a r k
```

palindrome

```
import java.util.Scanner;
public class palindrome {
    public static boolean ispolindrome(String pdrome){
        int n=pdrome.length();
    for(int i=0;i<n/2;i++){
        if(pdrome.charAt(i)!=pdrome.charAt(n-1-i)){
            return false;
        }
    }
    return true;}
    public static void main(String args[]){
        Scanner sc= new Scanner(System.in);
        String  pdrome=sc.nextLine();
       System.out.println( ispolindrome(pdrome));

    }

}
Output:
noon
true
```

```java
// program to find shortest distance

public class findindshortestdistance {
    public static float getshortestpath(String path){
        int x=0,y=0;
        for(int i=0;i<path.length();i++){
            //north
            if(path.charAt(i)=='N'){
                y++;
            }
            //south
            if(path.charAt(i)=='S'){
                y--;
            }
            if(path.charAt(i)=='W'){
                x--;
            }
            if(path.charAt(i)=='E'){
                x++;
            }

        }
        int x2=x*x;
        int y2=y*y;

       return((float)Math.sqrt(x2+y2));

    }

    public static void main(String args[]){
        String path="SN";
        System.out.println(getshortestpath(path));

    }

}
```

Substrings

```java
public class substrings {

    public static String printsubstring(String str,int si,int ei){
```

```
        String substring ="";// empty string
        for(int i=si;i<ei;i++){
            substring+=str.charAt(i);

        }
        return substring;
    }
    public static void main(String args[]){
        String str="saikiran";
    //    System.out.println( printsubstring(str,0,5));

    //JAVA contains inbuilt substring function str.substring(si,ei)

        System.out.println(str.substring(0,5));


    }

}
Output: saiki
```

Print largest string:

```
public class printlargeststrings {
    public static void main(String args[]){
        String fruits[]={"apple","mango","banana"};
         String largest=fruits[0];
        for(int i=0;i<fruits.length;i++){
            if(largest.compareTo(fruits[i])<0){
                largest=fruits[i];

            }
        }
        System.out.println(largest);

    }

}
Output:
Mango
--------------------------------
```

STRING BUILDERS

```
import java.util.*;
public class stringbuilder {
    public static void main(String args[]){
        StringBuilder sb= new StringBuilder("");
        for(char ch='a';ch<='z';ch++){// time complexity=o(26) or O(n^2)
            sb.append(ch);
        }
        System.out.println(sb);
        System.out.println(sb.length());




    }

}
Output:
abcdefghijklmnopqrstuvwxyz
26
```

Convert strings to capital after space

```
public class touppercase
{
    public static String Touppercase(String str){
        StringBuilder sb= new StringBuilder("");
        char ch=Character.toUpperCase(str.charAt(0));
        sb.append(ch);
        for(int i=1;i<str.length();i++){
            if(str.charAt(i)==' ' && i<str.length()){
                sb.append(str.charAt(i));
                i++;
                sb.append(Character.toUpperCase(str.charAt(i)));}

             else{
                sb.append(str.charAt(i));
             }};

            return sb.toString();
```

```
        }



    public static void main(String args[]){
        String str="iam sai kiran";
       System.out.println( Touppercase(str));

    }

        }
Output:
Iam Sai Kiran
```

String Compression :

```
public class Stringhcompression {
    public static String Compress(String str){
    String newstr="";
    for(int i=0;i<str.length();i++){
        Integer count=1;
        while(i<str.length()-1&&str.charAt(i)==str.charAt(i+1)){
            count++;
            i++;
        }
        newstr+=str.charAt(i);// if there is no repeat it prints that letter only
        if(count>1){
            newstr+=count.toString();
        }
    }
    return newstr;

    }
    public static void main(String args[]){
        String str="aaabbbccc";
        System.out.println(Compress(str));
   }

}
Output: a3b3c3


====================================
```

```java
//by using string builders

public class Stringhcompression {
    public static String Compress(String str){
    StringBuilder sb=new StringBuilder("");
    for(int i=0;i<str.length();i++){
        Integer count=1;
        while(i<str.length()-1&&str.charAt(i)==str.charAt(i+1)){
            count++;
            i++;
        }
        sb.append(str.charAt(i));// if there is no repeat it prints that letter
only
        if(count>1){
            sb.append(count.toString());
        }
    }
    return sb.toString();

    }
    public static void main(String args[]){
        String str="aaabbbccc";
        System.out.println(Compress(str));
    }

}
```

Output: a3b3c3

```java
//Count how many times lowercase vowels occurred in a String entered by the user


import java.util.*;
public class assq1 {
    public static void vowelscount(String str){
        int count=0;
        for(int i=0;i<str.length();i++){

            char ch=str.charAt(i);
```

```
            if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'){
                count++;

            }
        }
        System.out.println(count);

    }
    public static void main(String args[]){
        Scanner sc= new Scanner(System.in);
        String str=sc.nextLine();
        vowelscount(str);

    }

}
```

Extra questions:

```
public class assq2 {
    public static void main(String args[]) {
        String str="ShradhaDidi";
        String str1="ApnaCollege";
        String str2="ShradhaDidi";
        System.out.println(str.equals(str1) +" "+str.equals(str2));

}}
False true

-------------------------------------------
public class assq3 {
    public static void main(String args[])
     {
        String str="ApnaCollege".replace("l","");
     System.out.println(str);}

}
apnacoege
```

Determine if 2 Strings are anagrams of each other.

```
import java.util.Arrays;
public class assq4 {
    public static void main(String args[]){

        String str1="earth";
```

```
        String str2="heart";

        // convert them into lover case so that if we sort them we dont need to
compare each element
        str1=str1.toLowerCase();
        str2=str2.toLowerCase();
        // first check if lengths are same
        if(str1.length()==str2.length()){
        // convert it into character array so that we can sort
        char str1chararray[]=str1.toCharArray();
        char str2chararray[]=str2.toCharArray();
        //sort two strings so that both sequence will be same
        Arrays.sort(str1chararray);
        Arrays.sort(str2chararray);
    // if the arrays are same then it is anagram
    boolean result=Arrays.equals(str1chararray,str2chararray);
    if(result){
        System.out.println(str1 +" and "+str2+" are anagrams");
    }
    else{
        System.out.println(str1+" and "+str2+" are  not anagrams");
    }

    }
    else{
        System.out.println(str1+" and "+str2+" are  not anagrams");
    }
}

}
Output:
earth and heart are anagrams
```