Week -4

4a)Write a program to check if a given string is a pangram

Aim: To Write a program to check if a given string is a pangram

Program:

```
public class Pangram {

public static void checkPangram(String str)

{

str=str.toLowerCase();

boolean result=false;

if(str.length()<26){ result=false; }

else

{

for(char ch='a';ch<='z';ch++)

{

if((str.indexOf(ch)<0))

{

result=false;

break;

}    else{

result=true;

}

}

}
```

```
if(result)
{
System.out.println("str is pangram");
}   else{
System.out.println("str is not pangram");
}
}
 public static void main(String[] args){
   checkPangram("welcome  to java programming");
   checkPangram("abcdefghijklmnopqrstuvwxyz");

}}
```

Output:

```
C:\Users\admin\Desktop>javac Pangram.java

C:\Users\admin\Desktop>java Pangram
str is not pangram
str is pangram

C:\Users\admin\Desktop>
```

b) Write a program to find the most frequently occurring character in a string.

Aim: To write a program to find the most frequently occurring character in a string.

Program:

```
public class HelloWorld
{
    public static char findmost(String str)
{
   if(str==null || str.isEmpty())
{
System.out.println("String must not be null or empty.");
}
    int[] frequency=new int[256];
   for(char ch:str.toCharArray())
{
    frequency[ch]++;
}
     char most='\0';
     int maxCount=0;
        for(int i=0;i<frequency.length;i++){
      if(frequency[i]>maxCount){
      maxCount=frequency[i];
```

```
      most=(char) i;
     }
   }
        return most;
}
 public static void main(String[] args) {
String str="engineering";
 System.out.println("Most frequent character in "+ str +" is-"+findmost(str));
}
}
```

Output:

```
C:\Users\admin\Desktop>java HelloWorld
Most frequent character in engineering is-e

C:\Users\admin\Desktop>
```

c) Write a program to find all permutations of a given string.

Aim: To write a program to find all permutations of a given string.

Program:

```
class Demo {
     void printPermutations(String str) {

          char[] charArray=str.toCharArray();
  generatePermutations(charArray, 0,charArray.length -1);
  }
 void generatePermutations(char[] charArray,int left,int right)
 {
             if(left==right){
         System.out.println(new String(charArray));
     }
 else{
   for(int i=left;i<=right;i++){
  swap(charArray,left,i);
  generatePermutations(charArray,left+1,right);
   swap(charArray,left,i);
}
 }
```

```
  }
void swap(char[] charArray,int i,int j) {
char temp=charArray[i];
charArray[i]=charArray[j];
charArray[j]=temp;
 }
}
public class Permutation{
public static void main(String args[]){
Demo d=new Demo();
String str="sam";
d.printPermutations(str);
}
}
```

Output:

```
C:\Users\admin\Desktop>javac Permutation.java

C:\Users\admin\Desktop>java Permutation
sam
sma
asm
ams
mas
msa
```

d) Write a program to check if a given string is a anagram

Aim: To write a program to check if a given string is a anagram

Program:

```
import java.util.Arrays;
public class Acheck{
  public static boolean Anagrams(String s1,String s2)
{                                  // sort both strings
    char[] a1=s1.toCharArray();
    char[] a2=s2.toCharArray();
    Arrays.sort(a1);
    Arrays.sort(a2);
     return Arrays.equals(a1,a2);
}
   public static void main(String[] args)
   {
        String str1="arm";
        String str2="ram ";
              if(Anagrams(str1,str2)) {
      System.out.println("String is a anagram");
           }else{
```

```
            System.out.println("String is not a anagram");

             }

    }

}
```

Output:

```
C:\Users\admin\Desktop>java Acheck
String is a anagram

C:\Users\admin\Desktop>
```