

1.Do all same all program
for linkedlist
, hashset, and for hashmap
which you
in arraylist.

Answer:

1.

```
Import java.util.*;  
class hashmap{public static void  
main(String args[])  
{
```

```
HashMap<String,Integer> map = new  
HashMap<String,Integer>();
```

```
map.put("java",1);
```

```
map.put("c",20);  
    map.put("c++",30);  
map.put("ada",40;  
HashMap<String,Integer> map2=new  
HashMap<>();  
map2.put("c#",50);  
map2.put("python",60);
```

```
System.out.println("Is the key '5'  
present? " + map.containsKey(5));
```

```
System.out.println("Is the key '10'  
present? " + map.containsKey(10));
```

```
System.out.println("Is the value  
'World' present? " +  
map.containsValue("c"));
```

```
System.out.println("Is the map  
empty? " + map.isEmpty());
```

```
    System.out.println("The set is: "  
+ map.entrySet());
```

```
System.out.println("The Value is: "  
+ map.get(10));
```

```
System.out.println("The set is: " +  
map.keySet());
```

```
System.out.println("The size of the  
map is " + map.size());
```

```
map.put("J2SE",15);
```

```
HashMap<String,Integer>  
new_hash_map = new  
HashMap<String,Integer>(_hash_map.p  
utAll(map);
```

```
System.out.println("The cloned map  
look like this: " + map.clone());  
}  
}
```

2.

```
import java.util.*;
class hashset{public static void
main(String args[])
{HashSet<String> arr=new
HashSet<String>();
arr.add("collection framework in
java:");
arr.add("Arraylist");arr.add("Vecto
r");
arr.add("List");
```

```
HashSet<String> arr2=new
HashSet<String>();
arr2.add("Linked List");
arr2.add("Vector");
```

```
System.out.println("\nDoes set 1
contains set 2: "+
arr.containsAll(arr2));
```

```
System.out.println("Elements in
hashset: " +arr);
```

```
System.out.println() ;
```

```
boolean value = arr.equals(arr2) ;  
// print the value
```

```
System.out.println("Are both set  
equal: "+ value) ;
```

```
boolean flag =  
arr.contains("List") ;  
if (flag == true) {  
System.out.println("hashset  
contains element list") ;
```

```
System.out.println() ;  
}  
else{ System.out.println("hashset  
doesn't contains element list") ;
```

```
System.out.println() ; }
```

```
arr.remove(2) ;
```

```
System.out.println("hashset After  
Removing Element at index 2= "  
+arr);
```

```
System.out.println();
```

```
HashSet cloned_set = new HashSet();  
// Cloning the set using clone()
```

```
method cloned_set =  
(HashSet)arr.clone(); // Displaying  
the new Set after Cloning;
```

```
System.out.println("The new set: "  
+ cloned_set);
```

```
System.out.println("Size of  
hashset is :" + arr.size());
```

```
System.out.println();Iterator  
value2 = arr.iterator();  
// Displaying the values after it  
iterator values are
```

```
    while (value2.hasNext())
    {
System.out.println(value2.next());
    }

Object[] object = arr.toArray();

for(int i=0;i<object.length;i++)

{System.out.println("Value at index
"+i+" of Array converted from
hashset= "+object[i]);

System.out.println();}

System.out.println("HashCode value:
" + arr.hashCode());

boolean flag1 = arr.isEmpty();

if(flag1==true) {
```

```
System.out.println("ArrayList is  
Empty"); }
```

```
else{ System.out.println("ArrayList  
is not Empty");
```

```
} System.out.println();
```

```
arr.clear();
```

```
System.out.println("empty arraylist  
after using clear method: "  
+arr);}}
```

3.

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



```
{LinkedList<String> l1 = new
LinkedList<String>();

l1.add("A");

l1.add("B");

l1.addLast("C");

l1.addFirst("D");

l1.add(2, "E");

    System.out.println(l1);
    l1.remove(3);

l1.removeFirst();
    l1.set(1, "F");

for (String str : l1)

    System.out.print(str + " ");

    LinkedList sec_list = new
LinkedList();
```

```
    sec_list = (LinkedList)
l1.clone();
```

```
System.out.println("Second
LinkedList is:" + sec_list);
```

```
    System.out.println("The Object
that is replaced is: " + l1.set(2,
"G"));
```

```
System.out.println("The element is:
" + l1.get(2));
```

```
    System.out.println("Does the List
contains 'A': " +
l1.contains("A"));l1.push("Z");
```

```
    System.out.println(l1); String s =
l1.pop();
```

```
    System.out.println(s);
```

```
Collection<String> collect = new
ArrayList<String>();
```

```
    collect.add("A");
```

```
collect.add("Computer");  
l1.addAll(collect);  
System.out.println("The first  
element is: " + l1.getFirst());
```

```
System.out.println("The last  
element is: " + l1.getLast());
```

```
System.out.println("The first  
element is: " + l1.removeFirst());
```

```
System.out.println("The last  
element is: " + l1.removeLast());  
l1.addLast("Last");
```

```
l1.add("L");l1.add("M");l1.add("N")  
;
```

```
System.out.println("The list is as  
follows:");
```

```
ListIterator list_Iter =  
l1.listIterator(2);
```

```
while(list_Iter.hasNext()) {  
  
System.out.println(list_Iter.next()  
); }  
  
System.out.println("The first  
occurrence of F is at index: " +  
l1.indexOf("F"));  
  
l1.clear();  
  
System.out.println("List after  
clearing all elements: " + l1); }}
```

Assignment 6:

File IO problem

Create new file 'test.txt'.

Write a data in it of about
300 words.

Read a file 'test.txt' .

Calculate the letter in
it. (a,b,c,d...)

Constant in that file
(b,c,d,f...)

Vowel in that file (a, e, I
,o ,u)

Words in a complete
file. ("JAVA")

Calcualte how many time
one charater is
repeated. (a=10,b=50,z=34...)

Save that file with other
name 'test_copy.txt'.

Answer::

```
import java.io.*;

import java.util.*;

class CreateFile

{

static final int MAX_CHAR=256;

static int countChar(String str)

{
```

```
int count5[] = new int[MAX_CHAR];

int len = str.length();

for (int i2 = 0; i2 < len; i2++)

    count5[str.charAt(i2)]++;

char chz[] = new
char[str.length()];

for (int i2 = 0; i2 < len; i2++)

{

    chz[i2] = str.charAt(i2);

    int find = 0;

    for (int j2 = 0; j2 <= i2; j2++)

    {

        if (str.charAt(i2) == chz[j2])
            find++; } if (find == 1)
```

```
System.out.println("Number of  
Occurrence of " + str.charAt(i2) +  
" is:" + count5[str.charAt(i2)]);  
  
}return 0; }
```

```
public static void main(String[]  
args) throws IOException { //  
Accept a string String str = "File  
Handling in Java using " + "  
FileWriter and FileReader in java"  
+ "Java FileWriter and FileReader  
classes are used to write and read  
data from text files they are  
Character Stream classes It is  
recommended not to use the  
FileInputStream and  
FileOutputStream classes if you  
have to read and write any textual  
information as these are Byte  
stream classes " + "FileWriter is  
useful to create a file writing  
characters into it This class  
inherits from the OutputStream  
class The constructors of this
```


class assume that the default character encoding and the default byte buffer size are acceptable To specify these values yourself construct an OutputStreamWriter on a FileOutputStream" + "FileWriter is meant for writing streams of characters For writing streams of raw bytes consider using a FileOutputStream " + "Constructors FileWriter File file Constructs " + " a FileWriter object given a File object FileWriter File file boolean append constructs a FileWriter object given a File object FileWriter FileDescriptor fd constructs a FileWriter object associated with a file descriptor FileWriter String fileName constructs a FileWriter object given a " + " file name FileWriter String fileName Boolean append Constructs a FileWriter object given a " + " file name with a Boolean indicating whether or not to append the data writing " +

"This class inherit from the
InputStreamReader Class The
constructors " + "of this class
assume that the default character
encoding and the default byte
buffer size are appropriate To
specify these values yourself
construct an InputStreamReader on a
FileInputStream " + "FileReader is
meant for reading streams of
characters " + " For reading
streams of raw bytes consider using
a FileInputStream " ;

```
FileWriter fw=new  
FileWriter("test.txt");
```

```
    for (int i = 0; i < str.length();  
i++)
```

```
        fw.write(str.charAt(i));
```

```
System.out.println("Writing  
successful");
```

```
fw.close();
```

```
int ch;
```

```
    FileReader fr=null;
```

```
try{
```

```
fr = new FileReader("test.txt"); }
```

```
catch (FileNotFoundException fe) {  
System.out.println("File not  
found"); }
```

```
int vowels=0,consonants=0,p=0;while  
( (ch=fr.read()) !=-  
1) { //System.out.print((char)ch); } in  
t count = 0,count1=0; char ch1;  
str=str.toLowerCase();
```

```
    for(int k = 0; k < str.length();  
k++)
```

```
    { if(str.charAt(k) != ' ') {  
count++; } if(str.charAt(k) == 'a'  
|| str.charAt(k) == 'e' ||  
str.charAt(k) == 'i' ||  
str.charAt(k) == 'o' ||  
str.charAt(k) == 'u') { vowels++; }  

```

```
    else { consonants++;  

```

```
    }  
}
```

```
char ch2[] = new char[str.length()];
```

```
for(int l=0;l<str.length();l++)
```

```
{ ch2[l] = str.charAt(l);
```

```
if( ((l>0)&&(ch2[l]!=' ')&&(ch2[l-1]!=' ')) || ((ch2[0]!=' '  
' )&&(l==0)) ) count1++; }
```

```
System.out.println("Total number  
of characters in a string: " +  
count);
```

```
System.out.println("Total number of  
vowels in test.txt file is = "  
+vowels);
```

```
System.out.println("Total number of  
consonants in test.txt file is = "  
+consonants);
```

```
System.out.println("Total number of  
words in test.txt file is= "  
+count1);
```

```
countChar(str);fr.close();
```

```
} }
```

Assignment 7:

1. Write a generic method to find the maximal element in the range [begin, end] of a list.

i/p : {2, 62, 4, 78, 6,
10, 49, 20, 59, 43, 29, 30, 56, 89,
}

out: 89.

Importantly java.util.*;

```
Public class GFG {
```

```
    Public static void main(String  
args[]) throws Exception
```

```
{
```

```
List<Integer> list = new  
ArrayList<Integer>();
```

```
    list.add(2);
```

```
    list.add(62);
```

```
    list.add(4);
```

```
    list.add(78);
```

```
    list.add(6);
```

```
    list.add(10);
```

```
    list.add(49);
```

```
    list.add(20);
```

```
    list.add(50);
```

```
List.add(43) ;
```

```
List.add(29) ;
```

```
List.add(30) ;
```

```
List.add(56) ;
```

```
List.add(89) ;
```

```
System.out.println("List: "
+ list) ;
```

```
// getting minimum value
```

```
// using min() method
```

```
Int minList =
Collections.min(list) ;
```



```
        // getting maximum value

        // using max() method

        Int maxList =
Collections.max(list);


        // printing the minimum
value

        System.out.println("Begin
value of list is: "

                                +

minList);


        // printing the maximum
value

        System.out.println("End
value of list is: "
```

```
maxList) ;  
  
    }  
  
}
```

2. Write a generic method to count the number of elements in a collection that have a specific property (for example, odd integers, even number)

```
i/p : {2 , 4 , 6 , 7 ,  
8 , 9 , 90 , 78 , 41 , 56 , 79 , 45 , 65 , 85  
}
```

out put :

even : 7

odd : 7

```
Import java.io.*;
```

```
Class EvenOdd{
```

```

    Static void CountingEvenOdd(int
arr[],

                                Int
arr_size)

    {

        Int even_count =0;

        Int odd_count = 0;


        // loop to read all the
values in

        // the array
For(int I = 0 ; I <arr_size; i++)

        {

            // checking if a number
is

```

```
by 2 // completely divisible
```

```
If ((arr[i] & 1) == 1)
```

```
    Odd_count ++ ;
```

```
Else
```

```
    Even_count ++
```

```
;
```

```
}
```

```
System.out.println( "Number  
of even"
```

```
    + " elements = " +  
even_count
```

```
    + " Number of odd  
elements = "
```

+

```
odd_count) ;
```

```
}
```

```
// Driver Code
```

```
Public static void main  
(String[] args)
```

```
{
```

```
    Int arr[] = {2, 4,  
6,7,8,9,90,78,41,56,79,45,65,85};
```

```
    Int n = arr.length;
```

```
    CountingEvenOdd(arr, n);
```

```
}
```

```
}
```