

Name : Patel Jeshankumar K.

Date : 26/01/2024

Roll No : 038

Semester : 8th – Sem.

Year : 2023-24

Subject : (801/802) – JAVA

ProblemSheet-1

- **Assignment based on Java Collection Framework. Write one program for every category.**

A) Array List :

```
ArrayList<String> Books = new ArrayList<String>();

System.out.println("\nHow Many Books You Want to Insert ? ");
System.out.print("> ");

int n = in.nextInt();

System.out.println("\nEnter Books Name List : ");
for (int i = 0; i < n; i++) {
    Books.add(in.next());
}
```

- 1. Write a Java program to create an array list, add some colors (strings) and print out the collection.**

Source Code :

```
protected void printBooks(ArrayList<String> books)
{
    System.out.print("Available Book List : "+books);
    System.out.println();
}
```

Output :

```
Choose A Question :
1

How Many Books You Want to Insert ?
> 5

Enter Books Name List :
JAVA
ASP.NET(Core)
C++
JavaScript
PHP
Available Book List : [JAVA, ASP.NET(Core), C++, JavaScript, PHP]

--- ArrayList Collection Questions ---
1.Insert and Print.
```

2. Write a Java program to iterate through all elements in an array list.

Source Code :

```
protected void IterateArraylist(ArrayList<String> books)
{
    System.out.print("\n\nAvailable Books : ");
    int cnt = 0;
    for(String c: books)
    {
        System.out.print("[ "+cnt+" ] "+ c+"\t");
        cnt++;
    }
    System.out.println();
}
```

Output :

```
Choose A Question :
2

Available Books : [0] JAVA      [1] ASP.NET(Core)      [2] C++ [3] JavaScript [4] PHP

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
```

3. Write a Java program to insert an element into the array list at the first position.

Source Code :

```
protected void InsertAtFirstPosition(ArrayList<String> books)
{
    String book;
    System.out.print("\nBook List : "+books);

    System.out.print("\nAdd Book At First : ");
    book = in.next();
    books.add(0, book);
    System.out.println("Book Added At First!!!");
}
```

```

        System.out.print("\nBook List : "+books);
        System.out.println();
    }

```

Output :

```

Choose A Question :
3

Book List : [JAVA, ASP.NET(Core), C++, JavaScript, PHP]
Add Book At First : VB.NET
Book Added At First!!

Book List : [VB.NET, JAVA, ASP.NET(Core), C++, JavaScript, PHP]

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.

```

4. Write a Java program to retrieve an element (at a specified index) from a given array list.

Source Code :

```

protected void RetriveElement(ArrayList<String> Books)
{
    int i;
    System.out.print("\nBook List : "+Books);
    System.out.print("\nEnter Index : ");
    i = in.nextInt();

    System.out.println("Book at Index ["+i+"] : "+Books.get(i));
    System.out.println();
}

```

Output :

```

Choose A Question :
4

Book List : [VB.NET, JAVA, ASP.NET(Core), C++, JavaScript, PHP]
Enter Index : 4
Book at Index [4] : JavaScript

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrive an Element of Specified Position.

```

5. Write a Java program to update an array element by the given element.

Source Code :

```

protected void UpdateElement(ArrayList<String> Books)
{
    int i;

```

```

String book;

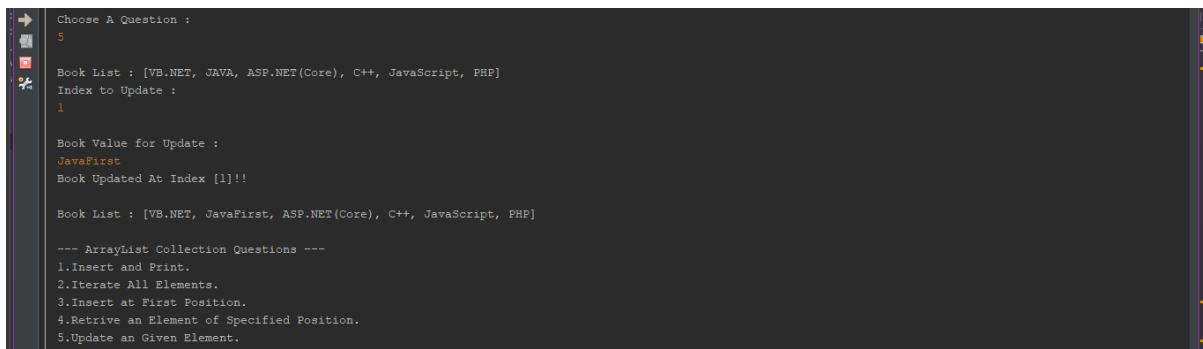
System.out.print("\nBook List : "+Books);
System.out.println("\nIndex to Update : ");
i = in.nextInt();

System.out.println("\nBook Value for Update : ");
book = in.next();

Books.set(i, book);
System.out.println("Book Updated At Index ["+i+"]!!!");
System.out.print("\nBook List : "+Books);
System.out.println();
}

```

Output :



```

Choose A Question :
5
Book List : [VB.NET, JAVA, ASP.NET(Core), C++, JavaScript, PHP]
Index to Update :
1
Book Value for Update :
JavaFirst
Book Updated At Index [1]!!!
Book List : [VB.NET, JavaFirst, ASP.NET(Core), C++, JavaScript, PHP]

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.

```

6. Write a Java program to remove the third element from an array list.

Source Code :

```

protected void RemoveThirdElement(ArrayList<String> Books)
{
    System.out.println("Third Element ["+Books.get(2)+"] Deleted!!!");
    Books.remove(2);
    System.out.println();
}

```

Output :

```
Choose A Question :
6
Third Element [ASP.NET(Core)] Deleted!!

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
```

7. Write a Java program to search for an element in an array list.

Source Code :

```
protected void SearchElement(ArrayList<String> Books)
{
    String book;

    System.out.print("\nBook List : "+Books);
    System.out.print("\nSearch a Book : ");
    book = in.next();
    System.out.println();

    for(int i=0;i<Books.size();i++)
    {
        if(Books.get(i).equalsIgnoreCase(book))
        {
            System.out.println("Book "+book+" Found At Index ["+i+"]!!");
        }
    }
}
```

Output :

```
Choose A Question :
7
Book List : [VB.NET, JavaFirst, C++, JavaScript, PHP]
Search a Book : C++
Book C++ Found At Index [2]!!

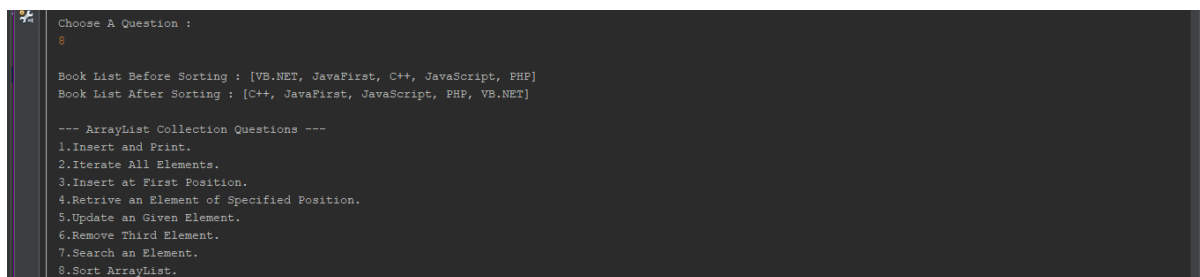
--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
```

8. Write a Java program to sort a given array list.

Source Code :

```
protected void SortArrayList(ArrayList<String> Books)
{
    System.out.print("\nBook List Before Sorting : "+Books);
    Collections.sort(Books);
    System.out.print("\nBook List After Sorting : "+Books);
    System.out.println();
}
```

Output :



```
Choose A Question :
8

Book List Before Sorting : [VB.NET, JavaFirst, C++, JavaScript, PHP]
Book List After Sorting : [C++, JavaFirst, JavaScript, PHP, VB.NET]

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
```

9. Write a Java program to copy one array list into another.

Source Code :

```
protected void CopyArrayListToAnotherArrayList(ArrayList<String> Books)
{
    ArrayList<String> CopyCat = new ArrayList<String>();
    System.out.print("\nBook List : "+Books);
    System.out.println("\n\nCopying Book List To Other List...");

    CopyCat.addAll(Books);

    System.out.println("Successfully Copped Book List To Other List...");
    System.out.print("\nCoppied Book List : "+CopyCat);
    System.out.println();
}
```

Output :

```
Choose A Question :
9

Book List : [C++, JavaFirst, JavaScript, PHP, VB.NET]

Copying Book List To Other List...
Successfully Copied Book List To Other List...

Copied Book List : [C++, JavaFirst, JavaScript, PHP, VB.NET]

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
```

10. Write a Java program to shuffle elements in an array list.

Source Code :

```
protected void SuffleElements(ArrayList<String> Books)
{
    System.out.print("\nBook List Before Suffling : "+Books);
    Collections.shuffle(Books);
    System.out.print("\nBook List After Suffling : "+Books);
    System.out.println();
}
```

Output :

```
Choose A Question :
10

Book List Before Suffling : [C++, JavaFirst, JavaScript, PHP, VB.NET]
Book List After Suffling : [JavaScript, VB.NET, C++, JavaFirst, PHP]

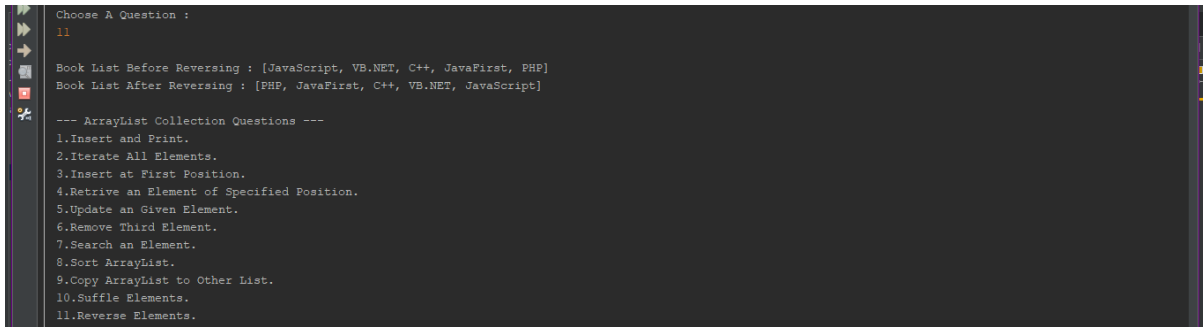
--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Suffle Elements.
```

11. Write a Java program to reverse elements in an array list.

Source Code :

```
protected void ReverseElements(ArrayList<String> Books)
{
    System.out.print("\nBook List Before Reversing : "+Books);
    Collections.reverse(Books);
    System.out.print("\nBook List After Reversing : "+Books);
    System.out.println();
}
```

Output :



```
Choose A Question :
11
Book List Before Reversing : [JavaScript, VB.NET, C++, JavaFirst, PHP]
Book List After Reversing : [PHP, JavaFirst, C++, VB.NET, JavaScript]

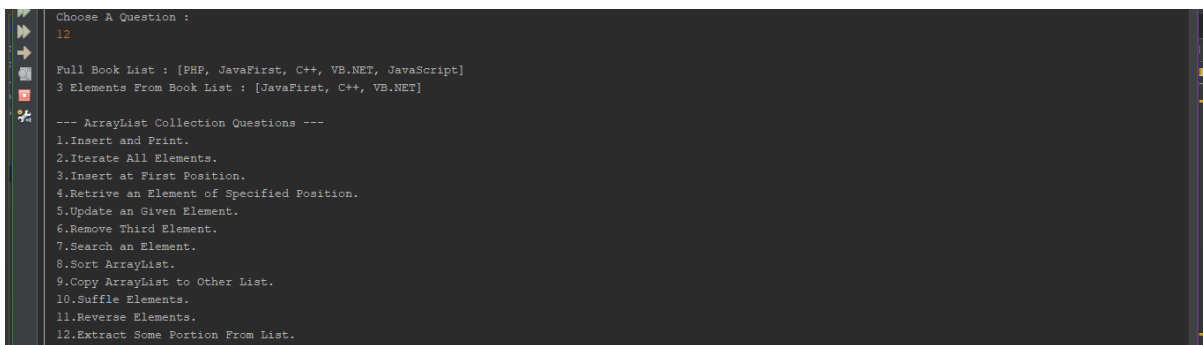
--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Shuffle Elements.
11.Reverse Elements.
```

12. Write a Java program to extract a portion of an array list.

Source Code :

```
protected void ExtractPortion(ArrayList<String> Books)
{
    System.out.print("\nFull Book List : "+Books);
    System.out.print("\n3 Elements From Book List : "+Books.subList(1, 4));
    System.out.println();
}
```

Output :



```
Choose A Question :
12
Full Book List : [PHP, JavaFirst, C++, VB.NET, JavaScript]
3 Elements From Book List : [JavaFirst, C++, VB.NET]

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Shuffle Elements.
11.Reverse Elements.
12.Extract Some Portion From List.
```

13. Write a Java program to compare two array lists.

Source Code :

```
protected void CompareArrayLists(ArrayList<String> Books)
{
    ArrayList<String> Book1 = new ArrayList<String>();
    ArrayList<String> Book2 = new ArrayList<String>();

    Book1.add("Java");
    Book1.add("JavaScript");
```



```

Book1.add("HTML");
Book1.add("CSS");

Book2.addAll(Books);

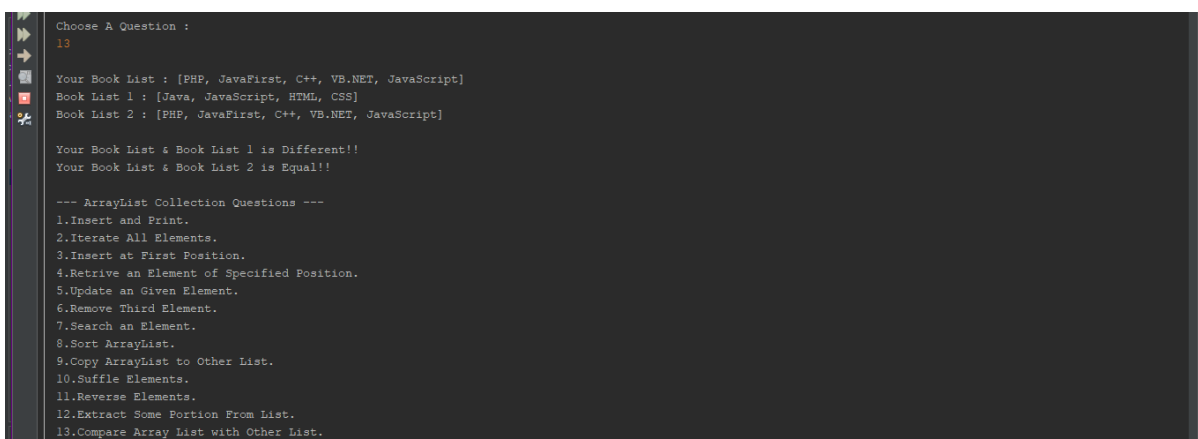
System.out.print("\nYour Book List : "+Books);
System.out.print("\nBook List 1 : "+Book1);
System.out.print("\nBook List 2 : "+Book2);
System.out.println();

if(Books.equals(Book1))
{
    System.out.print("\nYour Book List & Book List 1 is Equal!!");
}
else
{
    System.out.print("\nYour Book List & Book List 1 is Different!!");
}

if(Books.equals(Book2))
{
    System.out.print("\nYour Book List & Book List 2 is Equal!!");
}
else
{
    System.out.print("\nYour Book List & Book List 2 is Different!!");
}
System.out.println();
}

```

Output :



```

Choose A Question :
13
Your Book List : [PHP, JavaFirst, C++, VB.NET, JavaScript]
Book List 1 : [Java, JavaScript, HTML, CSS]
Book List 2 : [PHP, JavaFirst, C++, VB.NET, JavaScript]

Your Book List & Book List 1 is Different!!
Your Book List & Book List 2 is Equal!!

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Suffle Elements.
11.Reverse Elements.
12.Extract Some Portion From List.
13.Compare Array List with Other List.

```

14. Write a Java program that swaps two elements in an array list.

Source Code :

```
protected void SwapTwoElements(ArrayList<String> Books)
{
    String ele1, ele2;
    int i1, i2;
    i1 = 0;
    i2 = 0;

    System.out.print("\nBook List : "+Books);

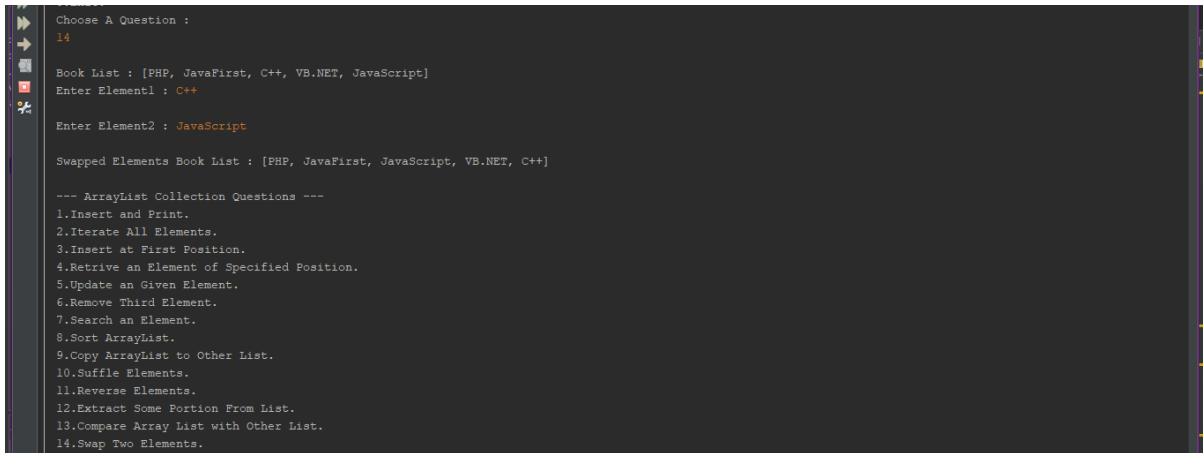
    System.out.print("\nEnter Element1 : ");
    ele1 = in.next();
    System.out.print("\nEnter Element2 : ");
    ele2 = in.next();

    for(int i=0;i<Books.size();i++)
    {
        if(Books.get(i).equalsIgnoreCase(ele1))
        {
            i1 = i;
        }
        else if(Books.get(i).equalsIgnoreCase(ele2))
        {
            i2 = i;
        }
    }

    Collections.swap(Books, i1, i2);

    System.out.print("\nSwapped Elements Book List : "+Books);
    System.out.println();
}
```

Output :



```
Choose A Question :
14
Book List : [PHP, JavaFirst, C++, VB.NET, JavaScript]
Enter Element1 : C++
Enter Element2 : JavaScript
Swapped Elements Book List : [PHP, JavaFirst, JavaScript, VB.NET, C++]

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Shuffle Elements.
11.Reverse Elements.
12.Extract Some Portion From List.
13.Compare Array List with Other List.
14.Swap Two Elements.
```

15. Write a Java program to join two array lists.

Source Code :

```
protected void JoinTwoArrayLists(ArrayList<String> Books)
{
    ArrayList<String> Books2 = new ArrayList<String>();

    int n;
    System.out.println("Insert Elements in Second Arraylist");

    System.out.println("How Many Books You Want to Insert ? ");
    System.out.print("> ");
    n = in.nextInt();

    System.out.println("\nEnter Books Name List : ");
    for (int i = 0; i < n; i++) {
        Books2.add(in.next());
    }

    System.out.print("\nYour Book List : "+Books);
    System.out.print("\nSecond Book List : "+Books2);
    System.out.println();

    Books.removeAll(Books2);
    Books.addAll(Books2);

    System.out.print("\nJoined Book List : "+Books);
    System.out.println();
}
```

Output :

```
Choose A Question :
15
Insert Elements in Second ArrayList
How Many Books You Want to Insert ?
> 3

Enter Books Name List :
Android
MongoDB
SQL

Your Book List : [PHP, JavaFirst, JavaScript, VB.NET, C++]
Second Book List : [Android, MongoDB, SQL]

Joined Book List : [PHP, JavaFirst, JavaScript, VB.NET, C++, Android, MongoDB, SQL]

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Shuffle Elements.
11.Reverse Elements.
12.Extract Some Portion From List.
13.Compare Array List with Other List.
14.Swap Two Elements.
15.Join Two Array Lists.
```

16. Write a Java program to clone an array list to another array list.

Source Code :

```
protected void CloneArrayListToOtherList(ArrayList<String> Books)
{
    ArrayList Books2 = (ArrayList)Books.clone();

    System.out.print("\nYour Book List : "+Books);
    System.out.print("\nCloned Book List : "+Books2);
    System.out.println();
}
```

Output :

```
Choose A Question :
16

Your Book List : [PHP, JavaFirst, JavaScript, VB.NET, C++, Android, MongoDB, SQL]
Cloned Book List : [PHP, JavaFirst, JavaScript, VB.NET, C++, Android, MongoDB, SQL]

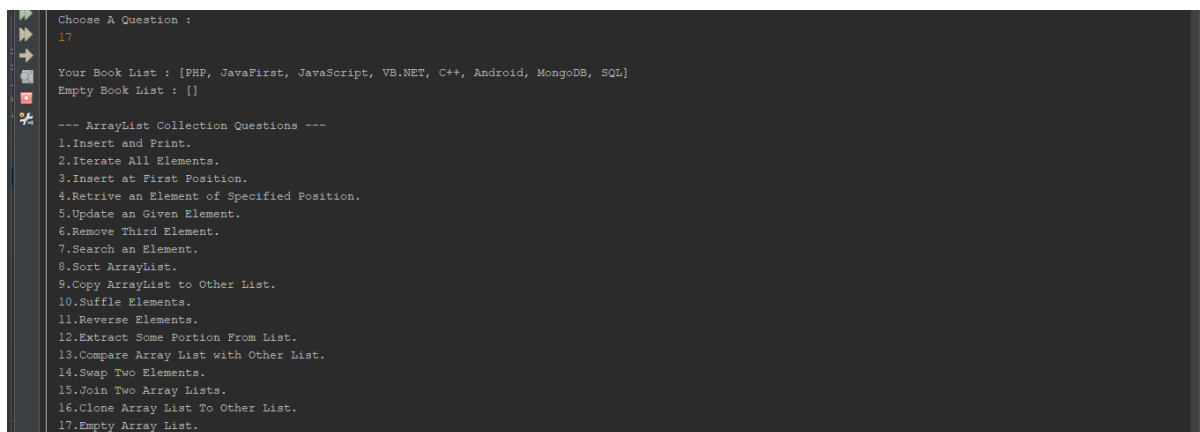
--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Shuffle Elements.
11.Reverse Elements.
12.Extract Some Portion From List.
13.Compare Array List with Other List.
14.Swap Two Elements.
15.Join Two Array Lists.
16.Clone Array List To Other List.
```

17. Write a Java program to empty an array list.

Source Code :

```
protected void MakeListEmpty(ArrayList<String> Books)
{
    System.out.print("\nYour Book List : "+Books);
    Books.removeAll(Books);
    System.out.print("\nEmpty Book List : "+Books);
    System.out.println();
}
```

Output :



```
Choose A Question :
17
Your Book List : [PHP, JavaFirst, JavaScript, VB.NET, C++, Android, MongoDB, SQL]
Empty Book List : []

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Shuffle Elements.
11.Reverse Elements.
12.Extract Some Portion From List.
13.Compare Array List with Other List.
14.Swap Two Elements.
15.Join Two Array Lists.
16.Clone Array List To Other List.
17.Empty Array List.
```

18. Write a Java program to test whether an array list is empty or not.

Source Code :

```
protected void CheckListEmpty(ArrayList<String> Books)
{
    if(Books.isEmpty())
    {
        System.out.print("\nYour Book List is Empty!!");
    }
    else
    {
        System.out.print("\nYour Book List is not Empty, contains : "+Books);
    }
    System.out.println();
}
```

Output :

```
Choose A Question :
18
Your Book List is Empty!!

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Shuffle Elements.
11.Reverse Elements.
12.Extract Some Portion From List.
13.Compare Array List with Other List.
14.Swap Two Elements.
15.Join Two Array Lists.
16.Clone Array List To Other List.
17.Empty Array List.
18.Check Empty Array List.
```

19. Write a Java program for trimming the capacity of an array list.

Source Code :

```
protected void TrimCapacity(ArrayList<String> Books)
{
    System.out.print("\nYour Book List : "+Books);
    Books.trimToSize();
    System.out.print("\nList Trimmed Size : "+Books.size());
    System.out.println();
}
```

Output :

```
Choose A Question :
19
Your Book List : [JAVA, Android, JavaScript, NodeJS, C++]
List Trimmed Size : 5

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Shuffle Elements.
11.Reverse Elements.
12.Extract Some Portion From List.
13.Compare Array List with Other List.
14.Swap Two Elements.
15.Join Two Array Lists.
16.Clone Array List To Other List.
17.Empty Array List.
18.Check Empty Array List.
19.Trim Capacity of Array List.
```

20. Write a Java program to increase an array list size.

Source Code :

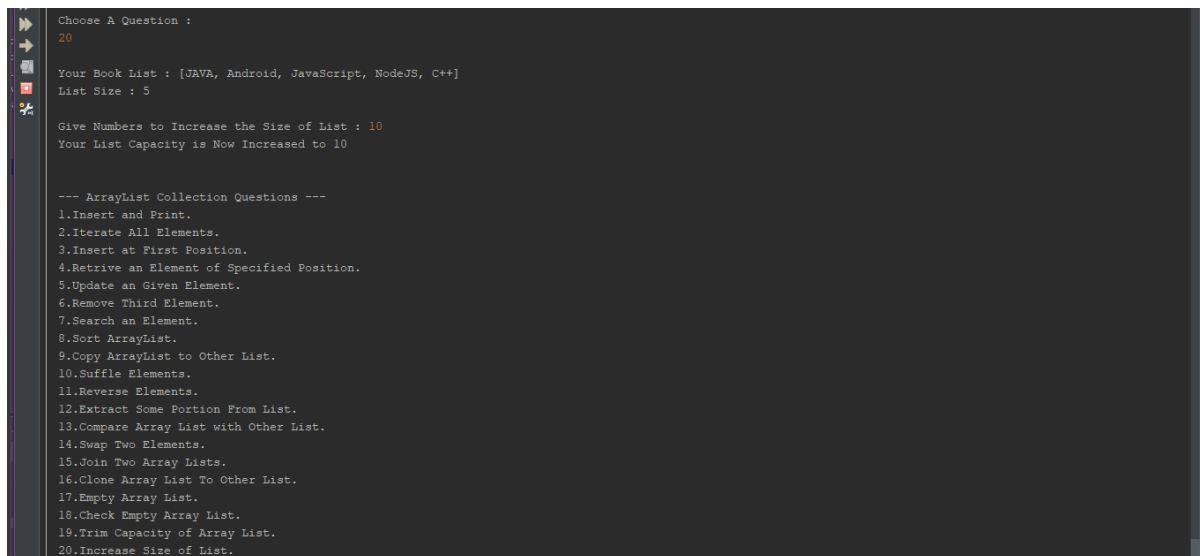
```
protected void IncreaseSize(ArrayList<String> Books)
{
    int capacity;

    System.out.print("\nYour Book List : "+Books);
    System.out.print("\nList Size : "+Books.size());

    System.out.print("\n\nGive Numbers to Increase the Size of List : ");
    capacity = in.nextInt();

    Books.ensureCapacity(capacity);
    System.out.println("Your List Capacity is Now Increased to "+capacity);
    System.out.println();
}
```

Output :



```
Choose A Question :
20

Your Book List : [JAVA, Android, JavaScript, NodeJS, C++]
List Size : 5

Give Numbers to Increase the Size of List : 10
Your List Capacity is Now Increased to 10

--- ArrayList Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Insert at First Position.
4.Retrieve an Element of Specified Position.
5.Update an Given Element.
6.Remove Third Element.
7.Search an Element.
8.Sort ArrayList.
9.Copy ArrayList to Other List.
10.Shuffle Elements.
11.Reverse Elements.
12.Extract Some Portion From List.
13.Compare Array List with Other List.
14.Swap Two Elements.
15.Join Two Array Lists.
16.Clone Array List To Other List.
17.Empty Array List.
18.Check Empty Array List.
19.Trim Capacity of Array List.
20.Increase Size of List.
```

21. Write a Java program to replace the second element of an ArrayList with the specified element.

Source Code :

```
protected void Replace2ndElement(ArrayList<String> Books)
{
    System.out.print("\nYour Book List : "+Books);
    System.out.println();
}
```

```

        System.out.print("\nGive Value to Replace 2nd Element : ");
        Books.set(1, in.next());

        System.out.println();
        System.out.print("\nUpdated Book List : "+Books);
        System.out.println();
    }
}

```

Output :

```

21.Replace 2nd Element.
22.Print Elements With Positions.
0.Exit.
Choose A Question :
21

Your Book List : [JAVA, Android, JavaScript, NodeJS, C++]

Give Value to Replace 2nd Element : BlackBook(HTML)

Updated Book List : [JAVA, BlackBook(HTML), JavaScript, NodeJS, C++]

```

22. Write a Java program to print all the elements of an ArrayList using the elements position.

Source Code :

```

protected void PrintAllElements(ArrayList<String> Books)
{
    System.out.println();
    int cnt = 0;
    for(String c: Books)
    {
        System.out.println("Position ["+cnt+"], Element["+ c+"]\t");
        cnt++;
    }
    System.out.println();
}

```

Output :

```

22.Print Elements With Positions.
0.Exit.
Choose A Question :
22

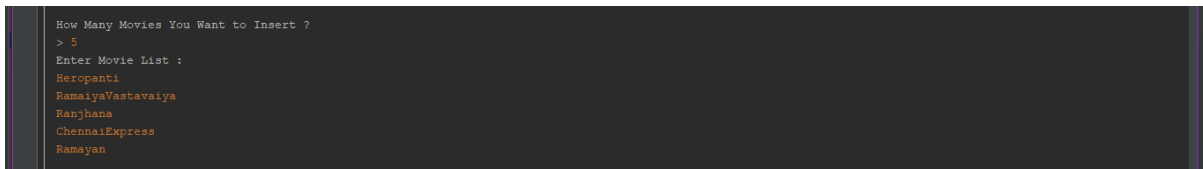
Position [0], Element [JAVA]
Position [1], Element [BlackBook(HTML)]
Position [2], Element [JavaScript]
Position [3], Element [NodeJS]
Position [4], Element [C++]

```


B) Linked List :

```
int ch1, n;
LinkedList<String> Movies = new LinkedList<String>();
System.out.println("\nHow Many Movies You Want to Insert ? ");
System.out.print("> ");
n = in.nextInt();

System.out.println("Enter Movie List : ");
for (int i = 0; i < n; i++) {
    Movies.add(in.next());
}
```



```
How Many Movies You Want to Insert ?
> 5
Enter Movie List :
Heropanti
RamaiyaVastavaiya
Ranjhana
ChennaiExpress
Ramayan
```

1. Write a Java program to append the specified element to the end of a linked list.

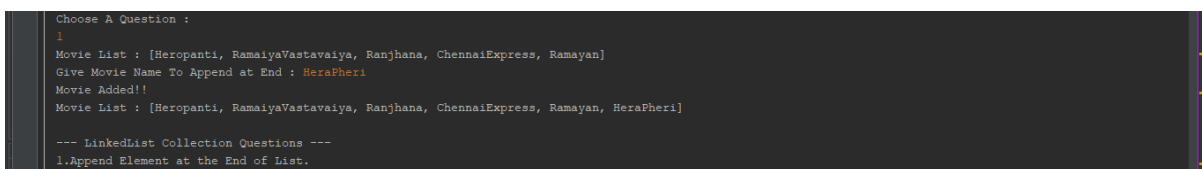
Source Code :

```
protected void AppenEnd(LinkedList<String> Movies) {
    System.out.print("Movie List : " + Movies);
    System.out.println();

    System.out.print("Give Movie Name To Append at End : ");
    Movies.addLast(in.next());

    System.out.println("Movie Added!!");
    System.out.print("Movie List : " + Movies);
    System.out.println();
}
```

Output :



```
Choose A Question :
1
Movie List : [Heropanti, RamaiyaVastavaiya, Ranjhana, ChennaiExpress, Ramayan]
Give Movie Name To Append at End : HeraPheri
Movie Added!!
Movie List : [Heropanti, RamaiyaVastavaiya, Ranjhana, ChennaiExpress, Ramayan, HeraPheri]

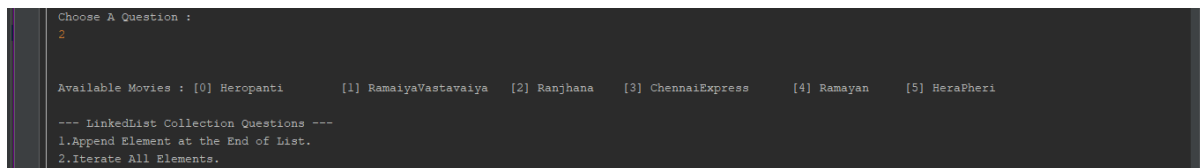
--- LinkedList Collection Questions ---
1.Append Element at the End of List.
```

2. Write a Java program to iterate through all elements in a linked list.

Source Code :

```
protected void IterateAllElements(LinkedList<String> Movies) {
    System.out.print("\n\nAvailable Movies : ");
    int cnt = 0;
    for (String m : Movies) {
        System.out.print "[" + cnt + " ] " + m + "\t";
        cnt++;
    }
    System.out.println();
}
```

Output :



```
Choose A Question :
2

Available Movies : [0] Heropanti      [1] RamaiyaVastavaiya  [2] Ranjhana    [3] ChennaiExpress  [4] Ramayan    [5] HeraPheri

--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
```

3. Write a Java program to iterate through all elements in a linked list starting at the specified position.

Source Code :

```
protected void IterateAllElementsPosition(LinkedList<String> Movies) {
    int pos;
    int cnt = 0;
    System.out.print("Give Position (Start with 0) : ");
    pos = in.nextInt();

    System.out.print("\n\nAvailable Movies : ");

    for (int i = pos; i < Movies.size(); i++) {
        System.out.print "[" + i + " ] " + Movies.get(i) + "\t";
    }
    System.out.println();
}
```

Output :

```
Choose A Question :
3
Give Position (Start with 0) : 2

Available Movies : [2] Ranjhana [3] ChennaiExpress [4] Ramayan [5] HeraPheri

--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
3.Iterate From Given Position.
```

4. Write a Java program to iterate a linked list in reverse order.

Source Code :

```
protected void IterateReverse(LinkedList<String> Movies) {
    System.out.print("\nMovie List : \n");
    for (int i = 0; i < Movies.size(); i++) {
        System.out.println("Position[" + i + "], Movie[" + Movies.get(i) + "]);
    }

    Collections.reverse(Movies);

    System.out.print("\nMovie List in Reverse Order : \n");
    for (int i = 0; i < Movies.size(); i++) {
        System.out.println("Position[" + i + "], Movie[" + Movies.get(i) + "]);
    }
    System.out.println();
}
```

Output :

```
Choose A Question :
4

Movie List :
Position[0], Movie[Heropanti]
Position[1], Movie[RamayiVastavaiya]
Position[2], Movie[Ranjhana]
Position[3], Movie[ChennaiExpress]
Position[4], Movie[Ramayan]
Position[5], Movie[HeraPheri]

Movie List in Reverse Order :
Position[0], Movie[HeraPheri]
Position[1], Movie[Ramayan]
Position[2], Movie[ChennaiExpress]
Position[3], Movie[Ranjhana]
Position[4], Movie[RamayiVastavaiya]
Position[5], Movie[Heropanti]

--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
3.Iterate From Given Position.
4.Iterate In Reverse Order.
```

5. Write a Java program to insert the specified element at the specified position in the linked list.

Source Code :

```
protected void InsertAtPosition(LinkedList<String> Movies) {
    System.out.print("\nMovie List : ");
    for (int i = 0; i < Movies.size(); i++) {
        System.out.print("[ " + i + " ] " + Movies.get(i) + "\t");
    }

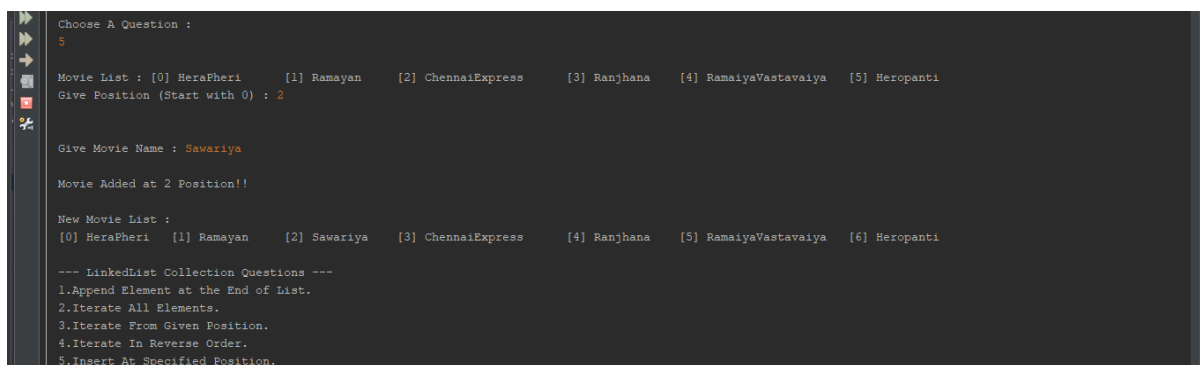
    int pos;

    System.out.print("\nGive Position (Start with 0) : ");
    pos = in.nextInt();
    System.out.println();

    System.out.print("\nGive Movie Name : ");
    Movies.add(pos, in.next());

    System.out.println("\nMovie Added at " + pos + " Position!!");
    System.out.print("\nNew Movie List : \n");
    for (int i = 0; i < Movies.size(); i++) {
        System.out.print("[ " + i + " ] " + Movies.get(i) + "\t");
    }
    System.out.println();
}
```

Output :



```
Choose A Question :
5
Movie List : [0] HeraPheri    [1] Ramayan    [2] ChennaiExpress    [3] Ranjhana    [4] RamaiyaVastavaiya    [5] Heropanti
Give Position (Start with 0) : 2
Give Movie Name : Sawariya
Movie Added at 2 Position!!
New Movie List :
[0] HeraPheri    [1] Ramayan    [2] Sawariya    [3] ChennaiExpress    [4] Ranjhana    [5] RamaiyaVastavaiya    [6] Heropanti

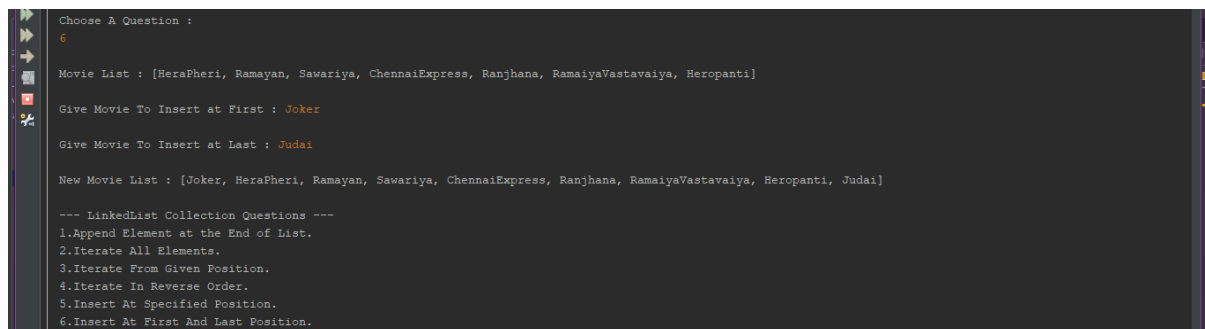
--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
3.Iterate From Given Position.
4.Iterate In Reverse Order.
5.Insert At Specified Position.
```

6. Write a Java program to insert elements into the linked list at the first and last positions.

Source Code :

```
protected void InsertAtFirstLast(LinkedList<String> Movies) {  
    System.out.print("\nMovie List : " + Movies);  
  
    System.out.println();  
    System.out.print("\nGive Movie To Insert at First : ");  
    Movies.addFirst(in.next());  
  
    System.out.print("\nGive Movie To Insert at Last : ");  
    Movies.addLast(in.next());  
  
    System.out.print("\nNew Movie List : " + Movies);  
    System.out.println();  
}
```

Output :



```
Choose A Question :  
6  
Movie List : [HeraPheri, Ramayan, Sawariya, ChennaiExpress, Ranjhana, RamaiyaVastavaiya, Heropanti]  
Give Movie To Insert at First : Joker  
Give Movie To Insert at Last : Judai  
New Movie List : [Joker, HeraPheri, Ramayan, Sawariya, ChennaiExpress, Ranjhana, RamaiyaVastavaiya, Heropanti, Judai]  
  
--- LinkedList Collection Questions ---  
1.Append Element at the End of List.  
2.Iterate All Elements.  
3.Iterate From Given Position.  
4.Iterate In Reverse Order.  
5.Insert At Specified Position.  
6.Insert At First And Last Position.
```

7. Write a Java program to insert the specified element at the front of a linked list.

Source Code :

```
protected void InsertAtFront(LinkedList<String> Movies) {  
    System.out.print("\nMovie List : " + Movies);  
  
    System.out.println();  
    System.out.print("\nGive Movie To Insert at Front : ");  
    Movies.addFirst(in.next());  
  
    System.out.print("\nNew Movie List : " + Movies);  
    System.out.println();  
}
```

Output :

```
Choose A Question :
7

Movie List : [Joker, HeraPheri, Ramayan, Sawariya, ChennaiExpress, Ranjhana, RamaiyaVastavaiya, Heropanti, Judai]

Give Movie To Insert at Front : GulabGang

New Movie List : [GulabGang, Joker, HeraPheri, Ramayan, Sawariya, ChennaiExpress, Ranjhana, RamaiyaVastavaiya, Heropanti, Judai]

--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
3.Iterate From Given Position.
4.Iterate In Reverse Order.
5.Insert At Specified Position.
6.Insert At First And Last Position.
7.Insert At Front.
```

8. Write a Java program to insert the specified element at the end of a linked list.

Source Code :

```
protected void InsertAtEnd(LinkedList<String> Movies) {
    System.out.print("\nMovie List : " + Movies);

    System.out.println();
    System.out.print("\nGive Movie To Insert at End : ");
    Movies.addLast(in.next());

    System.out.print("\nNew Movie List : " + Movies);
    System.out.println();
}
```

Output :

```
Choose A Question :
8

Movie List : [GulabGang, Joker, HeraPheri, Ramayan, Sawariya, ChennaiExpress, Ranjhana, RamaiyaVastavaiya, Heropanti, Judai]

Give Movie To Insert at End : JodhaAkbar

New Movie List : [GulabGang, Joker, HeraPheri, Ramayan, Sawariya, ChennaiExpress, Ranjhana, RamaiyaVastavaiya, Heropanti, Judai, JodhaAkbar]

--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
3.Iterate From Given Position.
4.Iterate In Reverse Order.
5.Insert At Specified Position.
6.Insert At First And Last Position.
7.Insert At Front.
8.Insert At End.
```

9. Write a Java program to insert some elements at the specified position into a linked list.

Source Code :

```
protected void InsertSomeElementsAtPosition(LinkedList<String> Movies) {
    int n;
    System.out.print("\nMovie List : " + Movies);

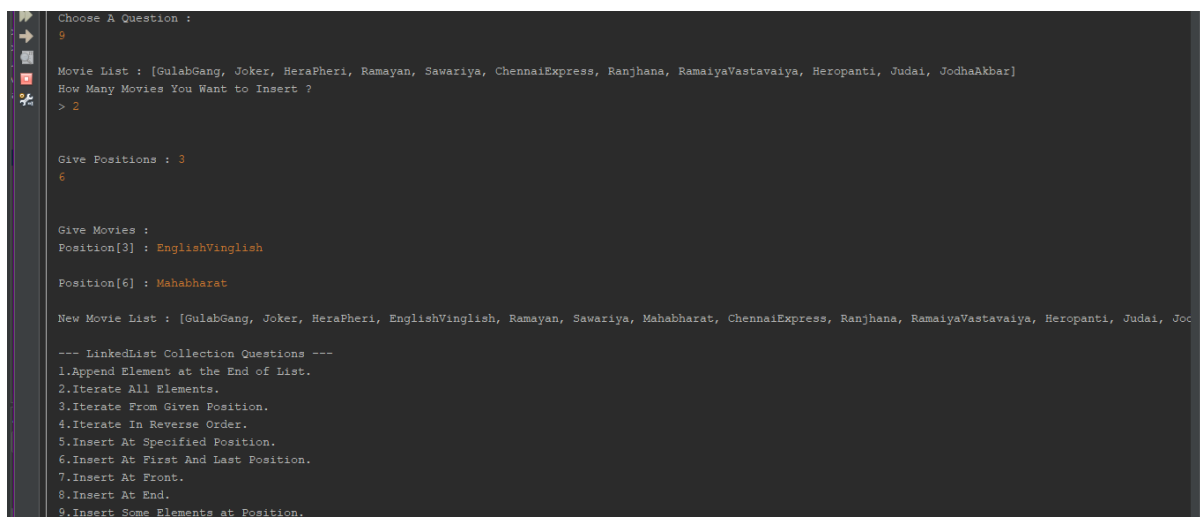
    System.out.println("\nHow Many Movies You Want to Insert ? ");
    System.out.print("> ");
    n = in.nextInt();
    int[] pos = new int[n];

    System.out.println();
    System.out.print("\nGive Positions : ");
    for (int i = 0; i < pos.length; i++) {
        pos[i] = in.nextInt();
    }

    System.out.println();
    System.out.print("\nGive Movies : ");
    for (int i : pos) {
        System.out.print("\nPosition[" + i + "] : ");
        Movies.add(i, in.next());
    }

    System.out.print("\nNew Movie List : " + Movies);
    System.out.println();
}
```

Output :



```
Choose A Question :
9
Movie List : [GulabGang, Joker, HeraPheri, Ramayan, Sawariya, ChennaiExpress, Ranjhana, RamaiyaVastavaiya, Heropanti, Judai, JodhaAkbar]
How Many Movies You Want to Insert ?
> 2

Give Positions : 3
6

Give Movies :
Position[3] : EnglishVinglish
Position[6] : Mahabharat

New Movie List : [GulabGang, Joker, HeraPheri, EnglishVinglish, Ramayan, Sawariya, Mahabharat, ChennaiExpress, Ranjhana, RamaiyaVastavaiya, Heropanti, Judai, JodhaAkbar]

--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
3.Iterate From Given Position.
4.Iterate In Reverse Order.
5.Insert At Specified Position.
6.Insert At First And Last Position.
7.Insert At Front.
8.Insert At End.
9.Insert Some Elements at Position.
```

10. Write a Java program to insert some elements at the specified position into a linked list.

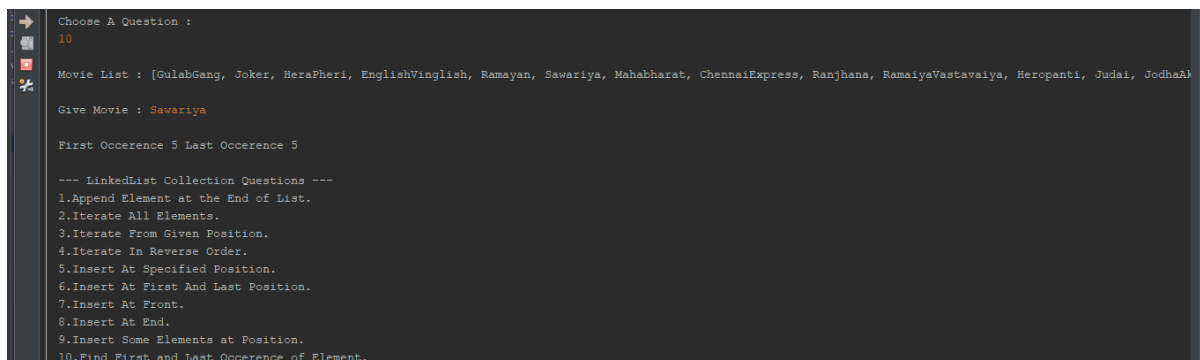
Source Code :

```
protected void GetFirstAndLastOccerence(LinkedList<String> Movies) {
    String ele;
    System.out.print("\nMovie List : " + Movies);

    System.out.println();
    System.out.print("\nGive Movie : ");
    ele = in.next();

    System.out.print("\nFirst Occerence " + Movies.indexOf(ele) + " Last Occerence " +
        Movies.lastIndexOf(ele));
    System.out.println();
}
```

Output :



```
Choose A Question :
10
Movie List : [GulabGang, Joker, HeraPheri, EnglishVinglish, Ramayan, Sawariya, Mahabharat, ChennaiExpress, Ranjhana, RamaiyaVastavaiya, Heropanti, Judai, JodhaA]
Give Movie : Sawariya
First Occerence 5 Last Occerence 5

--- LinkedList Collection Questions ---
1. Append Element at the End of List.
2. Iterate All Elements.
3. Iterate From Given Position.
4. Iterate In Reverse Order.
5. Insert At Specified Position.
6. Insert At First And Last Position.
7. Insert At Front.
8. Insert At End.
9. Insert Some Elements at Position.
10. Find First and Last Occerence of Element.
```

11. Write a Java program to display elements and their positions in a linked list.

Source Code :

```
protected void DisplayAllElementsPosition(LinkedList<String> Movies) {
    System.out.print("\nMovie List : \n");
    for (int i = 0; i < Movies.size(); i++) {
        System.out.println("Position[" + i + "], Movie[" + Movies.get(i) + "]);
    }

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


Output :

```
Choose A Question :
11

Movie List :
Position[0], Movie[GulabGang]
Position[1], Movie[Joker]
Position[2], Movie[HeraPheri]
Position[3], Movie[EnglishVinglish]
Position[4], Movie[Ramayan]
Position[5], Movie[Sawariya]
Position[6], Movie[Mahabharat]
Position[7], Movie[ChennaiExpress]
Position[8], Movie[Ranjhana]
Position[9], Movie[RamayiVastavaiya]
Position[10], Movie[Heropanti]
Position[11], Movie[Judai]
Position[12], Movie[JodhaAkbar]

--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
3.Iterate From Given Position.
4.Iterate In Reverse Order.
5.Insert At Specified Position.
6.Insert At First And Last Position.
7.Insert At Front.
8.Insert At End.
9.Insert Some Elements at Position.
10.Find First and Last Occurence of Element.
11.Display All Elements with Position.
```

12. Write a Java program to remove a specified element from a linked list.

Source Code :

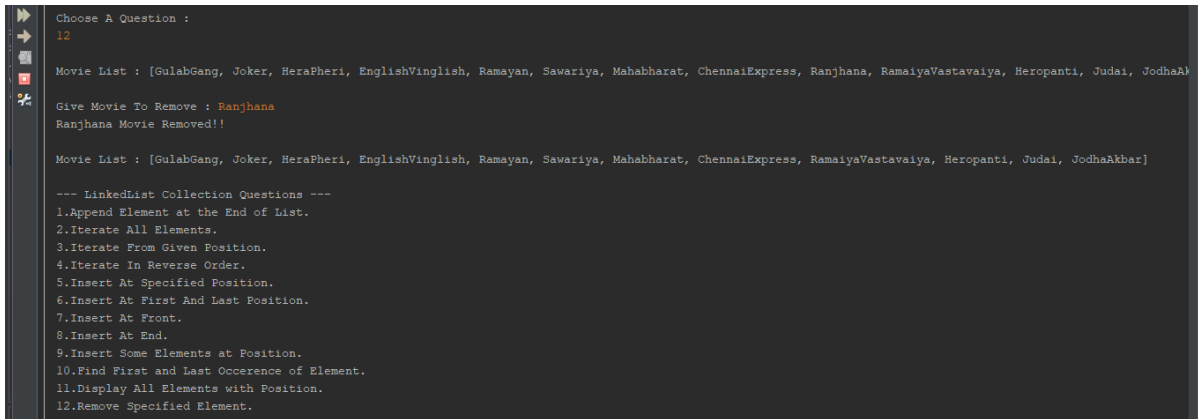
```
protected void RemoveElement(LinkedList<String> Movies) {
    String ele;
    System.out.print("\nMovie List : " + Movies);

    System.out.println();

    if (Movies.size() > 0) {
        System.out.print("\nGive Movie To Remove : ");
        ele = in.next();
        if (Movies.contains(ele)) {
            Movies.remove(ele);
            System.out.println(ele + " Movie Removed!!");
            System.out.print("\nMovie List : " + Movies);
        } else {
            System.out.println(ele + "Movie Not Found!!");
        }
    } else {
        System.out.println("Movie List Empty!!");
    }

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

Output :



```
Choose A Question :
12

Movie List : [GulabGang, Joker, HeraPheri, EnglishVinglish, Ramayan, Sawariya, Mahabharat, ChennaiExpress, Ranjhana, RamaiyaVastavaiya, Heropanti, Judai, JodhaAkbar]

Give Movie To Remove : Ranjhana
Ranjhana Movie Removed!!

Movie List : [GulabGang, Joker, HeraPheri, EnglishVinglish, Ramayan, Sawariya, Mahabharat, ChennaiExpress, RamaiyaVastavaiya, Heropanti, Judai, JodhaAkbar]

--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
3.Iterate From Given Position.
4.Iterate In Reverse Order.
5.Insert At Specified Position.
6.Insert At First And Last Position.
7.Insert At Front.
8.Insert At End.
9.Insert Some Elements at Position.
10.Find First and Last Occurence of Element.
11.Display All Elements with Position.
12.Remove Specified Element.
```

13. Write a Java program to remove the first and last elements from a linked list. Source Code :

```
protected void RemoveFirstAndLastElement(LinkedList<String> Movies) {
    String eleF, eleL;
    System.out.print("\nMovie List : " + Movies);

    System.out.println();

    if (Movies.size() > 0) {
        eleF = Movies.getFirst();
        eleL = Movies.getLast();

        Movies.removeFirst();
        System.out.println("First " + eleF + " Movie Removed!!");

        Movies.removeLast();
        System.out.println("Last " + eleL + " Movie Removed!!");
        System.out.print("\nMovie List : " + Movies);
    } else {
        System.out.println("Movie List Empty!!!");
    }
    System.out.println();
}
```

Output :

```
Choose A Question :
13

Movie List : [GulabGang, Joker, HeraPheri, EnglishVinglish, Ramayan, Sawariya, Mahabharat, ChennaiExpress, RamaiyaVastavaiya, Heropanti, Judai, JodhaAkbar]
First GulabGang Movie Removed!!
Last JodhaAkbar Movie Removed!!

Movie List : [Joker, HeraPheri, EnglishVinglish, Ramayan, Sawariya, Mahabharat, ChennaiExpress, RamaiyaVastavaiya, Heropanti, Judai]

--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
3.Iterate From Given Position.
4.Iterate In Reverse Order.
5.Insert At Specified Position.
6.Insert At First And Last Position.
7.Insert At Front.
8.Insert At End.
9.Insert Some Elements at Position.
10.Find First and Last Occerence of Element.
11.Display All Elements with Position.
12.Remove Specified Element.
13.Remove First And Last.
```

14. Write a Java program to remove all elements from a linked list.

Source Code :

```
protected void RemoveAllElements(LinkedList<String> Movies) {
    System.out.print("\nMovie List : " + Movies);

    System.out.println();

    if (Movies.size() > 0) {
        Movies.removeAll(Movies);
        System.out.println("All Movies Removed!!");
        System.out.print("\nMovie List : " + Movies);
    } else {
        System.out.println("Movie List Empty!!");
    }

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

Output :

```
Choose A Question :
14

Movie List : [Joker, HeraPheri, EnglishVinglish, Ramayan, Sawariya, Mahabharat, ChennaiExpress, RamaiyaVastavaiya, Heropanti, Judai]
All Movies Removed!!

Movie List : []

--- LinkedList Collection Questions ---
1.Append Element at the End of List.
2.Iterate All Elements.
3.Iterate From Given Position.
4.Iterate In Reverse Order.
5.Insert At Specified Position.
6.Insert At First And Last Position.
7.Insert At Front.
8.Insert At End.
9.Insert Some Elements at Position.
10.Find First and Last Occerence of Element.
11.Display All Elements with Position.
12.Remove Specified Element.
13.Remove First And Last.
14.Remove All Elements.
```

15. Write a Java program that swaps two elements in a linked list.

Source Code :

```
protected void SwapTwoElements(LinkedList<String> Movies) {
    String ele1, ele2;
    int i1, i2;
    i1 = 0;
    i2 = 0;

    System.out.print("\nMovie List : " + Movies);

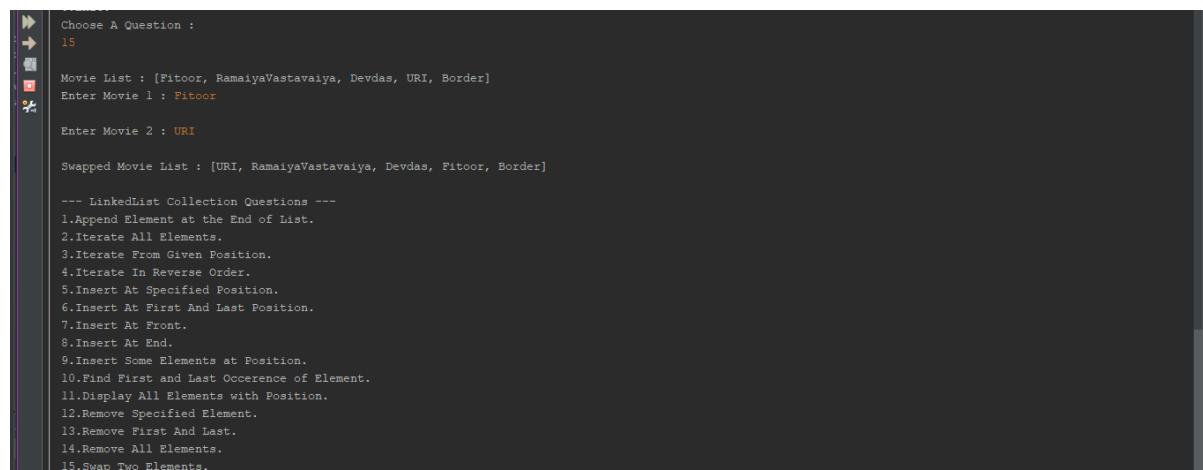
    System.out.print("\nEnter Movie 1 : ");
    ele1 = in.next();
    System.out.print("\nEnter Movie 2 : ");
    ele2 = in.next();

    for (int i = 0; i < Movies.size(); i++) {
        if (Movies.get(i).equalsIgnoreCase(ele1)) {
            i1 = i;
        } else if (Movies.get(i).equalsIgnoreCase(ele2)) {
            i2 = i;
        }
    }

    Collections.swap(Movies, i1, i2);

    System.out.print("\nSwapped Movie List : " + Movies);
    System.out.println();
}
```

Output :



```
Choose A Question :
15

Movie List : [Fittoor, RamaiyaVastavaiya, Devdas, URI, Border]
Enter Movie 1 : Fittoor
Enter Movie 2 : URI

Swapped Movie List : [URI, RamaiyaVastavaiya, Devdas, Fittoor, Border]

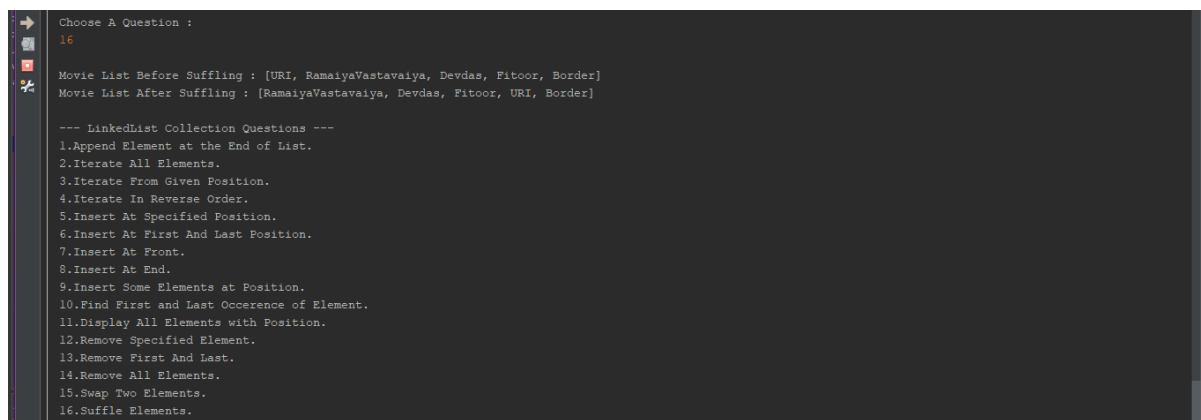
--- LinkedList Collection Questions ---
1. Append Element at the End of List.
2. Iterate All Elements.
3. Iterate From Given Position.
4. Iterate In Reverse Order.
5. Insert At Specified Position.
6. Insert At First And Last Position.
7. Insert At Front.
8. Insert At End.
9. Insert Some Elements at Position.
10. Find First and Last Occerence of Element.
11. Display All Elements with Position.
12. Remove Specified Element.
13. Remove First And Last.
14. Remove All Elements.
15. Swap Two Elements.
```

16. Write a Java program to shuffle elements in a linked list.

Source Code :

```
protected void SuffleElements(LinkedList<String> Movies) {  
    System.out.print("\nMovie List Before Suffling : " + Movies);  
    Collections.shuffle(Movies);  
    System.out.print("\nMovie List After Suffling : " + Movies);  
    System.out.println();  
}
```

Output :



```
Choose A Question :  
16  
  
Movie List Before Suffling : [URI, RamaiyaVastavaiya, Devdas, Fitoor, Border]  
Movie List After Suffling : [RamaiyaVastavaiya, Devdas, Fitoor, URI, Border]  
  
--- LinkedList Collection Questions ---  
1. Append Element at the End of List.  
2. Iterate All Elements.  
3. Iterate From Given Position.  
4. Iterate In Reverse Order.  
5. Insert At Specified Position.  
6. Insert At First And Last Position.  
7. Insert At Front.  
8. Insert At End.  
9. Insert Some Elements at Position.  
10. Find First and Last Occerence of Element.  
11. Display All Elements with Position.  
12. Remove Specified Element.  
13. Remove First And Last.  
14. Remove All Elements.  
15. Swap Two Elements.  
16. Suffle Elements.
```

17. Write a Java program to join two linked lists.

Source Code :

```
protected void JoinTwoLinkedLists(LinkedList<String> Movies) {  
    LinkedList<String> Movies2 = new LinkedList<String>();  
  
    int n;  
    System.out.println("Insert Movies in Second LinkedList");  
  
    System.out.println("How Many Movies You Want to Insert ? ");  
    System.out.print("> ");  
    n = in.nextInt();  
  
    System.out.println("\nEnter Movie Name List : ");  
    for (int i = 0; i < n; i++) {  
        Movies2.add(in.next());  
    }  
  
    System.out.print("\nYour Movie List : " + Movies);  
    System.out.print("\nSecond Movie List : " + Movies2);  
}
```

```

        System.out.println();

        Movies.removeAll(Movies2);
        Movies.addAll(Movies2);

        System.out.print("\nJoined Movies List : " + Movies);
        System.out.println();
    }
}

```

Output :

```

17.Join Two Linked Lists.
18.Copy Linked List To Other List.
19.Remove And Return First Element.
20.Retrieve But Not Return First Element.
21.Retrieve But Not Return Last Element.
22.Check Element Is Exist.
23.Convert Linked List To Array List.
24.Compare Two Linked List.
25.Check List Emptiness.
26.Replace Specified Element.
0.Exit.
Choose A Question :
17
Insert Movies in Second LinkedList
How Many Movies You Want to Insert ?
> 3

Enter Movie Name List :
HeraPheri
Dhamal
Welcome

Your Movie List : [RamaiyaVastavaiya, Devdas, Fitoor, URI, Border]
Second Movie List : [HeraPheri, Dhamal, Welcome]

Joined Movies List : [RamaiyaVastavaiya, Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]

```

18. Write a Java program to copy a linked list to another linked list.

Source Code :

```

protected void CopyLinkedListToAnotherLinkedList(LinkedList<String> Movies) {
    LinkedList<String> CopyCat = new LinkedList<String>();
    System.out.print("\nMovie List : " + Movies);
    System.out.println("\n\nCopying Movie List To Other List...");

    CopyCat.addAll(Movies);

    System.out.println("Successfully Copied Movie List To Other List...");
    System.out.print("\nCoppied Movie List : " + CopyCat);
    System.out.println();
}

```

Output :

```
18.Copy Linked List To Other List.
19.Remove And Return First Element.
20.Retrieve But Not Return First Element.
21.Retrieve But Not Return Last Element.
22.Check Element Is Exist.
23.Convert Linked List To Array List.
24.Compare Two Linked List.
25.Check List Emptiness.
26.Replace Specified Element.
0.Exit.
Choose A Question :
18

Movie List : [RamaiyaVastavaiya, Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]

Copying Movie List To Other List...
Successfully Copied Movie List To Other List...

Copied Movie List : [RamaiyaVastavaiya, Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]
```

19. Write a Java program to remove and return the first element of a linked list.

Source Code :

```
protected void RemoveAndReturnFirst(LinkedList<String> Movies) {
    System.out.print("\nMovie List : " + Movies);
    System.out.println("\nDeletinging First Movie From List...");
    System.out.println("Successfully Deleted "+Movies.removeFirst()+" From List...");
    System.out.print("\nUpdated Movie List : " + Movies);
    System.out.println();
}
```

Output :

```
19.Remove And Return First Element.
20.Retrieve But Not Return First Element.
21.Retrieve But Not Return Last Element.
22.Check Element Is Exist.
23.Convert Linked List To Array List.
24.Compare Two Linked List.
25.Check List Emptiness.
26.Replace Specified Element.
0.Exit.
Choose A Question :
19

Movie List : [RamaiyaVastavaiya, Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]
Deletinging First Movie From List...
Successfully Deleted RamaiyaVastavaiya From List...

Updated Movie List : [Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]
```

20. Write a Java program to retrieve, but not remove, the first element of a linked list.

Source Code :

```
protected void RetriveFirst(LinkedList<String> Movies) {
    System.out.print("\nMovie List : " + Movies);
    System.out.print("\nFirst Movie In List : "+Movies.peekFirst());
    System.out.println();
}
```

Output :



```
20.Retrieve But Not Return First Element.
21.Retrieve But Not Return Last Element.
22.Check Element Is Exist.
23.Convert Linked List To Array List.
24.Compare Two Linked List.
25.Check List Emptiness.
26.Replace Specified Element.
0.Exit.
Choose A Question :
20

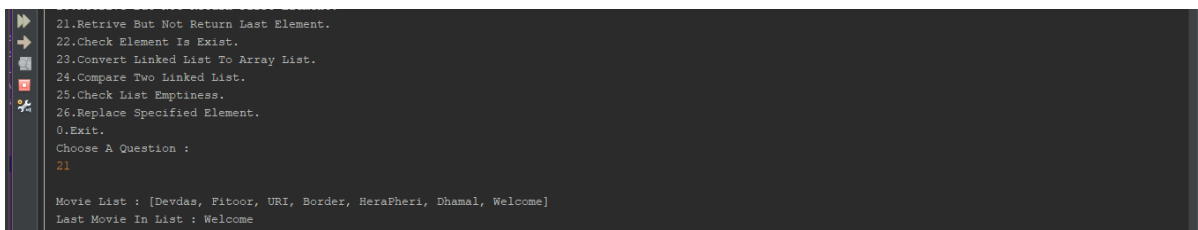
Movie List : [Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]
First Movie In List : Devdas
```

21. Write a Java program to retrieve, but not remove, the last element of a linked list.

Source Code :

```
protected void RetriveLast(LinkedList<String> Movies) {
    System.out.print("\nMovie List : " + Movies);
    System.out.print("\nLast Movie In List : "+Movies.peekLast());
    System.out.println();
}
```

Output :



```
21.Retrieve But Not Return Last Element.
22.Check Element Is Exist.
23.Convert Linked List To Array List.
24.Compare Two Linked List.
25.Check List Emptiness.
26.Replace Specified Element.
0.Exit.
Choose A Question :
21

Movie List : [Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]
Last Movie In List : Welcome
```

22. Write a Java program to check if a particular element exists in a linked list.

Source Code :

```
protected void CheckElement(LinkedList<String> Movies){
    String Movie;

    System.out.print("\nMovie List : "+Movies);
    System.out.print("\nSearch a Movie : ");
    Movie = in.next();
    System.out.println();

    if (Movies.contains(Movie)) {
        System.out.println("Movie " + Movie + " Found At Index [" + Movies.indexOf(Movie) +
            "]\n");
    }
}
```



```

else{
    System.out.println(Movie+" Movie Not Found!!");
}
System.out.println();
}

```

Output :

```

22. Check Element Is Exist.
23. Convert Linked List To Array List.
24. Compare Two Linked List.
25. Check List Emptiness.
26. Replace Specified Element.
0. Exit.
Choose A Question :
22
Movie List : [Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]
Search a Movie : URI
Movie URI Found At Index [2]!!

```

23. Write a Java program to convert a linked list to an array list.

Source Code :

```

protected void ConvertLinkedListToArrayList(LinkedList<String> Movies){
    System.out.print("\nMovie List(LinkedList) : "+Movies);

    ArrayList<String> Movie = new ArrayList<String>(Movies);
    System.out.print("\nMovie List(ArrayList) : "+Movie);
    System.out.println();
}

```

Output :

```

23. Convert Linked List To Array List.
24. Compare Two Linked List.
25. Check List Emptiness.
26. Replace Specified Element.
0. Exit.
Choose A Question :
23
Movie List(LinkedList) : [Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]
Movie List(ArrayList) : [Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]

```

24. Write a Java program to compare two linked lists.

Source Code :

```

protected void CompareLinkedLists(LinkedList<String> Movies){
    LinkedList<String> Movies1 = new LinkedList<String>();
    LinkedList<String> Movies2 = new LinkedList<String>();

    Movies1.add("Pari");
    Movies1.add("1920");
}

```

```

Movies1.add("Manikarnika");
Movies1.add("Mimi");

Movies2.addAll(Movies);

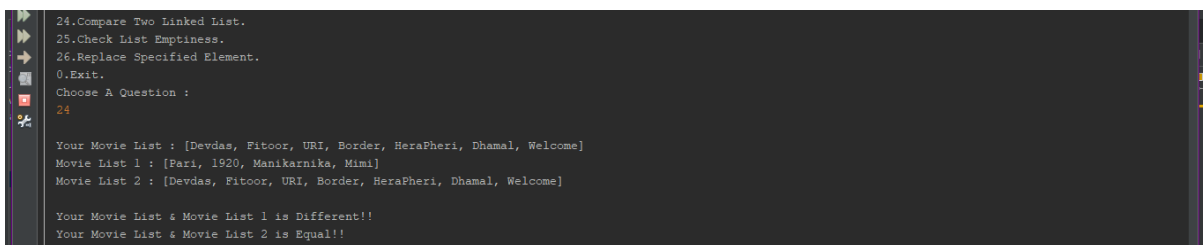
System.out.print("\nYour Movie List : "+Movies);
System.out.print("\nMovie List 1 : "+Movies1);
System.out.print("\nMovie List 2 : "+Movies2);
System.out.println();

if(Movies.equals(Movies1))
{
    System.out.print("\nYour Movie List & Movie List 1 is Equal!!");
}
else
{
    System.out.print("\nYour Movie List & Movie List 1 is Different!!");
}

if(Movies.equals(Movies2))
{
    System.out.print("\nYour Movie List & Movie List 2 is Equal!!");
}
else
{
    System.out.print("\nYour Movie List & Movie List 2 is Different!!");
}
System.out.println();
}

```

Output :



```

24.Compare Two Linked List.
25.Check List Emptiness.
26.Replace Specified Element.
0.Exit.
Choose A Question :
24
Your Movie List : [Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]
Movie List 1 : [Pari, 1920, Manikarnika, Mimi]
Movie List 2 : [Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]

Your Movie List & Movie List 1 is Different!!
Your Movie List & Movie List 2 is Equal!!

```

25. Write a Java program to check if a linked list is empty or not.

Source Code :

```
protected void CheckListEmpty(LinkedList<String> Movies){
    if(Movies.isEmpty())
    {
        System.out.print("\nYour Movie List is Empty!!");
    }
    else
    {
        System.out.print("\nYour Movie List is not Empty, contains : "+Movies);
    }
    System.out.println();
}
```

Output :

A screenshot of a Java IDE window. The title bar shows icons for file operations and a search icon. The main text area contains the following text: "25.Check List Emptiness.", "26.Replace Specified Element.", "0.Exit.", "Choose A Question :", "25", and "Your Movie List is not Empty, contains : [Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]". The text is in a monospaced font, and the background is dark.

26. Write a Java program to replace an element in a linked list.

Source Code :

```
protected void ReplaceElement(LinkedList<String> Movies){
    int i;
    String ele, ele2;
    System.out.print("\nYour Movie List : "+Movies);
    System.out.print("\nGive Movie Name To Replace From List : ");
    ele = in.next();
    if(Movies.contains(ele))
    {
        System.out.print("\nGive Movie Name To Replace : ");
        ele2 = in.next();

        Movies.set(Movies.indexOf(ele), ele2);
        System.out.println("List Updated!!");
        System.out.print("\nUpdated Movie List : "+Movies);
    }
    else
    {
        System.out.println(ele+" Movie Not Found!!");
    }
}
```

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

Output :

```
26.Replace Specified Element.  
0.Exit.  
Choose A Question :  
26  
  
Your Movie List : [Devdas, Fitoor, URI, Border, HeraPheri, Dhamal, Welcome]  
Give Movie Name To Replace From List : Dhamal  
  
Give Movie Name To Replace : Dhol  
List Updated!!  
  
Updated Movie List : [Devdas, Fitoor, URI, Border, HeraPheri, Dhol, Welcome]
```

C) Hash Set :

```
int n;

HashSet<String> Actors = new HashSet<String>();

System.out.println("\nHow Many Actors You Want to Insert ? ");
System.out.print("> ");
n = in.nextInt();

System.out.println("Enter Actor List : ");
for (int i = 0; i < n; i++) {
    Actors.add(in.next());
}
```

Output :



```
How Many Actors You Want to Insert ?
> 5
Enter Actor List :
HarshDingwani
ShrutiSinha
Shantanu
SahajSingh
RohanPal
```

1. Write a Java program to append the specified element to the end of a hash set.

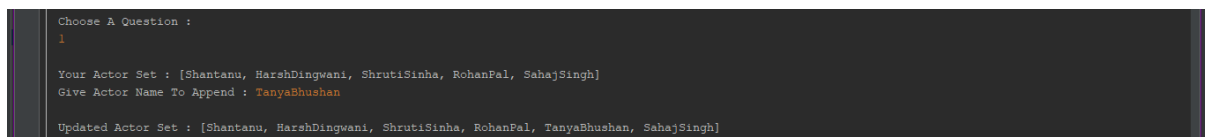
Source Code :

```
protected void AppendElement(HashSet<String> Actors) {
    System.out.print("\nYour Actor Set : " + Actors);

    System.out.print("\nGive Actor Name To Append : ");
    Actors.add(in.next());

    System.out.print("\nUpdated Actor Set : " + Actors);
    System.out.println();
}
```

Output :



```
Choose A Question :
1

Your Actor Set : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, SahajSingh]
Give Actor Name To Append : TanyaBhushan

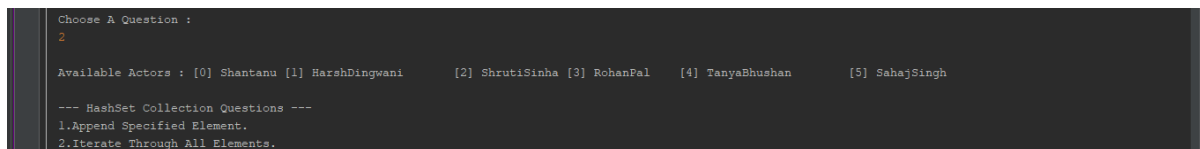
Updated Actor Set : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, TanyaBhushan, SahajSingh]
```

2. Write a Java program to iterate through all elements in a hash list.

Source Code :

```
protected void IterateAllElements(HashSet<String> Actors) {  
    System.out.print("\nAvailable Actors : ");  
    int cnt = 0;  
    for (String a : Actors) {  
        System.out.print "[" + cnt + "]" + a + "\t";  
        cnt++;  
    }  
    System.out.println();  
}
```

Output :



Choose A Question :
2
Available Actors : [0] Shantanu [1] HarshDingwani [2] ShrutiSinha [3] RohanPal [4] TanyaBhushan [5] SahajSingh
--- HashSet Collection Questions ---
1.Append Specified Element.
2.Iterate Through All Elements.

3. Write a Java program to get the number of elements in a hash set.

Source Code :

```
protected void GetNoOfElements(HashSet<String> Actors) {  
    System.out.print("\nNo. Of Actors : "+Actors.size());  
    System.out.println();  
}
```

Output :



Choose A Question :
3
No. Of Actors : 6
--- HashSet Collection Questions ---
1.Append Specified Element.
2.Iterate Through All Elements.
3.Get No. Of Elements.

4. Write a Java program to empty an hash set.

Source Code :

```
protected void MakeListEmpty(HashSet<String> Actors) {  
    System.out.print("\nActor Set : " + Actors);  
  
    System.out.println();  
}
```

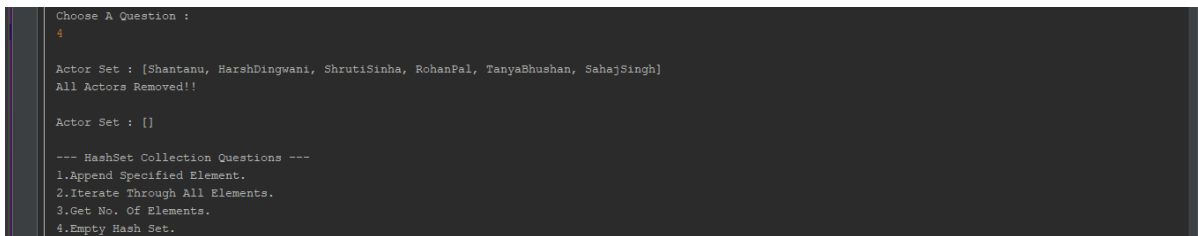
```

    if (Actors.size() > 0) {
        Actors.removeAll(Actors);
        System.out.println("All Actors Removed!!");
        System.out.print("\nActor Set : " + Actors);
    } else {
        System.out.println("Actor Set is Empty!!");
    }

    System.out.println();
}

```

Output :



```

Choose A Question :
4

Actor Set : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, TanyaBhushan, SahajSingh]
All Actors Removed!!

Actor Set : []

--- HashSet Collection Questions ---
1.Append Specified Element.
2.Iterate Through All Elements.
3.Get No. Of Elements.
4.Empty Hash Set.

```

5. Write a Java program to test if a hash set is empty or not.


Source Code :

```

protected void CheckListEmpty(HashSet<String> Actors) {
    if (Actors.isEmpty()) {
        System.out.print("\nYour Actor Set is Empty!!");
    } else {
        System.out.print("\nYour Actor Set is not Empty, contains : " + Actors);
    }
    System.out.println();
}

```

Output :



```

Choose A Question :
5

Your Actor Set is Empty!!

--- HashSet Collection Questions ---
1.Append Specified Element.
2.Iterate Through All Elements.
3.Get No. Of Elements.
4.Empty Hash Set.
5.Check List Emptiness.

```

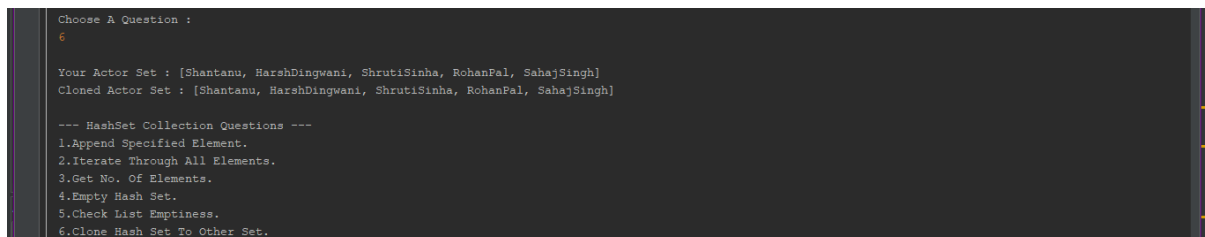
6. Write a Java program to clone a hash set to another hash set.

Source Code :

```
protected void CloneHashSetToOtherSet(HashSet<String> Actors){
    HashSet Actors2 = (HashSet)Actors.clone();

    System.out.print("\nYour Actor Set : "+Actors);
    System.out.print("\nCloned Actor Set : "+Actors2);
    System.out.println();
}
```

Output :



```
Choose A Question :
6

Your Actor Set : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, SahajSingh]
Cloned Actor Set : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, SahajSingh]

--- HashSet Collection Questions ---
1.Append Specified Element.
2.Iterate Through All Elements.
3.Get No. Of Elements.
4.Empty Hash Set.
5.Check List Emptiness.
6.Clone Hash Set To Other Set.
```

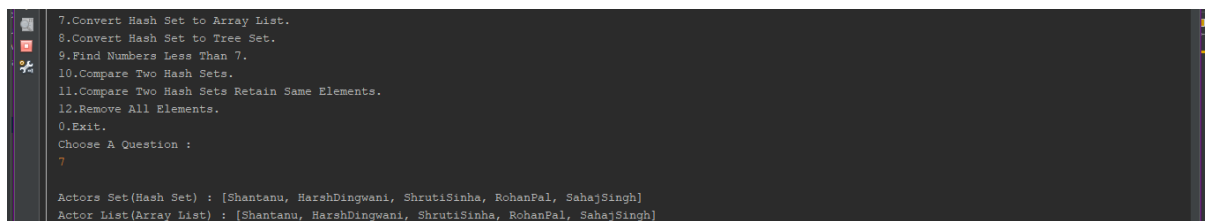
7. Write a Java program to convert a hash set to an array.

Source Code :

```
protected void ConvertHashSetToArrayList(HashSet<String> Actors) {
    System.out.print("\nActors Set(Hash Set) : " + Actors);

    List<String> Actor = new ArrayList<String>(Actors);
    System.out.print("\nActor List(Array List) : " + Actor);
    System.out.println();
}
```

Output :



```
7.Convert Hash Set to Array List.
8.Convert Hash Set to Tree Set.
9.Find Numbers Less Than 7.
10.Compare Two Hash Sets.
11.Compare Two Hash Sets Retain Same Elements.
12.Remove All Elements.
0.Exit.
Choose A Question :
7

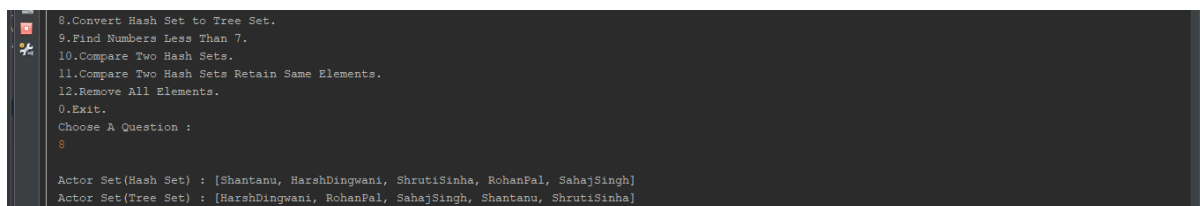
Actors Set(Hash Set) : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, SahajSingh]
Actor List(Array List) : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, SahajSingh]
```


8. Write a Java program to convert a hash set to a tree set.

Source Code :

```
protected void ConvertHashSetToTreeSet(HashSet<String> Actors) {  
    System.out.print("\nActor Set(Hash Set) : " + Actors);  
  
    Set<String> Actor = new TreeSet<String>(Actors);  
    System.out.print("\nActor Set(Tree Set) : " + Actor);  
    System.out.println();  
}
```

Output :



```
8.Convert Hash Set to Tree Set.  
9.Find Numbers Less Than 7.  
10.Compare Two Hash Sets.  
11.Compare Two Hash Sets Retain Same Elements.  
12.Remove All Elements.  
0.Exit.  
Choose A Question :  
8  
  
Actor Set(Hash Set) : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, SahajSingh]  
Actor Set(Tree Set) : [HarshDingwani, RohanPal, SahajSingh, Shantanu, ShrutiSinha]
```

9. Write a Java program to find numbers less than 7 in a tree set.

Source Code :

```
protected void NumbersLess7(){  
    int n;  
  
    HashSet<Integer> Numbers = new HashSet<Integer>();  
  
    System.out.println("\nHow Many Numbers You Want to Insert ? ");  
    System.out.print("> ");  
    n = in.nextInt();  
  
    System.out.println("Enter Integers : ");  
    for(int i=0;i<n;i++){  
        {  
            Numbers.add(in.nextInt());  
        }  
    }  
  
    System.out.print("\nNumbers < 7 : ");  
    for (Integer num : Numbers) {  
        if (num < 7) {  
            System.out.print(num + "\t");  
        }  
    }
```

```
}  
}
```

Output :

A screenshot of a Java IDE with a dark theme. The left sidebar shows a project tree with a file named '9.Find Numbers Less Than 7.'. The main editor area displays the following output:

```
9.Find Numbers Less Than 7.  
10.Compare Two Hash Sets.  
11.Compare Two Hash Sets Retain Same Elements.  
12.Remove All Elements.  
0.Exit.  
Choose A Question :  
9  
  
How Many Numbers You Want to Insert ?  
> 5  
Enter Integers :  
7  
2  
6  
9  
4  
  
Numbers < 7 : 2 4      6
```

10. Write a Java program to compare two hash set.

Source Code :

```
protected void CompareHashSets(HashSet<String> Actors) {  
    HashSet<String> Actors1 = new HashSet<String>();  
    HashSet<String> Actors2 = new HashSet<String>();  
  
    Actors1.add("AkshayKumar");  
    Actors1.add("DeepikaPadukon");  
    Actors1.add("SRK");  
    Actors1.add("HritikRoshan");  
  
    Actors2.addAll(Actors);  
  
    System.out.print("\nYour Actor List : " + Actors);  
    System.out.print("\nActor List 1 : " + Actors1);  
    System.out.print("\nActor List 2 : " + Actors2);  
    System.out.println();  
  
    if (Actors.equals(Actors1)) {  
        System.out.print("\nYour Actor List & Actor List 1 is Equal!!");  
    } else {  
        System.out.print("\nYour Actor List & Actor List 1 is Different!!");  
    }  
  
    if (Actors.equals(Actors2)) {  
        System.out.print("\nYour Actor List & Actor List 2 is Equal!!");  
    } else {  
        System.out.print("\nYour Actor List & Actor List 2 is Different!!");  
    }  
}
```

```

    }
    System.out.println();
}

```

Output :

```

10.Compare Two Hash Sets.
11.Compare Two Hash Sets Retain Same Elements.
12.Remove All Elements.
0.Exit.
Choose A Question :
10

Your Actor List : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, SahajSingh]
Actor List 1 : [HritikRoshan, AkshayKumar, SRK, DeepikaPadukon]
Actor List 2 : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, SahajSingh]

Your Actor List & Actor List 1 is Different!!
Your Actor List & Actor List 2 is Equal!!

```

11. Write a Java program to compare two sets and retain elements that are the same.

Source Code :

```

protected void CompareHashSetRetainSameElements() {
    HashSet<String> Actors = new HashSet<String>();
    HashSet<String> Actors1 = new HashSet<String>();
    HashSet<String> Actors2 = new HashSet<String>();

    System.out.println("\nHow Many Actors You Want to Insert ? ");
    System.out.print("> ");
    int n = in.nextInt();

    System.out.println("Enter Actor List 1 : ");
    for (int i = 0; i < n; i++) {
        Actors1.add(in.next());
    }
    Actors.addAll(Actors1);

    System.out.println("Enter Actor List 2 : ");
    for (int i = 0; i < n; i++) {
        Actors2.add(in.next());
    }

    System.out.print("\nActor List 1 : " + Actors1);
    System.out.print("\nActor List 2 : " + Actors2);
    System.out.println();

    if (Actors1.equals(Actors2)) {
        System.out.println("Your Actor List 1 & Actor List 2 is Equal!!");
    }
}

```

```

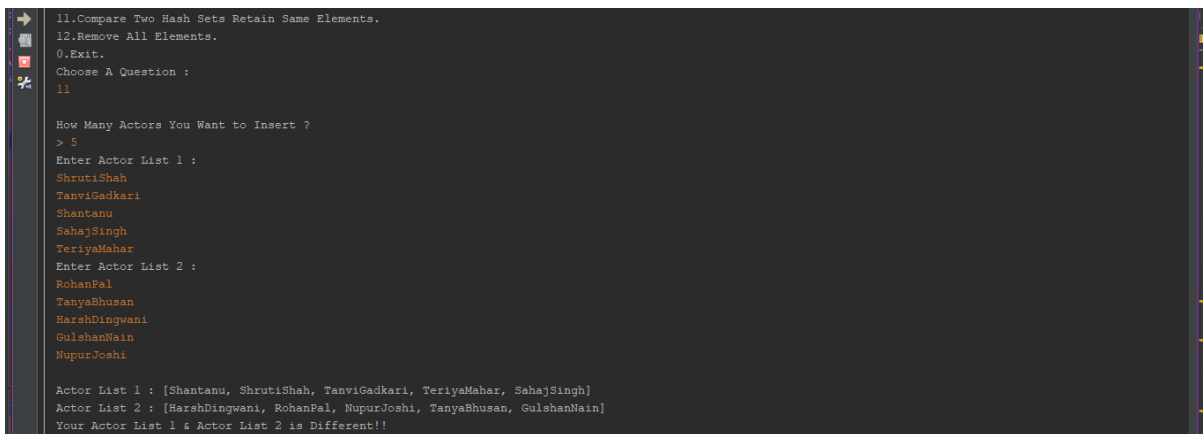
    } else {
        System.out.println("Your Actor List 1 & Actor List 2 is Different!!");
        Actors.retainAll(Actors2);

        if(Actors.size()>0)
        {
            System.out.print("\nSame Elements : "+Actors);
        }
    }

    System.out.println();
}

```

Output :



```

11.Compare Two Hash Sets Retain Same Elements.
12.Remove All Elements.
0.Exit.
Choose A Question :
11

How Many Actors You Want to Insert ?
> 5
Enter Actor List 1 :
ShrutiShah
TanviGadkari
Shantanu
SahajSingh
TeriyaMahar
Enter Actor List 2 :
RohanPal
TanyaBhusan
HarshDingwani
GulshanNain
NupurJoshi

Actor List 1 : [Shantanu, ShrutiShah, TanviGadkari, TeriyaMahar, SahajSingh]
Actor List 2 : [HarshDingwani, RohanPal, NupurJoshi, TanyaBhusan, GulshanNain]
Your Actor List 1 & Actor List 2 is Different!!

```

12. Write a Java program to remove all elements from a hash set.

Source Code :

```

protected void MakeListEmpty(HashSet<String> Actors) {
    System.out.print("\nActor Set : " + Actors);

    System.out.println();

    if (Actors.size() > 0) {
        Actors.removeAll(Actors);
        System.out.println("All Actors Removed!!");
        System.out.print("\nActor Set : " + Actors);
    } else {
        System.out.println("Actor Set is Empty!!");
    }

    System.out.println();
}

```

}

Output :

```
12.Remove All Elements.  
0.Exit.  
Choose A Question :  
12  
  
Actor Set : [Shantanu, HarshDingwani, ShrutiSinha, RohanPal, SahajSingh]  
All Actors Removed!!  
  
Actor Set : []
```

D) Tree Set :

```
TreeSet<String> Temples = new TreeSet<String>();
```

1. Write a Java program to create a tree set, add some colors (strings) and print out the tree set.

Source Code :

```
protected TreeSet<String> CreateInsertPrint(){
    int n;
    TreeSet<String> Temples = new TreeSet<String>();

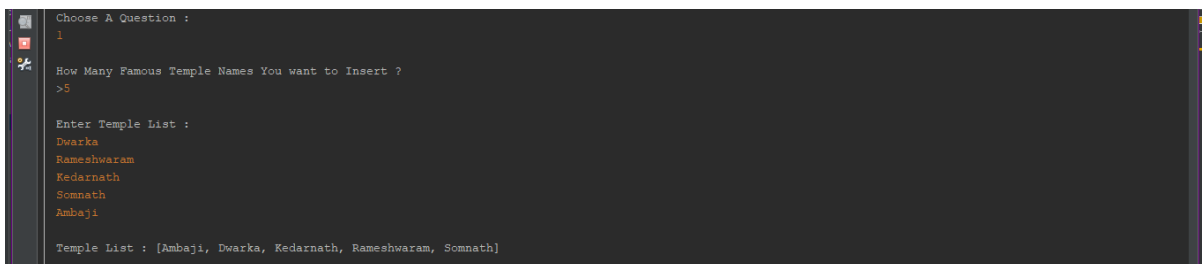
    System.out.println("\nHow Many Famous Temple Names You want to Insert ?");
    System.out.print(">");
    n = in.nextInt();

    System.out.println("\nEnter Temple List : ");
    for (int i = 0; i < n; i++) {
        Temples.add(in.next());
    }

    System.out.print("\nTemple List : "+Temples);
    System.out.println();

    return Temples;
}
```

Output :



```
Choose A Question :
1
How Many Famous Temple Names You want to Insert ?
>5
Enter Temple List :
Dwarka
Rameshwaram
Kedarnath
Somnath
Ambaji
Temple List : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]
```

2. Write a Java program to iterate through all elements in a tree set.

Source Code :

```
protected void IterateAllElements(TreeSet<String> Temples){
    int cnt = 1;
    System.out.println("\nTemple List : ");
```

```

for (String t: Temples) {
    System.out.print("[ "+cnt+" ] "+t+"\t");
    cnt++;
}
System.out.println();
}

```

Output :

```

Choose A Question :
2

Temple List :
[1] Ambaji      [2] Dwarka      [3] Kedarnath   [4] Rameshwaram [5] Somnath

--- ArrayList Collection Questions ---
1.Create, Insert and Print.
2.Iterate All Elements.

```

3. Write a Java program to add all the elements of a specified tree set to another tree set.

Source Code :

```

protected void ReverseElements(TreeSet<String> Temples){
    System.out.print("\nTemple List : "+Temples);
    System.out.println();

    NavigableSet<String> treeset = Temples.descendingSet();
    Iterator<String> templates = treeset.iterator();

    System.out.print("\nReversed Temple List : "+Temples);
    System.out.println();
}

```

Output :

```

Choose A Question :
3

Temple List : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]

Second Temple List : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]

```

4. Write a Java program to create a reverse order view of the elements contained in a given tree set.

Source Code :

```

protected void AddAllElementsToOtherSet(TreeSet<String> Temples){
    System.out.print("\nTemple List : "+Temples);
    System.out.println();
}

```

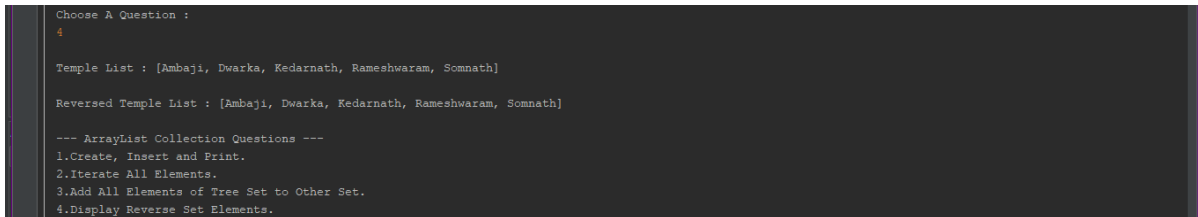
```

Set<String> temples2 = new TreeSet<String>(Temples);

System.out.print("\nSecond Temple List : "+temples2);
System.out.println();
}

```

Output :



```

Choose A Question :
4

Temple List : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]
Reversed Temple List : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]

--- ArrayList Collection Questions ---
1.Create, Insert and Print.
2.Iterate All Elements.
3.Add All Elements of Tree Set to Other Set.
4.Display Reverse Set Elements.

```

5. Write a Java program to get the first and last elements in a tree set.

Source Code :

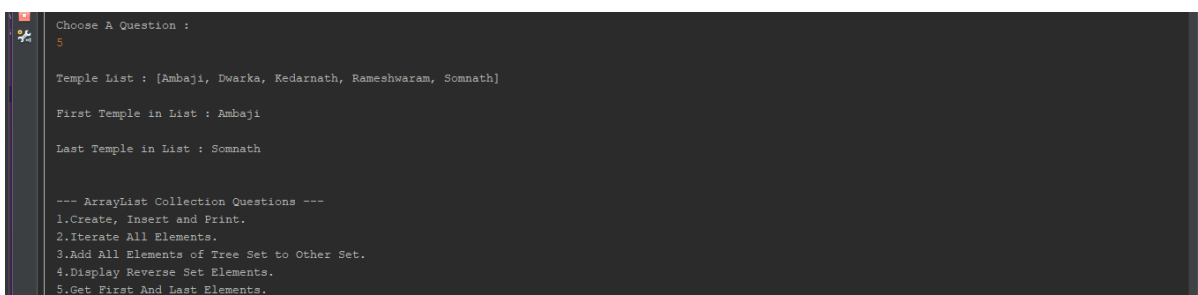
```

protected void GetFirstLastElements(TreeSet<String> Temples){
    System.out.print("\nTemple List : "+Temples);
    System.out.println();

    System.out.println("\nFirst Temple in List : "+Temples.first());
    System.out.println("\nLast Temple in List : "+Temples.last());
    System.out.println();
}

```

Output :



```

Choose A Question :
5

Temple List : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]

First Temple in List : Ambaji
Last Temple in List : Somnath

--- ArrayList Collection Questions ---
1.Create, Insert and Print.
2.Iterate All Elements.
3.Add All Elements of Tree Set to Other Set.
4.Display Reverse Set Elements.
5.Get First And Last Elements.

```

6. Write a Java program to clone a tree set list to another tree set.

Source Code :

```

protected void CloneTreeSetToOtherSet(TreeSet<String> Temples){
    TreeSet<String> Temples2 = new TreeSet<String>();
}

```



```

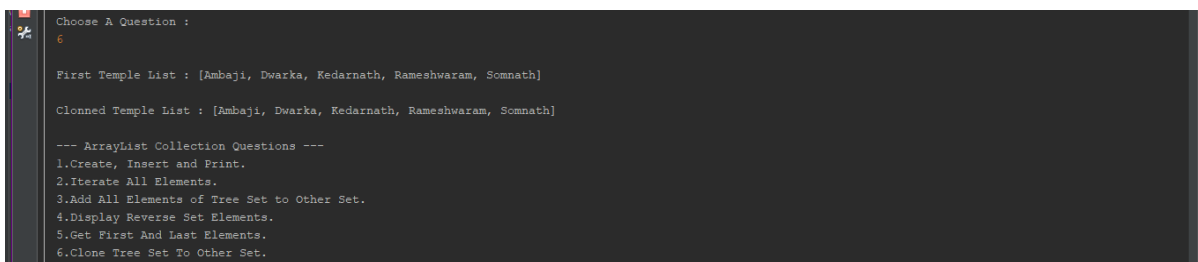
System.out.print("\nFirst Temple List : "+Temples);
System.out.println();

Temples2 = (TreeSet<String>)Temples.clone();

System.out.print("\nClonned Temple List : "+Temples2);
System.out.println();
}

```

Output :



```

Choose A Question :
6

First Temple List : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]
Clonned Temple List : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]

--- ArrayList Collection Questions ---
1.Create, Insert and Print.
2.Iterate All Elements.
3.Add All Elements of Tree Set to Other Set.
4.Display Reverse Set Elements.
5.Get First And Last Elements.
6.Clone Tree Set To Other Set.

```

7. Write a Java program to get the number of elements in a tree set.

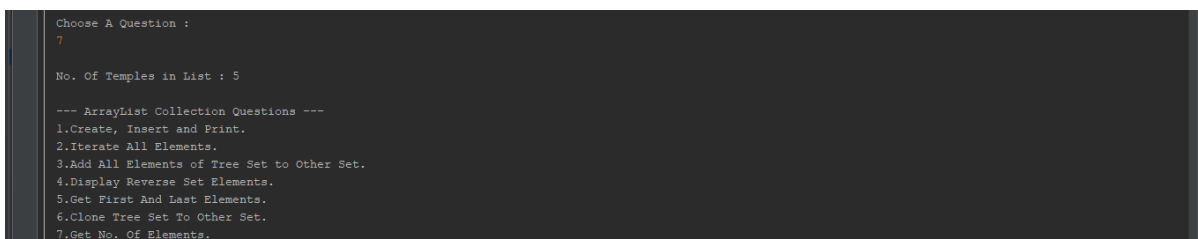
Source Code :

```

protected void GetNumberOfElements(TreeSet<String> Temples){
    System.out.print("\nNo. Of Temples in List : "+Temples.size());
    System.out.println();
}

```

Output :



```

Choose A Question :
7

No. Of Temples in List : 5

--- ArrayList Collection Questions ---
1.Create, Insert and Print.
2.Iterate All Elements.
3.Add All Elements of Tree Set to Other Set.
4.Display Reverse Set Elements.
5.Get First And Last Elements.
6.Clone Tree Set To Other Set.
7.Get No. Of Elements.

```

8. Write a Java program to compare two tree sets.

Source Code :

```

protected void CompareTwoTreeSets(TreeSet<String> Temples){
    TreeSet<String> Temples1 = new TreeSet<String>();
    TreeSet<String> Temples2 = new TreeSet<String>();
}

```

```

Temples1.add("Himalay");
Temples1.add("Minaxi(Madurai)");
Temples1.add("Rameshwaram");
Temples1.add("Kashi-Vishwanath");

Temples2.addAll(Temples);

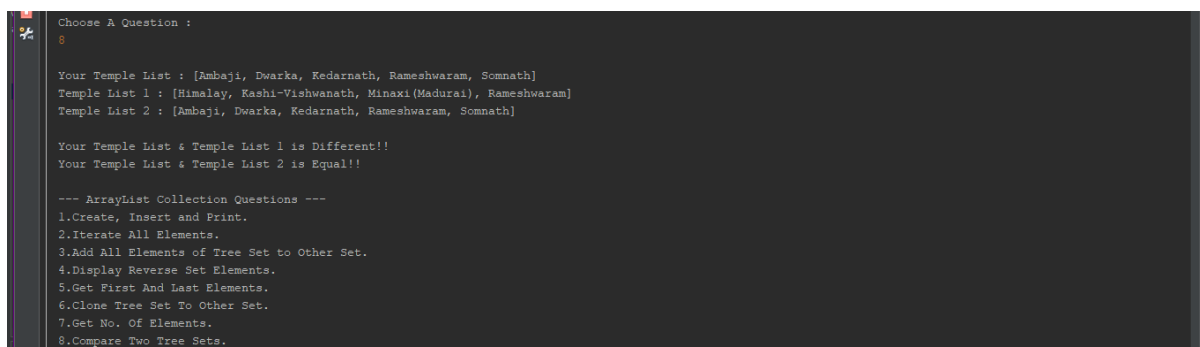
System.out.print("\nYour Temple List : "+Temples);
System.out.print("\nTemple List 1 : "+Temples1);
System.out.print("\nTemple List 2 : "+Temples2);
System.out.println();

if(Temples.equals(Temples1))
{
    System.out.print("\nYour Temple List & Temple List 1 is Equal!!");
}
else
{
    System.out.print("\nYour Temple List & Temple List 1 is Different!!");
}

if(Temples.equals(Temples2))
{
    System.out.print("\nYour Temple List & Temple List 2 is Equal!!");
}
else
{
    System.out.print("\nYour Temple List & Temple List 2 is Different!!");
}
System.out.println();
}

```

Output :



```

Choose A Question :
8

Your Temple List : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]
Temple List 1 : [Himalay, Kashi-Vishwanath, Minaxi(Madurai), Rameshwaram]
Temple List 2 : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]

Your Temple List & Temple List 1 is Different!!
Your Temple List & Temple List 2 is Equal!!

--- ArrayList Collection Questions ---
1.Create, Insert and Print.
2.Iterate All Elements.
3.Add All Elements of Tree Set to Other Set.
4.Display Reverse Set Elements.
5.Get First And Last Elements.
6.Clone Tree Set To Other Set.
7.Get No. Of Elements.
8.Compare Two Tree Sets.

```

9. Write a Java program to find numbers less than 7 in a tree set.

Source Code :

```
protected void NumbersLess7(){
    int n;

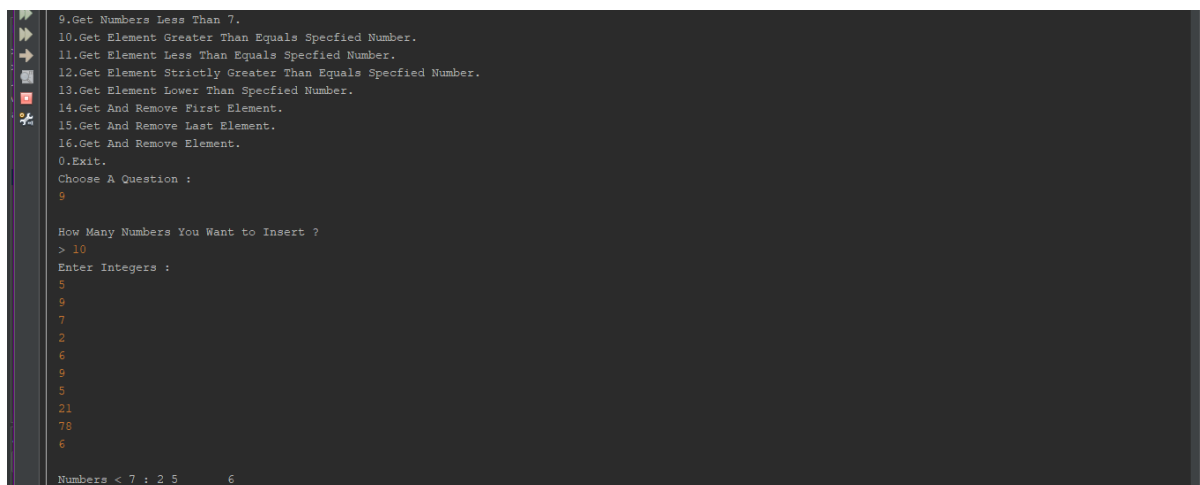
    TreeSet<Integer> Numbers = new TreeSet<Integer>();

    System.out.println("\nHow Many Numbers You Want to Insert ? ");
    System.out.print("> ");
    n = in.nextInt();

    System.out.println("Enter Integers : ");
    for(int i=0;i<n;i++)
    {
        Numbers.add(in.nextInt());
    }

    System.out.print("\nNumbers < 7 : ");
    for (Integer num : Numbers) {
        if (num < 7) {
            System.out.print(num + "\t");
        }
    }
}
```

Output :



```
9. Get Numbers Less Than 7.
10. Get Element Greater Than Equals Specified Number.
11. Get Element Less Than Equals Specified Number.
12. Get Element Strictly Greater Than Equals Specified Number.
13. Get Element Lower Than Specified Number.
14. Get And Remove First Element.
15. Get And Remove Last Element.
16. Get And Remove Element.
0. Exit.
Choose A Question :
9

How Many Numbers You Want to Insert ?
> 10
Enter Integers :
5
9
7
2
6
9
5
21
78
6

Numbers < 7 : 2 5 6
```

10. Write a Java program to get the element in a tree set which is greater than or equal to the given element.

Source Code :

```
protected void GetGreaterEqual(){
    int n, no;

    TreeSet<Integer> Numbers = new TreeSet<Integer>();

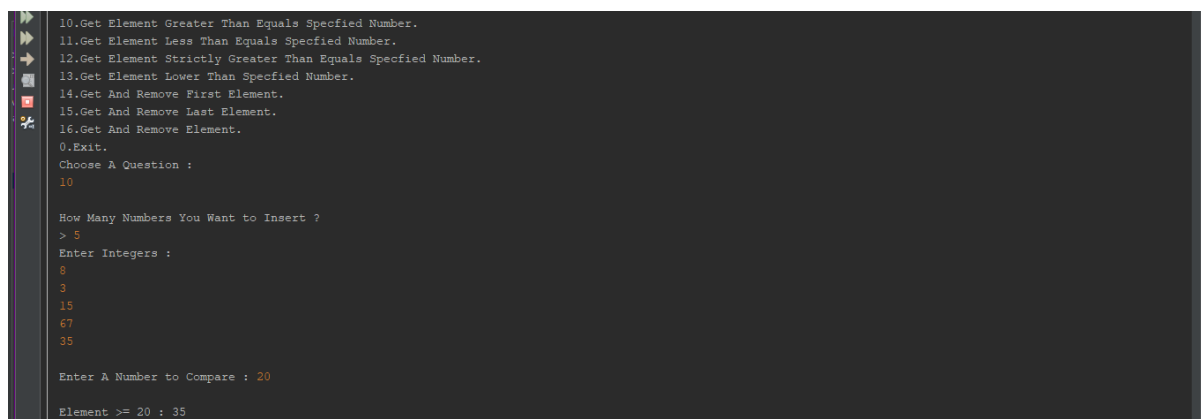
    System.out.println("\nHow Many Numbers You Want to Insert ? ");
    System.out.print("> ");
    n = in.nextInt();

    System.out.println("Enter Integers : ");
    for(int i=0;i<n;i++)
    {
        Numbers.add(in.nextInt());
    }

    System.out.print("\nEnter A Number to Compare : ");
    no = in.nextInt();

    System.out.print("\nElement >= "+no+" : "+Numbers.ceiling(no));
    System.out.println();
}
```

Output :



```
10.Get Element Greater Than Equals Specified Number.
11.Get Element Less Than Equals Specified Number.
12.Get Element Strictly Greater Than Equals Specified Number.
13.Get Element Lower Than Specified Number.
14.Get And Remove First Element.
15.Get And Remove Last Element.
16.Get And Remove Element.
0.Exit.
Choose A Question :
10

How Many Numbers You Want to Insert ?
> 5
Enter Integers :
8
3
15
67
35

Enter A Number to Compare : 20
Element >= 20 : 35
```

11. Write a Java program to get the element in a tree set less than or equal to the given element.

Source Code :

```
protected void GetLessEqual(){
    int n, no;

    TreeSet<Integer> Numbers = new TreeSet<Integer>();

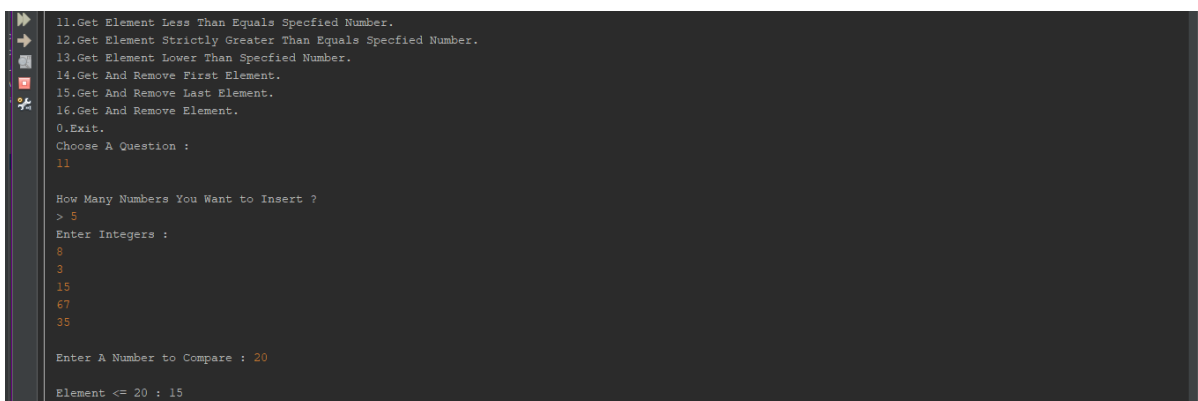
    System.out.println("\nHow Many Numbers You Want to Insert ? ");
    System.out.print("> ");
    n = in.nextInt();

    System.out.println("Enter Integers : ");
    for(int i=0;i<n;i++)
    {
        Numbers.add(in.nextInt());
    }

    System.out.print("\nEnter A Number to Compare : ");
    no = in.nextInt();

    System.out.print("\nElement <= "+no+" : "+Numbers.floor(no));
    System.out.println();
}
```

Output :



```
11.Get Element Less Than Equals Specified Number.
12.Get Element Strictly Greater Than Equals Specified Number.
13.Get Element Lower Than Specified Number.
14.Get And Remove First Element.
15.Get And Remove Last Element.
16.Get And Remove Element.
0.Exit.
Choose A Question :
11

How Many Numbers You Want to Insert ?
> 5
Enter Integers :
8
3
15
67
35

Enter A Number to Compare : 20

Element <= 20 : 15
```

12. Write a Java program to get the element in a tree set strictly greater than or equal to the given element.

Source Code :

```
protected void GetStrictlyGreaterEqual(){
    int n, no;

    TreeSet<Integer> Numbers = new TreeSet<Integer>();

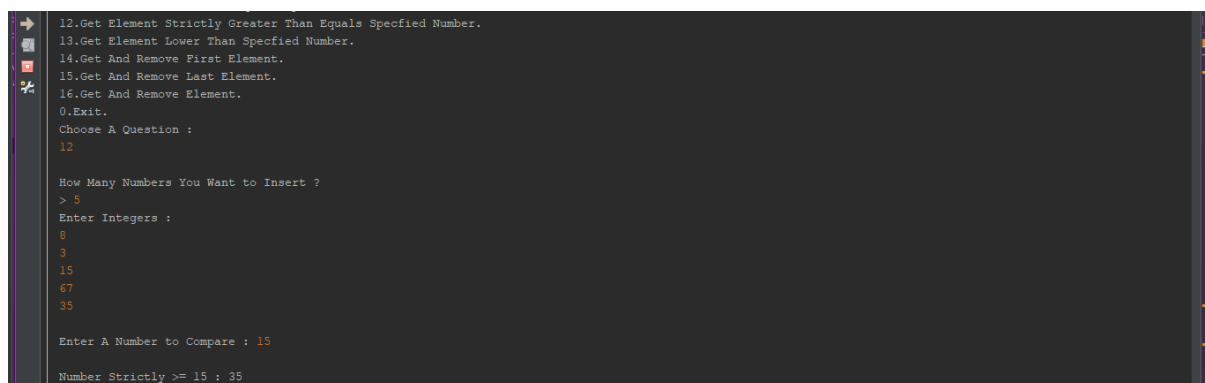
    System.out.println("\nHow Many Numbers You Want to Insert ? ");
    System.out.print("> ");
    n = in.nextInt();

    System.out.println("Enter Integers : ");
    for(int i=0;i<n;i++)
    {
        Numbers.add(in.nextInt());
    }

    System.out.print("\nEnter A Number to Compare : ");
    no = in.nextInt();

    System.out.print("\nNumber Strictly >= "+no+" : "+Numbers.higher(no));
    System.out.println();
}
```

Output :



```
12.Get Element Strictly Greater Than Equals Specified Number.
13.Get Element Lower Than Specified Number.
14.Get And Remove First Element.
15.Get And Remove Last Element.
16.Get And Remove Element.
0.Exit.
Choose A Question :
12

How Many Numbers You Want to Insert ?
> 5
Enter Integers :
8
3
15
67
35

Enter A Number to Compare : 15
Number Strictly >= 15 : 35
```

13. Write a Java program to get an element in a tree set that has a lower value than the given element.

Source Code :

```
protected void GetLower(){
    int n, no;

    TreeSet<Integer> Numbers = new TreeSet<Integer>();

    System.out.println("\nHow Many Numbers You Want to Insert ? ");
    System.out.print("> ");
    n = in.nextInt();

    System.out.println("Enter Integers : ");
    for(int i=0;i<n;i++)
    {
        Numbers.add(in.nextInt());
    }

    System.out.print("\nEnter A Number to Compare : ");
    no = in.nextInt();

    System.out.print("\nElement Lower Than "+no+" : "+Numbers.lower(no));
    System.out.println();
}
```

Output :



```
13.Get Element Lower Than Specified Number.
14.Get And Remove First Element.
15.Get And Remove Last Element.
16.Get And Remove Element.
0.Exit.
Choose A Question :
13

How Many Numbers You Want to Insert ?
> 8
Enter Integers :
3
15
67
35
8
2
20
78

Enter A Number to Compare : 50

Element Lower Than 50 : 35
```

14. Write a Java program to retrieve and remove the first element of a tree set.

Source Code :

```
protected void GetAndRemoveFirst(TreeSet<String> Temples){
    System.out.print("\nTemple List : "+Temples);
    System.out.print("\nGive Temple You Want TO Remove : ");
    System.out.println("\n"+Temples.pollFirst()+" Temple Removed!!");
    System.out.println();
}
```

Output :



```
14.Get And Remove First Element.
15.Get And Remove Last Element.
16.Get And Remove Element.
0.Exit.
Choose A Question :
14

Temple List : [Ambaji, Dwarka, Kedarnath, Rameshwaram, Somnath]
Give Temple You Want TO Remove :
Ambaji Temple Removed!!
```

15. Write a Java program to retrieve and remove the last element of a tree set.

Source Code :

```
protected void GetAndRemoveLast(TreeSet<String> Temples){
    System.out.print("\nTemple List : "+Temples);
    System.out.println("\n"+Temples.pollLast()+" Temple Removed!!");
    System.out.println();
}
```

Output :



```
15.Get And Remove Last Element.
16.Get And Remove Element.
0.Exit.
Choose A Question :
15

Temple List : [Dwarka, Kedarnath, Rameshwaram, Somnath]
Somnath Temple Removed!!
```

16. Write a Java program to remove a given element from a tree set.

Source Code :

```
protected void RemoveElement(TreeSet<String> Temples){
    String ele;
    System.out.print("\nTemple List : "+Temples);
    System.out.print("\nGive Temple You Want To Remove : ");
    ele = in.next();
}
```



```
    Temples.remove(ele);  
    System.out.println("\n"+ele+" Temple Removed!!");  
    System.out.println();  
}
```

Output :

```
16.Get And Remove Element.  
0.Exit.  
Choose A Question :  
16  
  
Temple List : [Dwarka, Kedarnath, Rameshwaram]  
Give Temple You Want To Remove : Rameshwaram  
  
Rameshwaram Temple Removed!!
```

E) Priority Queue :

```
PriorityQueue<String> Shows = new PriorityQueue<String>();
```

1. Write a Java program to create a priority queue, add some colors (strings) and print out the elements of the priority queue.

Source Code :

```
protected PriorityQueue<String> CreateInsertPrint(){
    int n;
    PriorityQueue<String> Shows = new PriorityQueue<String>();

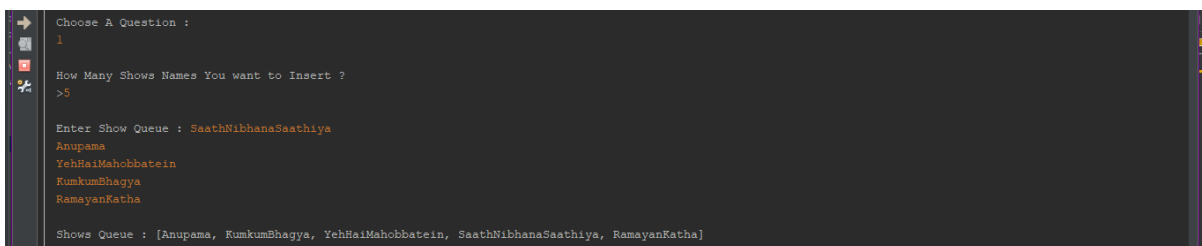
    System.out.println("\nHow Many Shows Names You want to Insert ?");
    System.out.print(">");
    n = in.nextInt();

    System.out.print("\nEnter Show Queue : ");
    for (int i = 0; i < n; i++) {
        Shows.add(in.next());
    }

    System.out.print("\nShows Queue : "+Shows);
    System.out.println();

    return Shows;
}
```

Output :



```
Choose A Question :
1
How Many Shows Names You want to Insert ?
>5

Enter Show Queue : SaathNibhanaSaathiya
Anupama
YehHaiMahabbatein
KumkumBhagya
RamayanKatha

Shows Queue : [Anupama, KumkumBhagya, YehHaiMahabbatein, SaathNibhanaSaathiya, RamayanKatha]
```

2. Write a Java program to iterate through all elements in the priority queue.

Source Code :

```
protected void IterateAllElements(PriorityQueue<String> Shows){
    int cnt = 1;
    System.out.println("\nShows Queue : ");
```

```

for (String s: Shows) {
    System.out.print("[ "+cnt+" ] "+s+"\t");
    cnt++;
}
System.out.println();
}

```

Output :

```

Choose A Question :
2
Shows Queue :
[1] Anupama      [2] KumkumBhagya      [3] YehHaiMahobbatein  [4] SaathNibhanaSaathiya  [5] RamayanKatha

--- Priority Queue Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.

```

3. Write a Java program to add all the elements of a priority queue to another priority queue.

Source Code :

```

protected void AllElementsToOtherQueue(PriorityQueue<String> Shows){
    PriorityQueue<String> Shows2 = new PriorityQueue<String>();
    System.out.print("\nFirst Shows Queue : "+Shows);
    Shows2.addAll(Shows);
    System.out.print("\nSecond Shows Queue : "+Shows2);
    System.out.println();
}

```

Output :

```

Choose A Question :
3
First Shows Queue : [Anupama, KumkumBhagya, YehHaiMahobbatein, SaathNibhanaSaathiya, RamayanKatha]
Second Shows Queue : [Anupama, KumkumBhagya, YehHaiMahobbatein, SaathNibhanaSaathiya, RamayanKatha]

--- Priority Queue Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Add All Elements Of Priority Queue to Other Queue.

```

4. Write a Java program to insert a given element into a priority queue.

Source Code :

```

protected void InsertAnElement(PriorityQueue<String> Shows){
    String show;
    System.out.print("\nGive Show Name : ");
    show = in.next();

    System.out.println(show+" Added To Queue!!");
}

```

```

Shows.add(show);
System.out.print("\nShows Queue : "+Shows);
System.out.println();
}

```

Output :

```

Choose A Question :
4

Give Show Name : Kasauti
Kasauti Added To Queue!!

Shows Queue : [Anupama, KunkumBhagya, Kasauti, SaathNibhanaSaathiya, RamayanKatha, YehHaiMahabbatein]

--- Priority Queue Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Add All Elements Of Priority Queue to Other Queue.
4.Add Element to Queue.

```

5. Write a Java program to remove all elements from a priority queue.

Source Code :

```

protected void RemoveAllElements(PriorityQueue<String> Shows){
    Shows.removeAll(Shows);
    System.out.println("All Shows Removed!!");
    System.out.print("\nShows Queue : "+Shows);
    System.out.println();
}

```

Output :

```

Choose A Question :
5
All Shows Removed!!

Shows Queue : []

--- Priority Queue Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Add All Elements Of Priority Queue to Other Queue.
4.Add Element to Queue.
5.Remove All Elements.

```

6. Write a Java program to count the number of elements in a priority queue.

Source Code :

```

protected void CountAllElements(PriorityQueue<String> Shows){
    System.out.print("\nShows Queue : "+Shows);
    System.out.print("\nNo. Of Shows : "+Shows.size());
    System.out.println();
}

```

Output :

```
Choose A Question :
6

Shows Queue : [Anupama, KumkumBhagya, Kasauti, SaathNibhanaSaathiya, RamayanKatha, YehHaiMahabbatein]
No. Of Shows : 6

--- Priority Queue Collection Questions ---
1.Insert and Print.
2.Iterate All Elements.
3.Add All Elements Of Priority Queue to Other Queue.
4.Add Element to Queue.
5.Remove All Elements.
6.Count All Elements.
```

7. Write a Java program to compare two priority queues.

Source Code :

```
protected void CompareTwoQueues(PriorityQueue<String> Shows){
    PriorityQueue<String> Shows1 = new PriorityQueue<String>();
    PriorityQueue<String> Shows2 = new PriorityQueue<String>();

    Shows1.add("Ramayan");
    Shows1.add("Mahabharat");
    Shows1.add("Ganesh");
    Shows1.add("Laxmi-Narayan");

    Shows2.addAll(Shows);

    System.out.print("\nYour Show Queue : "+Shows);
    System.out.print("\nShow List 1 : "+Shows1);
    System.out.print("\nShow List 2 : "+Shows2);
    System.out.println();

    if(Shows.equals(Shows1))
    {
        System.out.print("\nYour Show Queue & Show Queue 1 is Equal!!");
    }
    else
    {
        System.out.print("\nYour Show Queue & Show Queue 1 is Different!!");
    }

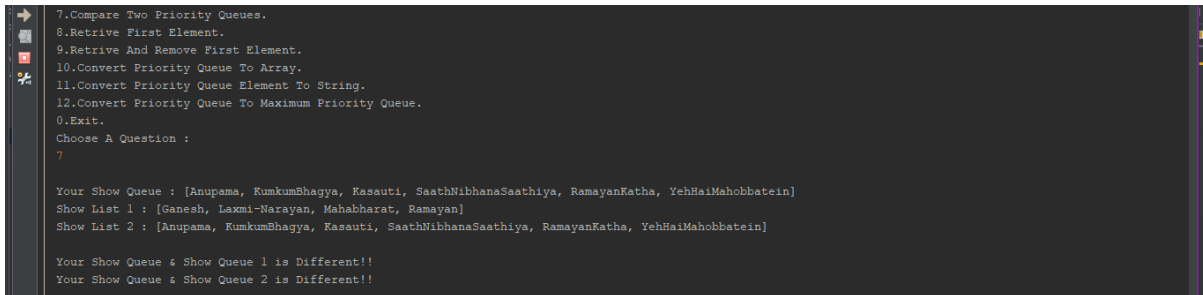
    if(Shows.equals(Shows2))
    {
        System.out.print("\nYour Show Queue & Show Queue 2 is Equal!!");
    }
    else
    {
```

```

        System.out.print("\nYour Show Queue & Show Queue 2 is Different!!");
    }
    System.out.println();
}

```

Output :



```

7.Compare Two Priority Queues.
8.Retrieve First Element.
9.Retrieve And Remove First Element.
10.Convert Priority Queue To Array.
11.Convert Priority Queue Element To String.
12.Convert Priority Queue To Maximum Priority Queue.
0.Exit.
Choose A Question :
7

Your Show Queue : [Anupama, KunkumBhagya, Kasauti, SaathNibhanaSaathiya, RamayanKatha, YehHaiMahabbatein]
Show List 1 : [Ganesh, Laxmi-Narayan, Mahabharat, Ramayan]
Show List 2 : [Anupama, KunkumBhagya, Kasauti, SaathNibhanaSaathiya, RamayanKatha, YehHaiMahabbatein]

Your Show Queue & Show Queue 1 is Different!!
Your Show Queue & Show Queue 2 is Different!!

```

8. Write a Java program to retrieve the first element of the priority queue.

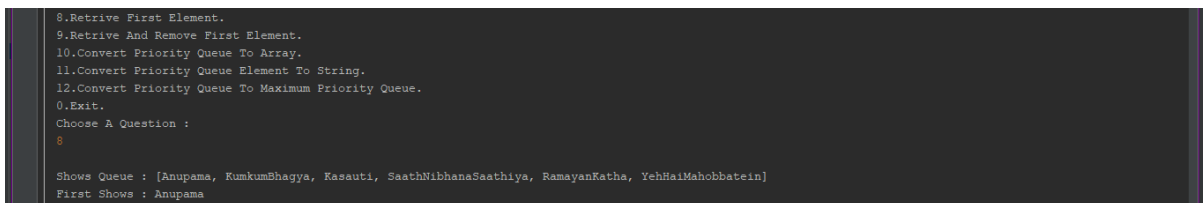
Source Code :

```

protected void RetriveFirstElement(PriorityQueue<String> Shows){
    System.out.print("\nShows Queue : "+Shows);
    System.out.print("\nFirst Shows : "+Shows.element());
    System.out.println();
}

```

Output :



```

8.Retrieve First Element.
9.Retrieve And Remove First Element.
10.Convert Priority Queue To Array.
11.Convert Priority Queue Element To String.
12.Convert Priority Queue To Maximum Priority Queue.
0.Exit.
Choose A Question :
8

Shows Queue : [Anupama, KunkumBhagya, Kasauti, SaathNibhanaSaathiya, RamayanKatha, YehHaiMahabbatein]
First Shows : Anupama

```

9. Write a Java program to retrieve and remove the first element.

Source Code :

```

protected void RetriveRemoveFirstElement(PriorityQueue<String> Shows){
    System.out.print("\nShows Queue : "+Shows);
    System.out.print("\nFirst Shows "+Shows.remove()+" Removed!!");
    System.out.print("\nShows Queue : "+Shows);
    System.out.println();
}

```

Output :

```
9.Retrieve And Remove First Element.
10.Convert Priority Queue To Array.
11.Convert Priority Queue Element To String.
12.Convert Priority Queue To Maximum Priority Queue.
0.Exit.
Choose A Question :
9

Shows Queue : [Anupama, KunkumBhagya, Kasauti, SaathNibhanaSaathiya, RamayanKatha, YehHaiMahobbatein]
First Shows Anupama Removed!!
Shows Queue : [Kasauti, KunkumBhagya, YehHaiMahobbatein, SaathNibhanaSaathiya, RamayanKatha]
```

10. Write a Java program to convert a priority queue to an array containing all its elements.

Source Code :

```
protected void ConvertPriorityQueueToArray(PriorityQueue<String> Shows){
    System.out.print("\nShows Queue : "+Shows);
    Object[] Shows2 = Shows.toArray();
    System.out.print("\nShows Array : ");
    for(int i=0;i<Shows2.length;i++){
        System.out.print(Shows2[i]+"t");
    }
    System.out.println();
}
```

Output :

```
10.Convert Priority Queue To Array.
11.Convert Priority Queue Element To String.
12.Convert Priority Queue To Maximum Priority Queue.
0.Exit.
Choose A Question :
10

Shows Queue : [Kasauti, KunkumBhagya, YehHaiMahobbatein, SaathNibhanaSaathiya, RamayanKatha]
Shows Array : Kasauti KunkumBhagya YehHaiMahobbatein SaathNibhanaSaathiya RamayanKatha
```

11. Write a Java program to convert a Priority Queue element to string representations.

Source Code :

```
protected void ConvertPriorityQueueElementToString(PriorityQueue<String> Shows){
    System.out.print("\nShows Queue : "+Shows);

    System.out.print("\nShows Queue : "+Shows.toString());
}
```

Output :

```
11.Convert Priority Queue Element To String.
12.Convert Priority Queue To Maximum Priority Queue.
0.Exit.
Choose A Question :
11

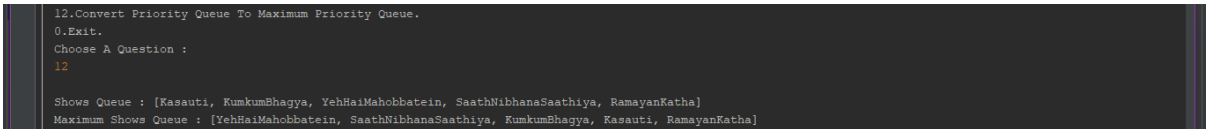
Shows Queue : [Kasauti, KunkumBhagya, YehHaiMahobbatein, SaathNibhanaSaathiya, RamayanKatha]
Shows Queue : [Kasauti, KunkumBhagya, YehHaiMahobbatein, SaathNibhanaSaathiya, RamayanKatha]
```

12. Write a Java program to change priorityQueue to maximum priority queue.

Source Code :

```
protected void ConvertPriorityQueueToMaximumPriorityQueue(PriorityQueue<String> Shows){  
    System.out.print("\nShows Queue : "+Shows);  
  
    PriorityQueue<String> Shows2 = new PriorityQueue<String>(Collections.reverseOrder());  
    Shows2.addAll(Shows);  
    System.out.print("\nMaximum Shows Queue : "+Shows2);  
}
```

Output :

A screenshot of a Java program's output in a dark-themed terminal. The output shows the program title '12.Convert Priority Queue To Maximum Priority Queue.', followed by '0.Exit.' and a prompt 'Choose A Question :'. The user has entered '12'. The program then displays two lines: 'Shows Queue : [Kasauti, KumkumBhagya, YehHaiMahabbatein, SaathNibhanaSaathiya, RamayanKatha]' and 'Maximum Shows Queue : [YehHaiMahabbatein, SaathNibhanaSaathiya, KumkumBhagya, Kasauti, RamayanKatha]'.

```
12.Convert Priority Queue To Maximum Priority Queue.  
0.Exit.  
Choose A Question :  
12  
  
Shows Queue : [Kasauti, KumkumBhagya, YehHaiMahabbatein, SaathNibhanaSaathiya, RamayanKatha]  
Maximum Shows Queue : [YehHaiMahabbatein, SaathNibhanaSaathiya, KumkumBhagya, Kasauti, RamayanKatha]
```


F) Hash Map :

```
HashMap<String, String> Employee = new HashMap<String, String>();
```

1. Write a Java program to associate the specified value with the specified key in a HashMap.

Source Code :

```
protected void AssociateKeyValues(HashMap<String, String> Employee){
    String Key, Value;

    System.out.println("\nHow Many Employees You Want to Enter ?");
    System.out.print(">");
    int n = in.nextInt();

    for(int i=0;i<n;i++)
    {
        System.out.print("\nEnter Employee["+i+"] Name : ");
        Key = in.next();
        System.out.print("\nEnter Employee["+i+"] Designation : ");
        Value = in.next();

        Employee.put(Key, Value);
        System.out.println();
    }

    System.out.println("Employee List : ");
    for(String K:Employee.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Designation : "+Employee.get(K));
    }
    System.out.println();
}
```

Output :

```
Choose A Question :
1
How Many Employees You Want to Enter ?
>5
Enter Employee[0] Name : FenilShah
Enter Employee[0] Designation : Developer

Enter Employee[1] Name : ChetanMukharjee
Enter Employee[1] Designation : Manager

Enter Employee[2] Name : RaveenaRaut
Enter Employee[2] Designation : WebDesigner

Enter Employee[3] Name : YasserDesai
Enter Employee[3] Designation : ProjectManager

Enter Employee[4] Name : TiaPanday
Enter Employee[4] Designation : CEO

Employee List :
Name : TiaPanday
Designation : CEO
Name : YasserDesai
Designation : ProjectManager
Name : RaveenaRaut
Designation : WebDesigner
Name : ChetanMukharjee
Designation : Manager
Name : FenilShah
Designation : Developer
```

2. Write a Java program to count the number of key-value (size) mappings in a map.

Source Code :

```
protected void CountMappings(HashMap<String, String> Employee){

    System.out.println("No. Of Employee : "+Employee.size());
    System.out.println();
}
```

Output :

```
Choose A Question :
2
No. Of Employee : 5

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Count Key-Value Mappings In Map.
```

3. Write a Java program to copy all mappings from the specified map to another map.

Source Code :

```
protected void CopyOneHashMapToOtherMap(HashMap<String, String> Employee){
    System.out.println("Your Employee List : ");
    for(String K:Employee.keySet())
```

```

{
    System.out.println("Name : "+K.toString());
    System.out.println("Designation : "+Employee.get(K));
}

HashMap<String, String> Employee2 = new HashMap<String, String>(Employee);

System.out.println("\nCoppied Employee List : ");
for(String K:Employee2.keySet())
{
    System.out.println("Name : "+K.toString());
    System.out.println("Designation : "+Employee2.get(K));
}
System.out.println();
}

```

Output :

```

Choose A Question :
3
Your Employee List :
Name : TiaPanday
Designation : CEO
Name : YasserDesai
Designation : ProjectManager
Name : RaveenaRaut
Designation : WebDesigner
Name : ChetanMukharjee
Designation : Manager
Name : FenilShah
Designation : Developer

Coppied Employee List :
Name : ChetanMukharjee
Designation : Manager
Name : TiaPanday
Designation : CEO
Name : YasserDesai
Designation : ProjectManager
Name : FenilShah
Designation : Developer
Name : RaveenaRaut
Designation : WebDesigner

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Count Key-Value Mappings In Map.
3.Copy One Hash Maps To Other Map.

```

4. Write a Java program to remove all mappings from a map.

Source Code :

```

protected void RemoveAllMappings(HashMap<String, String> Employee){
    if(Employee.size() <= 0)
    {
        System.out.println("No Employees Available!!!");
    }
    else
    {
        Employee.clear();
        System.out.println("All Employees Removed!!!");
    }
}

```


```

    }

    System.out.println("Your Employee List : ");
    for(String K:Employee.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Designation : "+Employee.get(K));
    }
    System.out.println();
}

```

Output :



```

Choose A Question :
4
All Employees Removed!!
Your Employee List :

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Count Key-Value Mappings In Map.
3.Copy One Hash Maps To Other Map.
4.Remove All Mappings From Map.

```

5. Write a Java program to check whether a map contains key-value mappings (empty) or not.

Source Code :

```

protected void CheckKeyValueMappings(HashMap<String, String> Employee){
    String Key, Value;

    System.out.print("\nEnter Employee Name : ");
    Key = in.next();
    System.out.print("\nEnter Employee Designation : ");
    Value = in.next();

    if(Employee.containsKey(Key) && Employee.containsValue(Value))
    {
        System.out.println("Hash Map Contains Employee "+Key+" With Designation "+Value+"...");
    }
    else
    {
        System.out.println("No Employee "+Key+" With Designation "+Value+" Found in Hash Map...");
    }
    System.out.println();
}

```

Output :

```
Choose A Question :
5

Enter Employee Name : HarshModi

Enter Employee Designation : Student
No Employee HarshModi With Designation Student Found in Hash Map...

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Count Key-Value Mappings In Map.
3.Copy One Hash Maps To Other Map.
4.Remove All Mappings From Map.
5.Check Map Contains Key-Value Mapping.
```

6. Write a Java program to get a shallow copy of a HashMap instance.

Source Code :

```
protected void GetShallowCopy(HashMap<String, String> Employee){
    System.out.println("Your Employee List : ");
    for(String K:Employee.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Designation : "+Employee.get(K));
    }

    HashMap<String, String> Employee2 = (HashMap<String, String>)Employee.clone();

    System.out.println("\nCoppied Employee List : ");
    for(String K:Employee2.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Designation : "+Employee2.get(K));
    }
    System.out.println();
}
```

Output :

```
Choose A Question :
6

Your Employee List :
Name : TiaPanday
Designation : CEO
Name : YasserDesai
Designation : ProjectManager
Name : RaveenaRaut
Designation : WebDesigner
Name : ChetanMukharjee
Designation : Manager
Name : FenilShah
Designation : Developer

Coppied Employee List :
Name : ChetanMukharjee
Designation : Manager
Name : TiaPanday
Designation : CEO
Name : YasserDesai
Designation : ProjectManager
Name : FenilShah
Designation : Developer
Name : RaveenaRaut
Designation : WebDesigner

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Count Key-Value Mappings In Map.
3.Copy One Hash Maps To Other Map.
4.Remove All Mappings From Map.
5.Check Map Contains Key-Value Mapping.
6.Get Shallow Copy of Hash Map.
```

7. Write a Java program to test if a map contains a mapping for the specified key.

Source Code :

```
protected void CheckKeyMappings(HashMap<String, String> Employee){
    String Key, Value;

    System.out.print("\nEnter Employee Name : ");
    Key = in.next();

    if(Employee.containsKey(Key))
    {
        System.out.println("Hash Map Contains Employee "+Key+" With Designation "+Employee.get(Key)+"...");
    }
    else
    {
        System.out.println("No Employee "+Key+" Found in Hash Map...");
    }
    System.out.println();
}
```

Output :

A screenshot of a Java IDE with a dark theme. On the left, a menu lists options: 7. Check Map Contains Key Mapping, 8. Check Map Contains Value Mapping, 9. Create Set View Of Mappings, 10. Get Value of Specified Key, 11. Get Key Set, 12. Get Value Collection, 0. Exit. The user has selected option 7. The main window shows the prompt "Enter Employee Name : TiaPanday" and the output "Hash Map Contains Employee TiaPanday With Designation CEO...".

```
7. Check Map Contains Key Mapping.
8. Check Map Contains Value Mapping.
9. Create Set View Of Mappings.
10. Get Value of Specified Key.
11. Get Key Set.
12. Get Value Collection.
0. Exit.
Choose A Question :
7

Enter Employee Name : TiaPanday
Hash Map Contains Employee TiaPanday With Designation CEO...
```

8. Write a Java program to test if a map contains a mapping for the specified value.

Source Code :

```
protected void CheckValueMappings(HashMap<String, String> Employee){
    String Value;

    System.out.print("\nEnter Employee Designation : ");
    Value = in.next();

    if(Employee.containsValue(Value))
    {
        System.out.println("Hash Map Contains Designation "+Value+"...");
    }
    else
```

```

    {
        System.out.println("No Designation "+Value+" Found in Hash Map...");
    }
    System.out.println();
}

```

Output :



```

8.Check Map Contains Value Mapping.
9.Create Set View Of Mappings.
10.Get Value of Specified Key.
11.Get Key Set.
12.Get Value Collection.
0.Exit.
Choose A Question :
8

Enter Employee Designation : Developer
Hash Map Contains Designation Developer...

```

9. Write a Java program to create a set view of the mappings contained in a map.

Source Code :

```

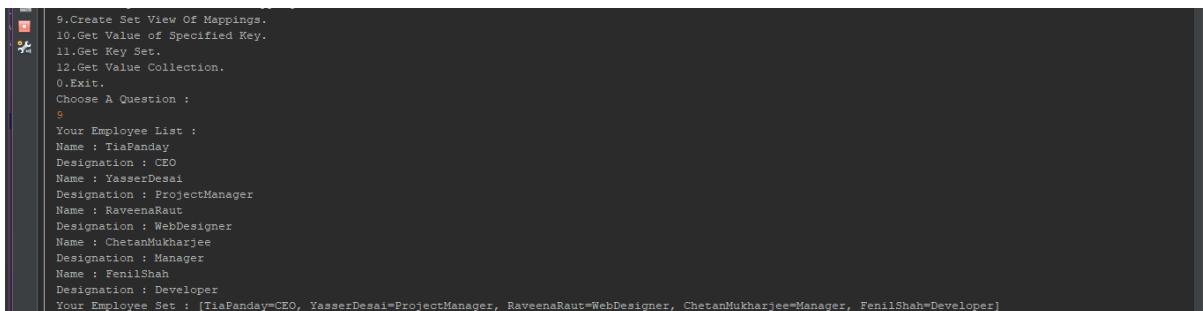
protected void CreateSetView(HashMap<String, String> Employee){
    System.out.println("Your Employee List : ");
    for(String K:Employee.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Designation : "+Employee.get(K));
    }

    Set Employee2 = Employee.entrySet();

    System.out.println("Your Employee Set : "+Employee2);
    System.out.println();
}

```

Output :



```

9.Create Set View Of Mappings.
10.Get Value of Specified Key.
11.Get Key Set.
12.Get Value Collection.
0.Exit.
Choose A Question :
9

Your Employee List :
Name : TiaPanday
Designation : CEO
Name : YasserDesai
Designation : ProjectManager
Name : RaveenaRaut
Designation : WebDesigner
Name : ChetanMukharjee
Designation : Manager
Name : FenilShah
Designation : Developer
Your Employee Set : [TiaPanday=CEO, YasserDesai=ProjectManager, RaveenaRaut=WebDesigner, ChetanMukharjee=Manager, FenilShah=Developer]

```

10. Write a Java program to get the value of a specified key in a map.

Source Code :

```
protected void GetValueOfKey(HashMap<String, String> Employee){
    String Key;
    System.out.print("\nEnter Employee Name : ");
    Key = in.next();

    if(Employee.containsKey(Key))
    {
        System.out.print("\nEmployee "+Key+" has "+Employee.get(Key)+" Designation ");
    }
    else
    {
        System.out.println("No Employee Found With Name "+Key);
    }
    System.out.println();
}
```

Output :

A screenshot of a Java program's output in a terminal window. The output shows a menu with options: 10. Get Value of Specified Key, 11. Get Key Set, 12. Get Value Collection, 0. Exit. The user has chosen option 10. The program then prompts for an employee name, and the user has entered 'YasserDesai'. The final output is 'Employee YasserDesai has ProjectManager Designation'.

11. Write a Java program to get a set view of the keys contained in this map.

Source Code :

```
protected void GetKeySet(HashMap<String, String> Employee){
    System.out.println("Your Employee List : ");
    for(String K:Employee.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Designation : "+Employee.get(K));
    }

    Set Employee2 = Employee.keySet();

    System.out.println("Your Employee Key Set : "+Employee2);
    System.out.println();
}
```


Output :

```
11.Get Key Set.
12.Get Value Collection.
0.Exit.
Choose A Question :
11
Your Employee List :
Name : TiaPanday
Designation : CEO
Name : YasserDesai
Designation : ProjectManager
Name : RaveenaRaut
Designation : WebDesigner
Name : ChetanMukharjee
Designation : Manager
Name : FenilShah
Designation : Developer
Your Employee Key Set : [TiaPanday, YasserDesai, RaveenaRaut, ChetanMukharjee, FenilShah]
```

12. Write a Java program to get a collection view of the values contained in this map.

Source Code :

```
protected void GetValueCollection(HashMap<String, String> Employee){
    System.out.println("Your Employee List : ");
    for(String K:Employee.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Designation : "+Employee.get(K));
    }

    Collection<String> Employee2 = Employee.values();

    System.out.println("Your Designation Collection : "+Employee2);
    System.out.println();
}
```

Output :

```
12.Get Value Collection.
0.Exit.
Choose A Question :
12
Your Employee List :
Name : TiaPanday
Designation : CEO
Name : YasserDesai
Designation : ProjectManager
Name : RaveenaRaut
Designation : WebDesigner
Name : ChetanMukharjee
Designation : Manager
Name : FenilShah
Designation : Developer
Your Designation Collection : [CEO, ProjectManager, WebDesigner, Manager, Developer]
```

G) Tree Map :

```
TreeMap<String, Integer> Series = new TreeMap<String, Integer>();
```

1. Write a Java program to associate the specified value with the specified key in a Tree Map.

Source Code :

```
protected void AssociateKeyValues(TreeMap<String, Integer> Series){
    String Key;
    Integer Value;

    System.out.println("\nHow Many Series You Want to Enter ?");
    System.out.print(">");
    int n = in.nextInt();

    for(int i=0;i<n;i++)
    {
        System.out.print("\nEnter Series["+i+"] Name : ");
        Key = in.next();
        System.out.print("\nEnter Series["+i+"] Episodes : ");
        Value = in.nextInt();

        Series.put(Key, Value);
        System.out.println();
    }

    System.out.println("Series List : ");
    for(String K:Series.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Episodes : "+Series.get(K));
    }
    System.out.println();
}
```

Output :

```
Choose A Question :
1
How Many Series You Want to Enter ?
>5
Enter Series[0] Name : CampusBeats
Enter Series[0] Episodes : 30

Enter Series[1] Name : Wednesday
Enter Series[1] Episodes : 8

Enter Series[2] Name : 2Gether
Enter Series[2] Episodes : 15

Enter Series[3] Name : Fittrat
Enter Series[3] Episodes : 22

Enter Series[4] Name : SweetHome
Enter Series[4] Episodes : 26

Series List :
Name : 2Gether
Episodes : 15
Name : CampusBeats
Episodes : 30
Name : Fittrat
Episodes : 22
Name : SweetHome
Episodes : 26
Name : Wednesday
Episodes : 8

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
```

2. Write a Java program to copy Tree Maps content to another Tree Map. Source Code :

```
protected void CopyOneTreeMapToOtherMap(TreeMap<String, Integer> Series){
    System.out.println("Your Series List : ");
    for(String K:Series.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Episodes : "+Series.get(K));
    }

    TreeMap<String, Integer> Series2 = new TreeMap<String, Integer>(Series);

    System.out.println("\nCoppied Employee List : ");
    for(String K:Series2.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Episodes : "+Series2.get(K));
    }
    System.out.println();
}
```

Output :

```
Choose A Question :
2
Your Series List :
Name : 2Gether
Episodes : 15
Name : CampusBeats
Episodes : 30
Name : Fittrat
Episodes : 22
Name : SweetHome
Episodes : 26
Name : Wednesday
Episodes : 8

Copied Employee List :
Name : 2Gether
Episodes : 15
Name : CampusBeats
Episodes : 30
Name : Fittrat
Episodes : 22
Name : SweetHome
Episodes : 26
Name : Wednesday
Episodes : 8

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
```

3. Write a Java program to search for a key in a Tree Map.

Source Code :

```
protected void SearchKey(TreeMap<String, Integer> Series){
    String Key;

    System.out.print("\nEnter Series Name : ");
    Key = in.next();

    if(Series.containsKey(Key))
    {
        System.out.println("Tree Map Contains Series "+Key+" With No. of Episodes
"+Series.get(Key)+"...");
    }
    else
    {
        System.out.println("No Series "+Key+" Found in Tree Map...");
    }
    System.out.println();
}
```

Output :

```
Choose A Question :
3

Enter Series Name : Fittrat
Tree Map Contains Series Fittrat With No. of Episodes 22...

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
3.Search For a Key in Map.
```

4. Write a Java program to search for a value in a Tree Map.

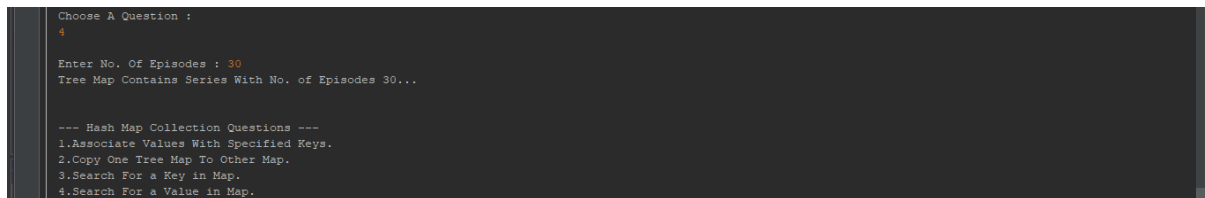
Source Code :

```
protected void SearchValue(TreeMap<String, Integer> Series){
    Integer Value;

    System.out.print("\nEnter No. Of Episodes : ");
    Value = in.nextInt();

    if(Series.containsValue(Value))
    {
        System.out.println("Tree Map Contains Series With No. of Episodes "+Value+"...");
    }
    else
    {
        System.out.println("No Series Found in Tree Map with No. of Episodes "+Value+"...");
    }
    System.out.println();
}
```

Output :



The screenshot shows the execution of the Java program. It starts with a prompt "Choose A Question :" followed by the user input "4". Then, it prompts "Enter No. Of Episodes : 30" and displays the output "Tree Map Contains Series With No. of Episodes 30...". Below this, a menu titled "--- Hash Map Collection Questions ---" is shown with five options: 1. Associate Values With Specified Keys, 2. Copy One Tree Map To Other Map, 3. Search For a Key in Map, 4. Search For a Value in Map, and 5. Get All Keys in Map.

5. Write a Java program to get all keys from a Tree Map.

Source Code :

```
protected void GetAllKeys(TreeMap<String, Integer> Series){
    System.out.print("\nKey List : "+Series.keySet());
    System.out.println();
}
```

Output :



The screenshot shows the execution of the Java program. It starts with a prompt "Choose A Question :" followed by the user input "5". Then, it displays the output "Key List : [2Gether, CampusBeats, Fittrat, SweetHome, Wednesday]". Below this, the same menu titled "--- Hash Map Collection Questions ---" is shown with five options: 1. Associate Values With Specified Keys, 2. Copy One Tree Map To Other Map, 3. Search For a Key in Map, 4. Search For a Value in Map, and 5. Get All Keys in Map.

6. Write a Java program to delete all elements from a Tree Map.

Source Code :

```
protected void DeleteAllElements(TreeMap<String, Integer> Series){
    Series.clear();
    System.out.println("All Elements Deleted...");
    System.out.println("Your Series List : ");
    for(String K:Series.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Episodes : "+Series.get(K));
    }
    System.out.println();
}
```

Output :



```
Choose A Question :
6
All Elements Deleted...
Your Series List :

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
3.Search For a Key in Map.
4.Search For a Value in Map.
5.Get All Keys in Map.
6.Delete All Elements.
```

7. Write a Java program to sort keys in a Tree Map by using a comparator.

Source Code :

```
class sort_Keys implements Comparator<String>{

    @Override
    public int compare(String o1, String o2) {
        return o1.compareTo(o2);
    }

}

protected void SortKeysUsingComparator(TreeMap<String, Integer> Series){
    TreeMap<String, Integer> Series2 = new TreeMap<String, Integer>(new sort_Keys());
    Series2.putAll(Series);

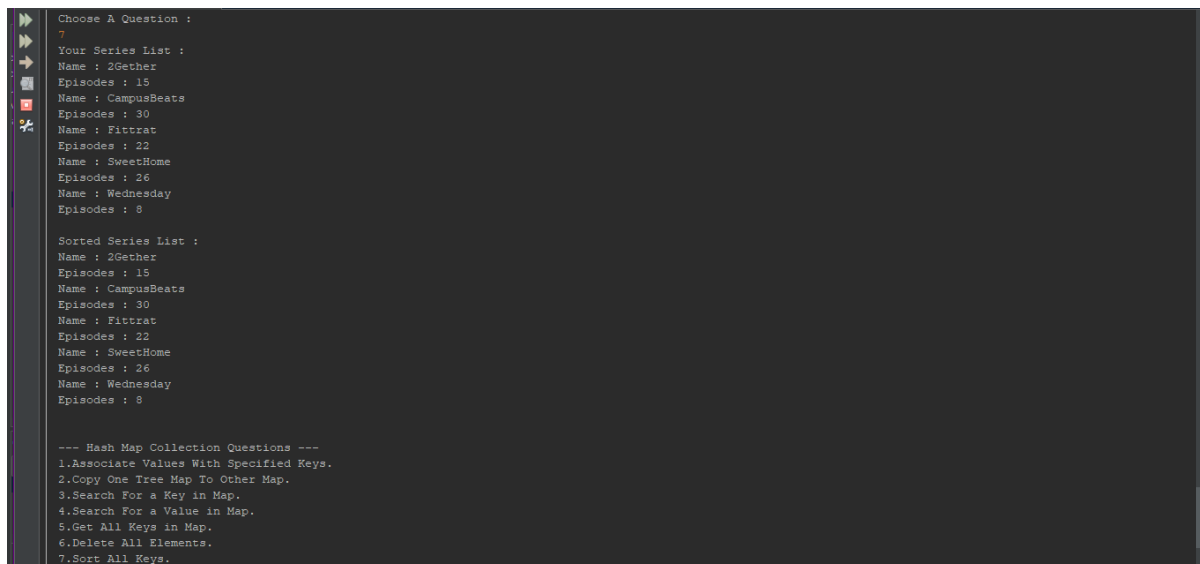
    System.out.println("Your Series List : ");
    for(String K:Series.keySet())
    {
```

```

        System.out.println("Name : "+K.toString());
        System.out.println("Episodes : "+Series.get(K));
    }
    System.out.println();
    System.out.println("Sorted Series List : ");
    for(String K:Series2.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Episodes : "+Series2.get(K));
    }
    System.out.println();
}

```

Output :



```

Choose A Question :
7
Your Series List :
Name : 2Gether
Episodes : 15
Name : CampusBeats
Episodes : 30
Name : Fitrrat
Episodes : 22
Name : SweetHome
Episodes : 26
Name : Wednesday
Episodes : 8

Sorted Series List :
Name : 2Gether
Episodes : 15
Name : CampusBeats
Episodes : 30
Name : Fitrrat
Episodes : 22
Name : SweetHome
Episodes : 26
Name : Wednesday
Episodes : 8

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
3.Search For a Key in Map.
4.Search For a Value in Map.
5.Get All Keys in Map.
6.Delete All Elements.
7.Sort All Keys.

```

8. Write a Java program to get a key-value mapping associated with the greatest key and the least key in a map.

Source Code :

```

protected void GetGreaterLeastKeyValue(TreeMap<String, Integer> Series){
    System.out.println("Your Series List : ");
    for(String K:Series.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Episodes : "+Series.get(K));
    }

    System.out.println("Greatest Key-Value in List : "+Series.firstEntry());
}

```

```

        System.out.println("Least Key-Value in List : "+Series.lastEntry());
        System.out.println();
    }
}

```

Output :



```

Choose A Question :
8
Your Series List :
Name : 2Gether
Episodes : 15
Name : CampusBeats
Episodes : 30
Name : Fittrat
Episodes : 22
Name : SweetHome
Episodes : 26
Name : Wednesday
Episodes : 8
Greatest Key-Value in List : 2Gether=15
Least Key-Value in List : Wednesday=8

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
3.Search For a Key in Map.
4.Search For a Value in Map.
5.Get All Keys in Map.
6.Delete All Elements.
7.Sort All Keys.
8.Get Greatest & Least Key-Value.

```

9. Write a Java program to get the first (lowest) key and the last (highest) key currently in a map.

Source Code :

```

protected void GetFirstLastKey(TreeMap<String, Integer> Series){
    System.out.println("Your Series List : ");
    for(String K:Series.keySet())
    {
        System.out.println("Name : "+K.toString());
        System.out.println("Episodes : "+Series.get(K));
    }

    System.out.println("First Key in List : "+Series.firstKey());
    System.out.println("Last Key in List : "+Series.lastKey());
    System.out.println();
}

```

Output :



```

Choose A Question :
9
Your Series List :
Name : 2Gether
Episodes : 15
Name : CampusBeats
Episodes : 30
Name : Fittrat
Episodes : 22
Name : SweetHome
Episodes : 26
Name : Wednesday
Episodes : 8
First Key in List : 2Gether
Last Key in List : Wednesday

```


10. Write a Java program to get a reverse order view of the keys contained in a given map.

Source Code :

```
protected void GetReversedKeys(TreeMap<String, Integer> Series){
    System.out.print("\nYour Key List : "+Series.keySet());
    System.out.print("\nReversed Key List : "+Series.descendingMap());
    System.out.println();
}
```

Output :

```
Choose A Question :
10

Your Key List : [2Gether, CampusBeats, Fittrat, SweetHome, Wednesday]
Reversed Key List : (Wednesday=8, SweetHome=26, Fittrat=22, CampusBeats=30, 2Gether=15)

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
3.Search For a Key in Map.
4.Search For a Value in Map.
5.Get All Keys in Map.
6.Delete All Elements.
7.Sort All Keys.
8.Get Greatest & Least Key-Value.
9.Get First & Last Key.
10.Get Reversed Key Mapping.
```

11. Write a Java program to get a key-value mapping associated with the greatest key less than or equal to the given key.

Source Code :

```
protected void GetKeyValueGreatestLessEqual(TreeMap<String, Integer> Series){
    System.out.print("\nYour Key List : "+Series);
    System.out.print("\nEnter Key : ");
    String Key = in.next();

    System.out.print("\nKey-Value Associated To Greatest key Less Equal to "+Key+" : "+Series.floorEntry(Key));
    System.out.println();
}
```

Output :

```
Choose A Question :
11

Your Key List : (2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8)
Enter Key : Fittrat

Key-Value Associated To Greatest key Less Equal to Fittrat : Fittrat=22

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
3.Search For a Key in Map.
4.Search For a Value in Map.
5.Get All Keys in Map.
6.Delete All Elements.
7.Sort All Keys.
8.Get Greatest & Least Key-Value.
9.Get First & Last Key.
10.Get Reversed Key Mapping.
11.Get Key Value Associated to Greatest Key Less Than Equal to Key.
```

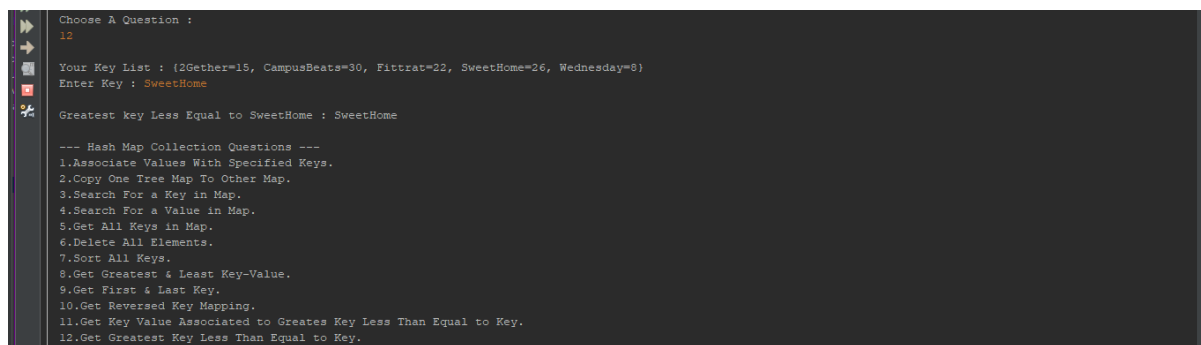
12. Write a Java program to get the greatest key less than or equal to the given key.

Source Code :

```
protected void GetGreatestKeyLessEqual(TreeMap<String, Integer> Series){
    System.out.print("\nYour Key List : "+Series);
    System.out.print("\nEnter Key : ");
    String Key = in.next();

    System.out.print("\nGreatest key Less Equal to "+Key+" : "+Series.floorKey(Key));
    System.out.println();
}
```

Output :

A screenshot of a Java IDE showing the output of a program. The output text is: "Choose A Question : 12", "Your Key List : {2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8}", "Enter Key : SweetHome", "Greatest key Less Equal to SweetHome : SweetHome". Below this, there is a list of 12 questions related to Hash Map collection operations.

```
Choose A Question :
12
Your Key List : {2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8}
Enter Key : SweetHome
Greatest key Less Equal to SweetHome : SweetHome

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
3.Search For a Key in Map.
4.Search For a Value in Map.
5.Get All Keys in Map.
6.Delete All Elements.
7.Sort All Keys.
8.Get Greatest & Least Key-Value.
9.Get First & Last Key.
10.Get Reversed Key Mapping.
11.Get Key Value Associated to Greatest Key Less Than Equal to Key.
12.Get Greatest Key Less Than Equal to Key.
```

13. Write a Java program to get the portion of a map whose keys are strictly less than a given key.

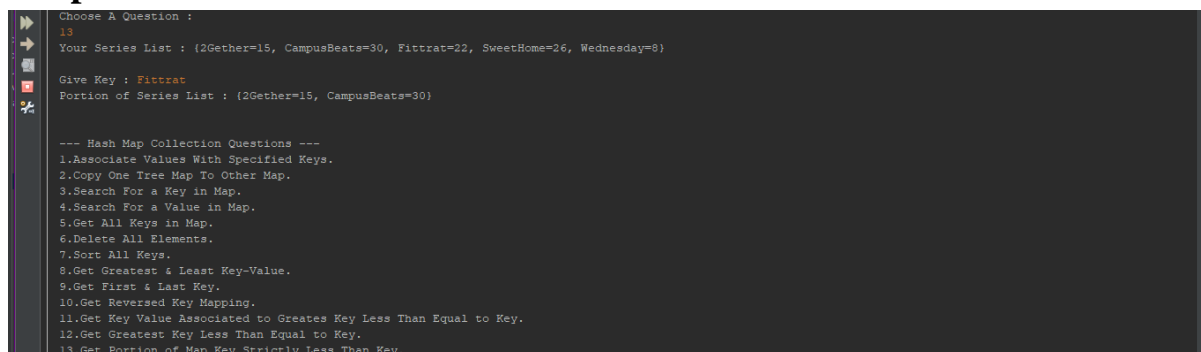
Source Code :

```
protected void GetPortionStriuctlyLessThanKey(TreeMap<String, Integer> Series){
    System.out.println("Your Series List : "+Series);

    System.out.print("\nGive Key : ");
    String Key = in.next();

    System.out.println("Portion of Series List : "+Series.headMap(Key));
    System.out.println();
}
```

Output :

A screenshot of a Java IDE showing the output of a program. The output text is: "Choose A Question : 13", "Your Series List : {2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8}", "Give Key : Fittrat", "Portion of Series List : {2Gether=15, CampusBeats=30}". Below this, there is a list of 13 questions related to Hash Map collection operations.

```
Choose A Question :
13
Your Series List : {2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8}
Give Key : Fittrat
Portion of Series List : {2Gether=15, CampusBeats=30}

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
3.Search For a Key in Map.
4.Search For a Value in Map.
5.Get All Keys in Map.
6.Delete All Elements.
7.Sort All Keys.
8.Get Greatest & Least Key-Value.
9.Get First & Last Key.
10.Get Reversed Key Mapping.
11.Get Key Value Associated to Greatest Key Less Than Equal to Key.
12.Get Greatest Key Less Than Equal to Key.
13.Get Portion of Map Key Strictly Less Than Key.
```

14. Write a Java program to get the portion of this map whose keys are less than (or equal to, if inclusive is true) a given key.

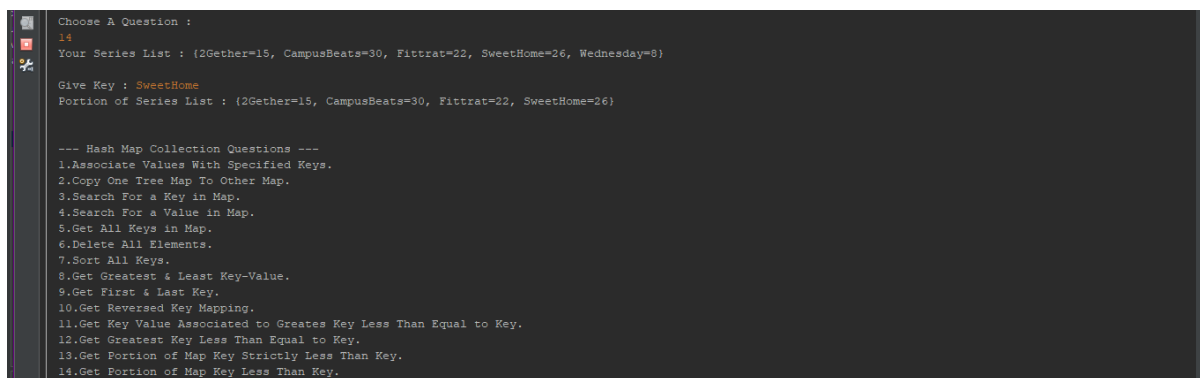
Source Code :

```
protected void GetPortionLessThanKey(TreeMap<String, Integer> Series){
    System.out.println("Your Series List : "+Series);

    System.out.print("\nGive Key : ");
    String Key = in.next();

    System.out.println("Portion of Series List : "+Series.headMap(Key, true));
    System.out.println();
}
```

Output :



```
Choose A Question :
14
Your Series List : {2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8}

Give Key : SweetHome
Portion of Series List : {2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26}

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
3.Search For a Key in Map.
4.Search For a Value in Map.
5.Get All Keys in Map.
6.Delete All Elements.
7.Sort All Keys.
8.Get Greatest & Least Key-Value.
9.Get First & Last Key.
10.Get Reversed Key Mapping.
11.Get Key Value Associated to Greatest Key Less Than Equal to Key.
12.Get Greatest Key Less Than Equal to Key.
13.Get Portion of Map Key Strictly Less Than Key.
14.Get Portion of Map Key Less Than Key.
```

15. Write a Java program to get the least key strictly greater than the given key. Return null if there is no such key.

Source Code :

```
protected void GetLeastKeyStrictlyGreaterThanKey(TreeMap<String, Integer> Series){
    System.out.print("\nYour Key List : "+Series);
    System.out.print("\nEnter Key : ");
    String Key = in.next();

    System.out.print("\nLeast key Strictly Greater than "+Key+" : "+Series.higherKey(Key));
    System.out.println();
}
```

Output :

```
Choose A Question :
15

Your Key List : {2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8}
Enter Key : Fittrat

Least Key Strictly Greater than Fittrat : SweetHome

--- Hash Map Collection Questions ---
1.Associate Values With Specified Keys.
2.Copy One Tree Map To Other Map.
3.Search For a Key in Map.
4.Search For a Value in Map.
5.Get All Keys in Map.
6.Delete All Elements.
7.Sort All Keys.
8.Get Greatest & Least Key-Value.
9.Get First & Last Key.
10.Get Reversed Key Mapping.
11.Get Key Value Associated to Greatest Key Less Than Equal to Key.
12.Get Greatest Key Less Than Equal to Key.
13.Get Portion of Map Key Strictly Less Than Key.
14.Get Portion of Map Key Less Than Key.
15.Get Least Key Strictly Greater Than Key.
```

16. Write a Java program to get a key-value mapping associated with the greatest key strictly less than the given key. Return null if there is no such key.

Source Code :

```
protected void GetKeyValueGreatestKeyLessThan(TreeMap<String, Integer> Series){
    System.out.print("\nYour Key List : "+Series);
    System.out.print("\nEnter Key : ");
    String Key = in.next();

    System.out.print("\nKey-Value Associated To Greatest Key Strictly Less than "+Key+" : 
"+Series.lowerEntry(Key));
    System.out.println();
}
```

Output :

```
16.Get Key Value Associated to Greatest Key Strictly Less Than Key.
17.Get Greatest Key Strictly Less Than Key.
18.Get Navigable Set View.
19.Remove And Get Key-Value Associated With Least Key.
20.Remove And Get Key-Value Associated With Greatest Key.
21.Get Portion Of Map From Start(Inclusive) And End(Exclusive) Key.
22.Get Portion Of Map From Start And End Key.
23.Get Portion Of Map Start From Greater Than Equal to Key.
24.Get Portion Of Map Start From Greater Than Key.
25.Get Key Value Associated to Least Key Less Than Equal to Key.
26.Get Least Key Greater Than Key.
0.Exit.
Choose A Question :
16

Your Key List : {2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8}
Enter Key : CampusBeats

Key-Value Associated To Greatest Key Strictly Less than CampusBeats : 2Gether=15
```

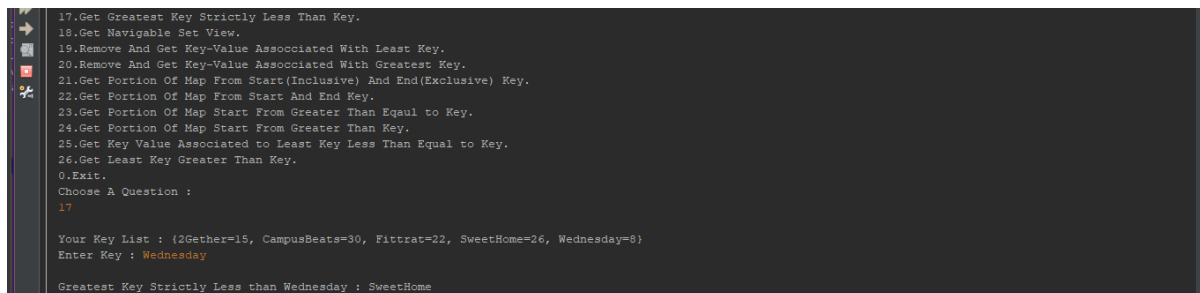
**17. Write a Java program to get the greatest key strictly less than the given key.
Return null if there is no such key.**

Source Code :

```
protected void GetGreatestKeyStrictlyLessThan(TreeMap<String, Integer> Series){
    System.out.print("\nYour Key List : "+Series);
    System.out.print("\nEnter Key : ");
    String Key = in.next();

    System.out.print("\nGreatest Key Strictly Less than "+Key+" : "+Series.lowerKey(Key));
    System.out.println();
}
```

Output :



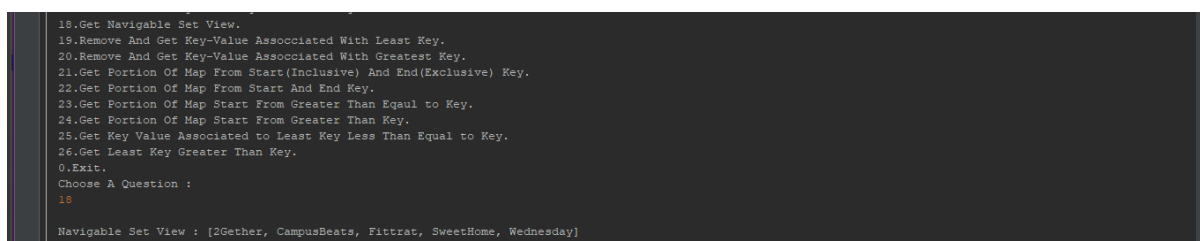
```
17.Get Greatest Key Strictly Less Than Key.
18.Get Navigable Set View.
19.Remove And Get Key-Value Associated With Least Key.
20.Remove And Get Key-Value Associated With Greatest Key.
21.Get Portion Of Map From Start(Inclusive) And End(Exclusive) Key.
22.Get Portion Of Map From Start And End Key.
23.Get Portion Of Map Start From Greater Than Equal to Key.
24.Get Portion Of Map Start From Greater Than Key.
25.Get Key Value Associated to Least Key Less Than Equal to Key.
26.Get Least Key Greater Than Key.
0.Exit.
Choose A Question :
17
Your Key List : {2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8}
Enter Key : Wednesday
Greatest Key Strictly Less than Wednesday : SweetHome
```

18. Write a Java program to get a NavigableSet view of keys in a map.

Source Code :

```
protected void GetNavigableSetView(TreeMap<String, Integer> Series){
    System.out.print("\nNavigable Set View : "+Series.navigableKeySet());
    System.out.println();
}
```

Output :



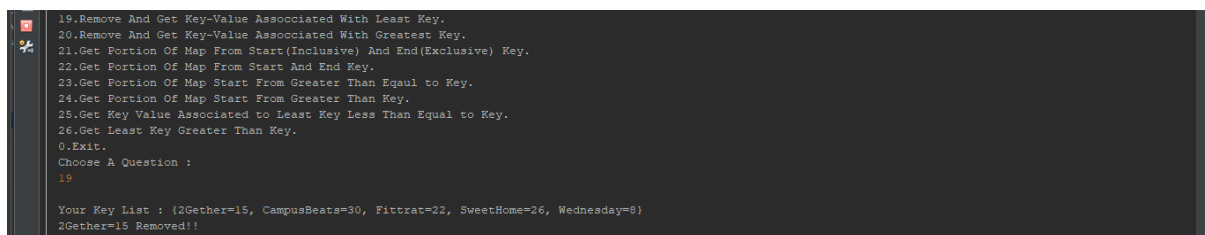
```
18.Get Navigable Set View.
19.Remove And Get Key-Value Associated With Least Key.
20.Remove And Get Key-Value Associated With Greatest Key.
21.Get Portion Of Map From Start(Inclusive) And End(Exclusive) Key.
22.Get Portion Of Map From Start And End Key.
23.Get Portion Of Map Start From Greater Than Equal to Key.
24.Get Portion Of Map Start From Greater Than Key.
25.Get Key Value Associated to Least Key Less Than Equal to Key.
26.Get Least Key Greater Than Key.
0.Exit.
Choose A Question :
18
Navigable Set View : [2Gether, CampusBeats, Fittrat, SweetHome, Wednesday]
```

19. Write a Java program to remove and get a key-value mapping associated with the least key in a map.

Source Code :

```
protected void RemoveReturnLeastKeyValue(TreeMap<String, Integer> Series){  
    System.out.print("\nYour Key List : "+Series);  
    System.out.println("\n"+Series.pollFirstEntry()+" Removed!!");  
    System.out.println();  
}
```

Output :



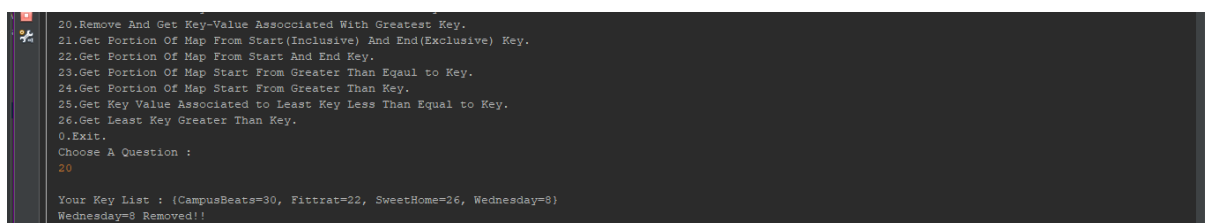
```
19.Remove And Get Key-Value Associated With Least Key.  
20.Remove And Get Key-Value Associated With Greatest Key.  
21.Get Portion Of Map From Start(Inclusive) And End(Exclusive) Key.  
22.Get Portion Of Map From Start And End Key.  
23.Get Portion Of Map Start From Greater Than Equal to Key.  
24.Get Portion Of Map Start From Greater Than Key.  
25.Get Key Value Associated to Least Key Less Than Equal to Key.  
26.Get Least Key Greater Than Key.  
0.Exit.  
Choose A Question :  
19  
  
Your Key List : {2Gether=15, CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8}  
2Gether=15 Removed!!
```

20. Write a Java program to remove and get a key-value mapping associated with the greatest key in this map.

Source Code :

```
protected void RemoveReturnGreatestKeyValue(TreeMap<String, Integer> Series){  
    System.out.print("\nYour Key List : "+Series);  
    System.out.println("\n"+Series.pollLastEntry()+" Removed!!");  
    System.out.println();  
}
```

Output :



```
20.Remove And Get Key-Value Associated With Greatest Key.  
21.Get Portion Of Map From Start(Inclusive) And End(Exclusive) Key.  
22.Get Portion Of Map From Start And End Key.  
23.Get Portion Of Map Start From Greater Than Equal to Key.  
24.Get Portion Of Map Start From Greater Than Key.  
25.Get Key Value Associated to Least Key Less Than Equal to Key.  
26.Get Least Key Greater Than Key.  
0.Exit.  
Choose A Question :  
20  
  
Your Key List : {CampusBeats=30, Fittrat=22, SweetHome=26, Wednesday=8}  
Wednesday=8 Removed!!
```

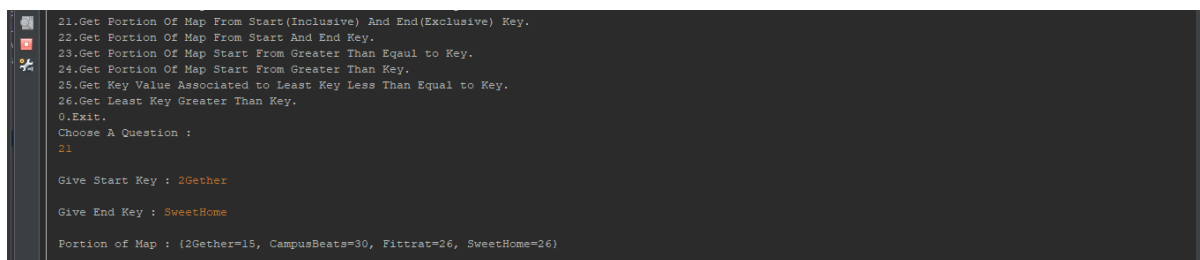
21. Write a Java program to get the portion of a map whose keys range from a given key (inclusive) to another key (exclusive).

Source Code :

```
protected void GetPortionInclusiveExclusive(TreeMap<String, Integer> Series){
    System.out.print("\nGive Start Key : ");
    String Start = in.next();
    System.out.print("\nGive End Key : ");
    String End = in.next();

    System.out.println("\nPortion of Map : "+Series.subMap(Start, true, End, true));
    System.out.println();
}
```

Output :



```
21.Get Portion Of Map From Start(Inclusive) And End(Exclusive) Key.
22.Get Portion Of Map From Start And End Key.
23.Get Portion Of Map Start From Greater Than Equal to Key.
24.Get Portion Of Map Start From Greater Than Key.
25.Get Key Value Associated to Least Key Less Than Equal to Key.
26.Get Least Key Greater Than Key.
0.Exit.
Choose A Question :
21

Give Start Key : 2Gether
Give End Key : SweetHome

Portion of Map : {2Gether=15, CampusBeats=30, Fitrrat=26, SweetHome=26}
```

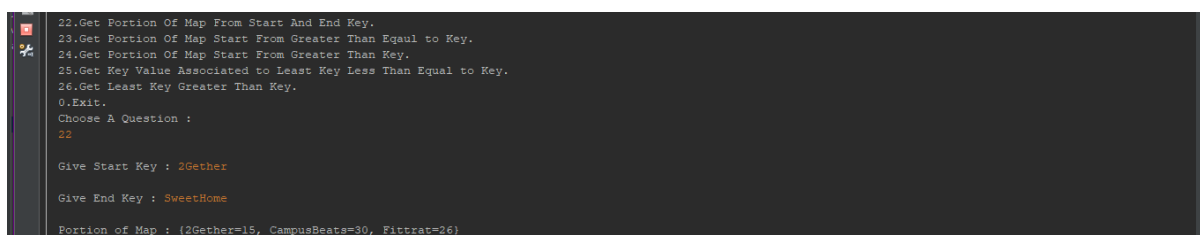
22. Write a Java program to get the portion of a map whose keys range from a given key to another key.

Source Code :

```
protected void GetPortion(TreeMap<String, Integer> Series){
    System.out.print("\nGive Start Key : ");
    String Start = in.next();
    System.out.print("\nGive End Key : ");
    String End = in.next();

    System.out.println("\nPortion of Map : "+Series.subMap(Start, End));
    System.out.println();
}
```

Output :



```
22.Get Portion Of Map From Start And End Key.
23.Get Portion Of Map Start From Greater Than Equal to Key.
24.Get Portion Of Map Start From Greater Than Key.
25.Get Key Value Associated to Least Key Less Than Equal to Key.
26.Get Least Key Greater Than Key.
0.Exit.
Choose A Question :
22

Give Start Key : 2Gether
Give End Key : SweetHome

Portion of Map : {2Gether=15, CampusBeats=30, Fitrrat=26}
```

23. Write a Java program to get a portion of a map whose keys are greater than or equal to a given key.

Source Code :

```
protected void GetPortionKeyGreaterEqualKey(TreeMap<String, Integer> Series){
    System.out.print("\nGive Key : ");
    String Start = in.next();

    System.out.println("\nPortion of Map : "+Series.tailMap(Start));
    System.out.println();
}
```

Output :

```
23.Get Portion Of Map Start From Greater Than Equal to Key.
24.Get Portion Of Map Start From Greater Than Key.
25.Get Key Value Associated to Least Key Less Than Equal to Key.
26.Get Least Key Greater Than Key.
0.Exit.
Choose A Question :
23

Give Key : Wednesday

Portion of Map : (Wednesday=8)
```

24. Write a Java program to get a portion of a map whose keys are greater than a given key.

Source Code :

```
protected void GetPortionKeyGreaterThanKey(TreeMap<String, Integer> Series){
    System.out.print("\nGive Key : ");
    String Start = in.next();

    System.out.println("\nPortion of Map : "+Series.tailMap(Start, true));
    System.out.println();
}
```

Output :

```
24.Get Portion Of Map Start From Greater Than Key.
25.Get Key Value Associated to Least Key Less Than Equal to Key.
26.Get Least Key Greater Than Key.
0.Exit.
Choose A Question :
24

Give Key : CampusBeats

Portion of Map : (CampusBeats=30, Fittrat=26, SweetHome=26, Wednesday=8)
```

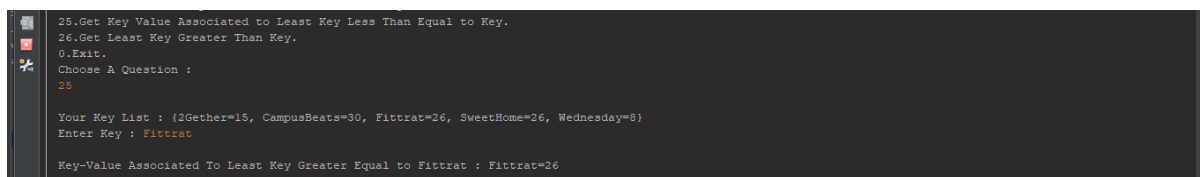

25. Write a Java program to get a key-value mapping associated with the least key greater than or equal to the given key. Return null if there is no such key.

Source Code :

```
protected void GetKeyValueLeastGreaterEqual(TreeMap<String, Integer> Series){
    System.out.print("\nYour Key List : "+Series);
    System.out.print("\nEnter Key : ");
    String Key = in.next();

    System.out.print("\nKey-Value Associated To Least Key Greater Equal to "+Key+" : "+Series.ceilingEntry(Key));
    System.out.println();
}
```

Output :

A screenshot of a Java IDE showing the execution of a program. The output text is: 25.Get Key Value Associated to Least Key Less Than Equal to Key. 26.Get Least Key Greater Than Key. 0.Exit. Choose A Question : 25 Your Key List : {2Gether=15, CampusBeats=30, Fittrat=26, SweetHome=26, Wednesday=8} Enter Key : Fittrat Key-Value Associated To Least Key Greater Equal to Fittrat : Fittrat=26

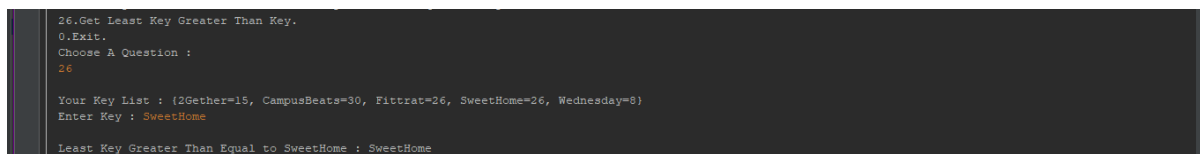
26. Write a Java program to get the least key greater than or equal to the given key. Returns null if there is no such key.

Source Code :

```
protected void GetLeastKeyGreaterThan(TreeMap<String, Integer> Series){
    System.out.print("\nYour Key List : "+Series);
    System.out.print("\nEnter Key : ");
    String Key = in.next();

    System.out.print("\nLeast Key Greater Than Equal to "+Key+" : "+Series.ceilingKey(Key));
    System.out.println();
}
```

Output :

A screenshot of a Java IDE showing the execution of a program. The output text is: 26.Get Least Key Greater Than Key. 0.Exit. Choose A Question : 26 Your Key List : {2Gether=15, CampusBeats=30, Fittrat=26, SweetHome=26, Wednesday=8} Enter Key : SweetHome Least Key Greater Than Equal to SweetHome : SweetHome
