Java Assignment

Project Team:

Aditya Bhandari(LE) : 00110107216
Deepak (LE): 00310107216
Prashant Sharma(LE): 00410107216
Ravi Kumar:(LE) 00510107216
Saiful Hasan (LE): 00710107216
Prithiviraj
Rahul Pandit
Shubham Maurya

How to Run the project:

Requirement:

In order to run this application you must have jvm installed in your system.

How to run the application:

1. Extract the jar file in a folder.
2. Now go to the filder directory from command prompt
3. Now compile the file by **javac Assignment.java**

```
imsaiful@Saiful-PC:~/Desktop/Java$ javac Assignment.java
```

4. File will be competed succesfully and a new .class woll be generated inside same directory.

5. Now run .class File by **java Assignment** command

```
imsaiful@Saiful-PC:~/Desktop/Java$ java Assignment
```

6. Application will ask you to enter the lower limit.Enter any integer number between range -2,147,483,648 and a maximum value of 2,147,483,647.

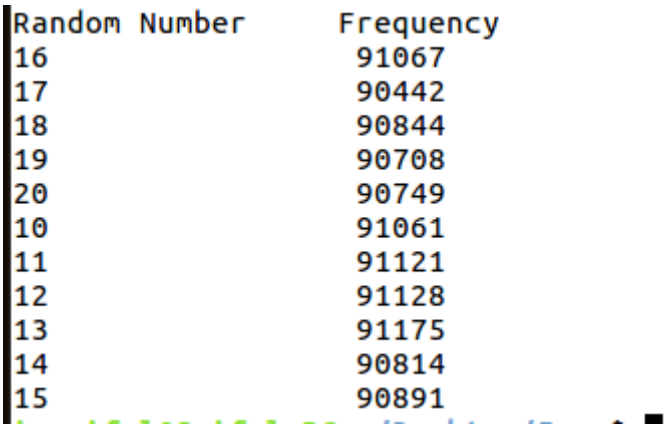
7. Application will ask you to enter the upper limit.Enter any integer number between range lower limit and a maximum value of 2,147,483,647.Do not enter the upper limit smaller then lower limit.

8. Enter how many times you want to generate the random number. It should be greater then zero.

```
imsaiful@Saiful-PC:~/Desktop/Java$ java Assignment
Enter the lower limit
10
Enter the upper limit
20
Enter how many time you want to generate Random Number
1000000
```

9. Now press the enter the command.

10. Random number will geerated between the lower and upper limit. The random value will be store in the HashMap along with how many times it is generated.



Random Number	Frequency
16	91067
17	90442
18	90844
19	90708
20	90749
10	91061
11	91121
12	91128
13	91175
14	90814
15	90891

Result:

The Random number and frequecy are the part of HashMap<Integer, Integer> type where random number is generated between the lower limit and upper limit and Frequency show how many times it is generated.

Statement:

Generate the Random number between two given values. The random number can be generated 2^32 bit time and the range of Random number must be between -2^32 – 2^32. Store the random number in HashMap as key. If a random number is genearted again then increse the value of the key in Map. Print the HashMap contining random number generated and the frequency of each elements.

Input Constraint:

min- minimum value
max -maximum value
n- Numbers of time random number generated

HashMap:

A HashMap in Java is a unordered data structure that maps a key to a value and has a lookup time of O(1), or constant time.

Java HashMap class implements the map interface by using a hashtable. It inherits AbstractMap class and implements Map interface.
The important points about Java HashMap class are:

- A HashMap contains values based on the key.
- It contains only unique elements.
- It may have one null key and multiple null values.

- It maintains no order.

HashMap class Parameters

Let's see the Parameters for java.util.HashMap class.

- K: It is the type of keys maintained by this map.
- V: It is the type of mapped values.

The way you declare a HashMap in Java if you want to map one integer to another is:

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

Methods of Java HashMap class

Method	Description
void clear()	It is used to remove all of the mappings from this map.
boolean containsKey(Object key)	It is used to return true if this map contains a mapping for the specified key.
boolean containsValue(Object value)	It is used to return true if this map maps one or more keys to the specified value.
boolean isEmpty()	It is used to return true if this map contains no key-value mappings.
Object clone()	It is used to return a shallow copy of this HashMap instance: the keys and values themselves are not cloned.
Set entrySet()	It is used to return a collection view of the mappings contained in this map.
Set keySet()	It is used to return a set view of the keys contained in this map.
Object put(Object key, Object value)	It is used to associate the specified value with the specified key in this map.
int size()	It is used to return the number of key-value mappings in this map.
Collection values()	It is used to return a collection view of the values contained in this map.

Code:

```
import java.util.*;
class Assignment
{
	public static void main(String[] args)
	{
		Scanner in=new Scanner(System.in);
		//Hashmap Data structure
		HashMap<Integer,Integer> map=new HashMap<Integer,Integer>();
		System.out.println("Enter the lower limit");
		int min=in.nextInt();
		System.out.println("ENter the upper limit");
		int max=in.nextInt();
		if(min>=max)
		{
			System.out.println("Maximum value shoul be greater then minimum");
			System.exit(0);
		}

		Random r=new Random();
		System.out.println("Enter how many time you want to generate Random Number");
		int n=in.nextInt();
		if(n<=0)
		{
			System.out.println("You should generate the random number at least one
time");
			System.exit(0);
		}
		int i=0;
		while(i<n)
```

```
{


        boolean flag=false;
        //Random function to generate the random value
        int randomNum=r.nextInt((max - min) + 1) + min;
        //check whether the random value is already exist in the hashmap or not
        for(Map.Entry m:map.entrySet())
                {
                        //if random number exist we will increase the vaalue of the key by one
                        if(randomNum==(int)m.getKey())
                        {
                                map.put(randomNum, map.get(randomNum) + 1);
                                flag=true;
                        }


                }
                        //if value does not exist then we will put the value in hashmap along with the initial key value one.
        if(flag==false)
        {
                map.put(randomNum,1);
        }
        i++;
}
        //print the random number and how many times a random number is generated within the range.
        System.out.println("Random Number"+"     "+"Frequency");
        for(Map.Entry m:map.entrySet())
        {
                System.out.println(m.getKey()+"                "+m.getValue());
        }
```

```
    }
}
```

Output:

```
imsaiful@Saiful-PC:~/Desktop$ java Assignment
Enter the lower limit
10
ENter the upper limit
5
Maximum value shoul be greater then minimum
imsaiful@Saiful-PC:~/Desktop$ java Assignment
Enter the lower limit
10
ENter the upper limit
20
Enter how many time you want to generate Random Number
-100
You should generate the random number at least one time
imsaiful@Saiful-PC:~/Desktop$ java Assignment
Enter the lower limit
10
ENter the upper limit
25
Enter how many time you want to generate Random Number
100000000
Random Number     Frequency
10                 6251377
11                 6248257
12                 6254044
13                 6248915
14                 6251361
15                 6251230
16                 6249697
17                 6250579
18                 6249183
19                 6251612
20                 6247710
21                 6246926
22                 6250620
23                 6252016
24                 6247416
25                 6249057
imsaiful@Saiful-PC:~/Desktop$
```

Result:

Study of HashMap structure performed which contain unique key and increase the value.