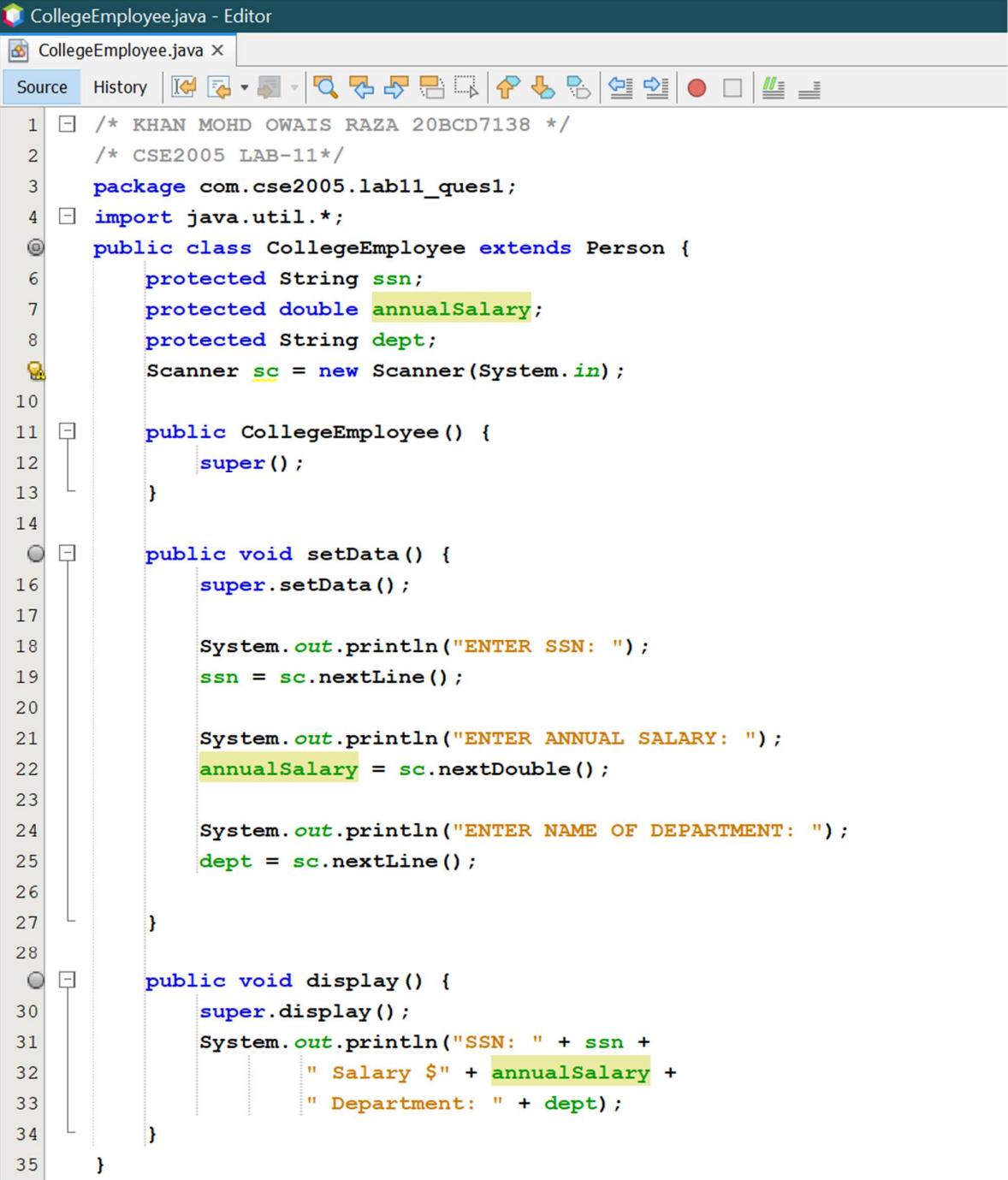


CollegeEmployee.java –



```
CollegeEmployee.java - Editor
CollegeEmployee.java X
Source History | I C F < > S M D T B L E | A D C P | R C | H L

1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-11*/
3  package com.cse2005.lab11_ques1;
4  import java.util.*;
5  public class CollegeEmployee extends Person {
6      protected String ssn;
7      protected double annualSalary;
8      protected String dept;
9      Scanner sc = new Scanner(System.in);
10
11     public CollegeEmployee() {
12         super();
13     }
14
15     public void setData() {
16         super.setData();
17
18         System.out.println("ENTER SSN: ");
19         ssn = sc.nextLine();
20
21         System.out.println("ENTER ANNUAL SALARY: ");
22         annualSalary = sc.nextDouble();
23
24         System.out.println("ENTER NAME OF DEPARTMENT: ");
25         dept = sc.nextLine();
26     }
27
28     public void display() {
29         super.display();
30         System.out.println("SSN: " + ssn +
31                           " Salary $" + annualSalary +
32                           " Department: " + dept);
33     }
34 }
35 }
```

Student.java –

Student.java - Editor

Student.java X

Source History

```
1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-11*/
3  package com.cse2005.lab11_ques1;
4  import java.util.*;
5  public class Student extends Person
6  {
7      private String major;
8      private double gpa;
9      Scanner sc = new Scanner(System.in);
10
11     public Student() {
12         super();
13     }
14
15     @Override
16     public void setData() {
17         super.setData();
18         System.out.println("ENTER THE STUDENT'S MAJOR");
19         major = sc.nextLine();
20         System.out.println("ENTER THE GPA");
21         gpa = sc.nextDouble();
22     }
23     @Override
24     public void display() {
25         super.display();
26         System.out.println("Major: " + major + " GPA: " + gpa);
27     }
28 }
```

Faculty.java –

The screenshot shows a Java code editor window titled "Faculty.java - Editor". The code is a class named Faculty that extends CollegeEmployee. It includes methods for setting data and displaying information, and it uses a Scanner object to read input from System.in. The code is color-coded for syntax highlighting.

```
1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-11*/
3  package com.cse2005.lab11_ques1;
4  import java.util.*;
5  public class Faculty extends CollegeEmployee
6  {
7      private boolean isTenured = false;
8      Scanner sc = new Scanner(System.in);
9      public Faculty() {
10          super();
11      }
12      @Override
13      public void setData() {
14          String temp;
15          super.setData();
16          System.out.println("IS THIS FACULTY MEMBER TENURED - YES or NO");
17          temp = sc.nextLine();
18          if(temp.charAt(0) == 'Y' || temp.charAt(0) == 'y')
19              isTenured = true;
20      }
21      @Override
22      public void display() {
23          super.display();
24          if(isTenured)
25              System.out.println("FACULTY MEMBER IS TENURED");
26          else
27              System.out.println("FACULTY MEMBER IS NOT TENURED");
28      }
29 }
```

GenericQueueDemo.java –

The screenshot shows a Java code editor window titled "GenericQueueDemo.java - Editor". The file contains the following code:

```
1  /* KHAN MOHD OWAIS RAZA 20BCD7138
2   CSE2005 LAB-11
3   */
4  package com.cse2005.lab11_ques1;
5  import java.util.*;
6  interface GenericQueuable<Type>{
7      abstract <Type extends Person> void insertEnd(Type e);
8      abstract void removeBegin();
9      abstract void printQueue();
10     abstract boolean isQueueEmpty();
11 }
12 class GenericQueue<Type> implements GenericQueuable<Type> {
13     int i=0;
14     Object queue[];
15     GenericQueue(Type a[]) {
16         queue=a;
17     }
18     public <Type extends Person>void insertEnd(Type e) {
19         queue[i]=e;
20         System.out.println(e + " IS INSERTED IN THE QUEUE");
21         i++;
22     }
23     public void removeBegin() {
24         System.out.println(queue[0]+ " IS REMOVED FROM THE QUEUE");
25         for(int i=0;i<queue.length-1;i++){
26             queue[i]=queue[i+1];
27         }
28         queue[queue.length-1]=null;
29     }
30     public void printQueue() {
31         if(!isQueueEmpty()) {
32             for(int i=0;queue[i]!=null;i++) {
33                 System.out.println(queue[i]);
34             }
35         }
36         else {
37             System.out.println("QUEUE IS EMPTY");
38         }
39     }
40     public boolean isQueueEmpty() {
41         if(i==0) {
42             return true;
43         }
44     }
}
```

```
45
46     {
47         return false;
48     }
49 }
50 public class GenericQueueDemo{
51     public static void main(String args[]) {
52         Scanner sc=new Scanner(System.in);
53         Person p[]=new Person[10];
54         CollegeEmployee ce[]=new CollegeEmployee[10];
55         Student s[]=new Student[10];
56         GenericQueue g=new GenericQueue(p);
57         GenericQueue g1=new GenericQueue(ce);
58         GenericQueue g2=new GenericQueue(s);
59         g.printQueue();
60         for(int i=0;i<10;i++) {
61             Person c=new Person(i);
62             g.insertEnd(c);
63         }
64         g.removeBegin();
65         g.removeBegin();
66         g.removeBegin();
67         g.removeBegin();
68         g.removeBegin();
69         g.printQueue();
70         g.isEmpty();
71         g1.printQueue();
72         for(int i=0;i<10;i++)
73         {
74             CollegeEmployee c1=new CollegeEmployee(i);
75             g1.insertEnd(c1);
76         }
77         g1.removeBegin();
78         g1.removeBegin();
79         g1.removeBegin();
80         g1.removeBegin();
81         g1.removeBegin();
82         g1.printQueue();
83         g1.isEmpty();
84         g2.printQueue();
85         for(int i=0;i<10;i++) {
86             Student s2=new Student(i);
87             g2.insertEnd(s2);
88         }
89         g2.removeBegin();
90         g2.removeBegin();
91         g2.removeBegin();
92         g2.removeBegin();
93         g2.removeBegin();
94         g2.printQueue();
95         g2.isEmpty();
96         sc.close();
97     }
98 }
```

Output console –

```
Output - Run (Lab11Question1) - Editor
Output - Run (Lab11Question1) X

cd C:\Users\Owais\Documents\NetBeansProjects\Lab11Question1; "JAVA_HOME=
Scanning for projects...

-----< com.cse2005:Lab11Question1 >-----
Building Lab11Question1 1.0-SNAPSHOT
-----[ jar ]-----

--- maven-resources-plugin:2.6:resources (default-resources) @ Lab11Question1 ---
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory C:\Users\Owais\Documents\NetBeansProjects\Lab11Question1\src\main\resources

--- maven-compiler-plugin:3.1:compile (default-compile) @ Lab11Question1 ---
Changes detected - recompiling the module!
Compiling 1 source file to C:\Users\Owais\Documents\NetBeansProjects\Lab11Question1\target\classes

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab11Question1 ---
QUEUE IS EMPTY
Person 1 is INSERTED IN THE QUEUE
Person 2 is INSERTED IN THE QUEUE
Person 3 is INSERTED IN THE QUEUE
Person 4 is INSERTED IN THE QUEUE
Person 5 is INSERTED IN THE QUEUE
Person 6 is INSERTED IN THE QUEUE
Person 7 is INSERTED IN THE QUEUE
Person 8 is INSERTED IN THE QUEUE
Person 9 is INSERTED IN THE QUEUE
Person 10 is INSERTED IN THE QUEUE
Person 1 is REMOVED FROM THE QUEUE
Person 2 is REMOVED FROM THE QUEUE
Person 3 is REMOVED FROM THE QUEUE
Person 4 is REMOVED FROM THE QUEUE
Person 5 is REMOVED FROM THE QUEUE
Person 6 is REMOVED FROM THE QUEUE
Person 7 is REMOVED FROM THE QUEUE
Person 8 is REMOVED FROM THE QUEUE
Person 9 is REMOVED FROM THE QUEUE
Person 10 is REMOVED FROM THE QUEUE
QUEUE IS EMPTY
CollegeEmployee object 1 is INSERTED IN THE QUEUE
CollegeEmployee object 2 is INSERTED IN THE QUEUE
CollegeEmployee object 3 is INSERTED IN THE QUEUE
CollegeEmployee object 4 is INSERTED IN THE QUEUE
CollegeEmployee object 5 is INSERTED IN THE QUEUE
CollegeEmployee object 6 is INSERTED IN THE QUEUE
CollegeEmployee object 7 is INSERTED IN THE QUEUE
CollegeEmployee object 8 is INSERTED IN THE QUEUE
CollegeEmployee object 9 is INSERTED IN THE QUEUE
CollegeEmployee object 10 is INSERTED IN THE QUEUE
CollegeEmployee object 1 is REMOVED FROM THE QUEUE
CollegeEmployee object 2 is REMOVED FROM THE QUEUE
CollegeEmployee object 3 is REMOVED FROM THE QUEUE
CollegeEmployee object 4 is REMOVED FROM THE QUEUE
CollegeEmployee object 5 is REMOVED FROM THE QUEUE
CollegeEmployee object 7 is REMOVED FROM THE QUEUE
CollegeEmployee object 8 is REMOVED FROM THE QUEUE
CollegeEmployee object 9 is REMOVED FROM THE QUEUE
CollegeEmployee object 10 is REMOVED FROM THE QUEUE
QUEUE IS EMPTY
```

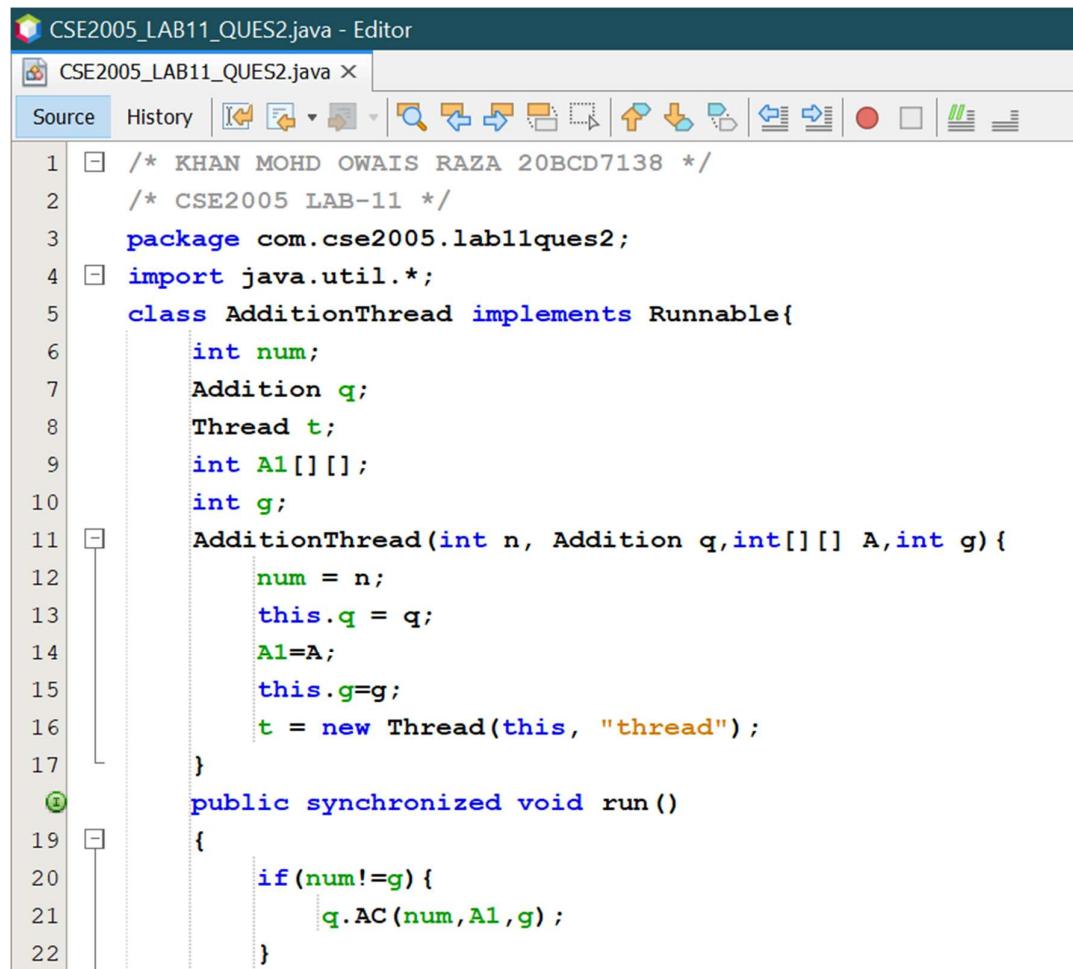
```
Student object 1 is INSERTED IN THE QUEUE
Student object 2 is INSERTED IN THE QUEUE
Student object 3 is INSERTED IN THE QUEUE
Student object 4 is INSERTED IN THE QUEUE
Student object 5 is INSERTED IN THE QUEUE
Student object 6 is INSERTED IN THE QUEUE
Student object 7 is INSERTED IN THE QUEUE
Student object 8 is INSERTED IN THE QUEUE
Student object 9 is INSERTED IN THE QUEUE
Student object 10 is INSERTED IN THE QUEUE
-----
Student object 1 is REMOVED FROM THE QUEUE
Student object 2 is REMOVED FROM THE QUEUE
Student object 3 is REMOVED FROM THE QUEUE
Student object 4 is REMOVED FROM THE QUEUE
Student object 5 is REMOVED FROM THE QUEUE
Student object 6 is REMOVED FROM THE QUEUE
Student object 7 is REMOVED FROM THE QUEUE
Student object 8 is REMOVED FROM THE QUEUE
Student object 9 is REMOVED FROM THE QUEUE
Student object 10 is REMOVED FROM THE QUEUE
```

```
-----  
BUILD SUCCESS  
-----
```

```
Total time: 3.889 s  
Finished at: 2021-12-20T22:20:47+05:30  
-----
```

Q.2]

Using runnable interface –



The screenshot shows a Java code editor window titled "CSE2005_LAB11_QUEST2.java - Editor". The code implements a Runnable interface for thread management.

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-11 */
package com.cse2005.lab11ques2;
import java.util.*;
class AdditionThread implements Runnable{
    int num;
    Addition q;
    Thread t;
    int A1[][];
    int g;
    AdditionThread(int n, Addition q,int[][] A,int g){
        num = n;
        this.q = q;
        A1=A;
        this.g=g;
        t = new Thread(this, "thread");
    }
    public synchronized void run(){
        if(num!=g){
            q.AC(num,A1,g);
        }
    }
}
```

```
23     }
24     else{
25         q.AC();
26     }
27 }
28 class Addition{
29     static int b[]=new int [10];
30     static int i=0;
31     static int sum=0;
32     public synchronized void AC(int num,int[][] A,int n)
33     {
34         for (int i=0; i<n; i++)
35         {
36             b[i]=b[i]+A[num][i];
37         }
38     }
39     public synchronized void AC()
40     {
41         for(int i=0;i<b.length;i++){
42             sum=sum+b[i];
43         }
44     }
45 }
46 public class CSE2005_LAB11_QUEST2{
47     public static void main(String[] args){
48         Scanner sc=new Scanner(System.in);
49         int n=sc.nextInt();
50         try{
51             if(n<5||n>7){
52                 throw new boundingException();
53             }
54             Random rand = new Random();
55             int A[][]=new int[n][n];
56             for (int i =0; i < n; i++)
57             {
58                 for (int j = 0; j< n; j++){
59                     A[i][j]=rand.nextInt(100);
60                 }
61             }
62             Addition q = new Addition();
63             for(int i=0;i<n;i++){
64                 AdditionThread ob=new AdditionThread(i,q,A,n);
65                 ob.t.start();
66             }
67             AdditionThread ob1=new AdditionThread(n,q,A,n);
68             ob1.t.start();
69             while(q.sum==0){
70                 try {
71                     Thread.currentThread().sleep(1000);
72                 }
73                 catch (InterruptedException e) {
74                     Thread.currentThread().interrupt();
75                 }
76             }
77             System.out.println(q.sum);
78         }
79     }
80 }
```

```
79         catch(boundingException e){
80             System.out.println(e);
81         }
82     }
83     class boundingException extends Exception{
84         @Override
85         public String toString(){
86             return "ARRAY CANNOT BE TAKEN WITH THE GIVEN SIZE";
87         }
88     }
89 }
```

Output - Run (CSE2005_LAB11_QUEST2) - Editor

Output - Run (CSE2005_LAB11_QUEST2) X

```
cd C:\Users\Owais\Documents\NetBeansProjects\Lab11Ques2; "JAVA_HOME=C:\\\\Pr
Running NetBeans Compile On Save execution. Phase execution is skipped and
Scanning for projects...

-----< com.cse2005:Lab11Ques2 >-----
[Building Lab11Ques2 1.0-SNAPSHOT]
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab11Ques2 ---
3
ARRAY CANNOT BE TAKEN WITH THE GIVEN SIZE
-----
BUILD SUCCESS
-----
Total time: 6.717 s
Finished at: 2021-12-20T23:10:13+05:30
-----
```

Output - Run (CSE2005_LAB11_QUEST2) - Editor

Output - Run (CSE2005_LAB11_QUEST2) X

```
cd C:\Users\Owais\Documents\NetBeansProjects\Lab11Ques2; "JAVA_HOME=C:\\\\P
Running NetBeans Compile On Save execution. Phase execution is skipped an
Scanning for projects...

-----< com.cse2005:Lab11Ques2 >-----
[Building Lab11Ques2 1.0-SNAPSHOT]
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab11Ques2 ---
5
1192
-----
BUILD SUCCESS
-----
Total time: 7.412 s
Finished at: 2021-12-20T23:11:34+05:30
-----
```

By extending thread class –

```
CSE2005_Lab11_Q2.java - Editor
CSE2005_Lab11_Q2.java X

Source History | I O P v S M F G H D E L R T C B V Z

1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-11 */
3  package com.cse2005.lab11ques2;
4  import java.util.*;
5  class AdditionThread extends Thread{
6      int num;
7      Addition q;
8      Thread t;
9      int A1[][];
10     int g;
11     AdditionThread(int n, Addition q,int[][] A,int g){
12         num = n;
13         this.q = q;
14         A1=A;
15         this.g=g;
16         t = new Thread(this, "thread");
17     }
18     @Override
19     public synchronized void run(){
20         if(num!=g){
21             q.AC(num,A1,g);
22         }
23         else{
24             q.AC();
25         }
26     }
27     class Addition{
28         static int b[]=new int [10];
29         static int i=0;
30         static int sum=0;
31         public synchronized void AC(int num,int[][] A,int n){
32             for (int i=0; i<n; i++){
33                 b[i]=b[i]+A[num][i];
34             }
35         }
36         public synchronized void AC(){
37             for(int i=0;i<b.length;i++){
38                 sum=sum+b[i];
39             }
40         }
41     }
}
```

```
42     public class CSE2005_Lab11_Q2{
43         public static void main(String[] args)
44         {
45             Scanner sc=new Scanner(System.in);
46             int n=sc.nextInt();
47             try{
48                 if(n<5 || n>7){
49                     throw new boundingException();
50                 }
51                 Random rand = new Random();
52                 int A[][]=new int[n][n];
53                 for (int i =0; i < n; i++){
54                     for (int j = 0; j< n; j++){
55                         A[i][j]=rand.nextInt(100);
56                     }
57                 }
58                 Addition q = new Addition();
59                 for(int i=0;i<n;i++){
60                     AdditionThread ob=new AdditionThread(i,q,A,n);
61                     ob.t.start();
62                 }
63                 AdditionThread ob1=new AdditionThread(n,q,A,n);
64                 ob1.t.start();
65                 while(q.sum==0){
66                     try {
67                         Thread.currentThread().sleep(1000);
68                     }
69                     catch (InterruptedException e) {
70                         Thread.currentThread().interrupt();
71                     }
72                 }
73                 System.out.println(q.sum);
74             }
75             catch(boundingException e){
76                 System.out.println(e);
77             }
78         }
79     }
80     class boundingException extends Exception{
81         @Override
82         public String toString(){
83             return "ARRAY CANNOT BE TAKEN WITH THE GIVEN SIZE";
84         }
85     }
86 }
```

Output - Run (CSE2005_Lab11_Q2) - Editor

Output - Run (CSE2005_Lab11_Q2) X

```
cd C:\Users\Owais\Documents\NetBeansProjects\Lab11Ques2; "JAVA_HOME=C:\\\\P
Running NetBeans Compile On Save execution. Phase execution is skipped an
Scanning for projects...

-----< com.cse2005:Lab11Ques2 >-----
[Building Lab11Ques2 1.0-SNAPSHOT]
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab11Ques2 ---
5
621

-----BUILD SUCCESS-----
Total time: 7.018 s
Finished at: 2021-12-20T23:24:35+05:30
-----
```

Output - Run (CSE2005_Lab11_Q2) - Editor

Output - Run (CSE2005_Lab11_Q2) X

```
cd C:\Users\Owais\Documents\NetBeansProjects\Lab11Ques2; "JAVA_HOME=C:\\\\P
Running NetBeans Compile On Save execution. Phase execution is skipped an
Scanning for projects...

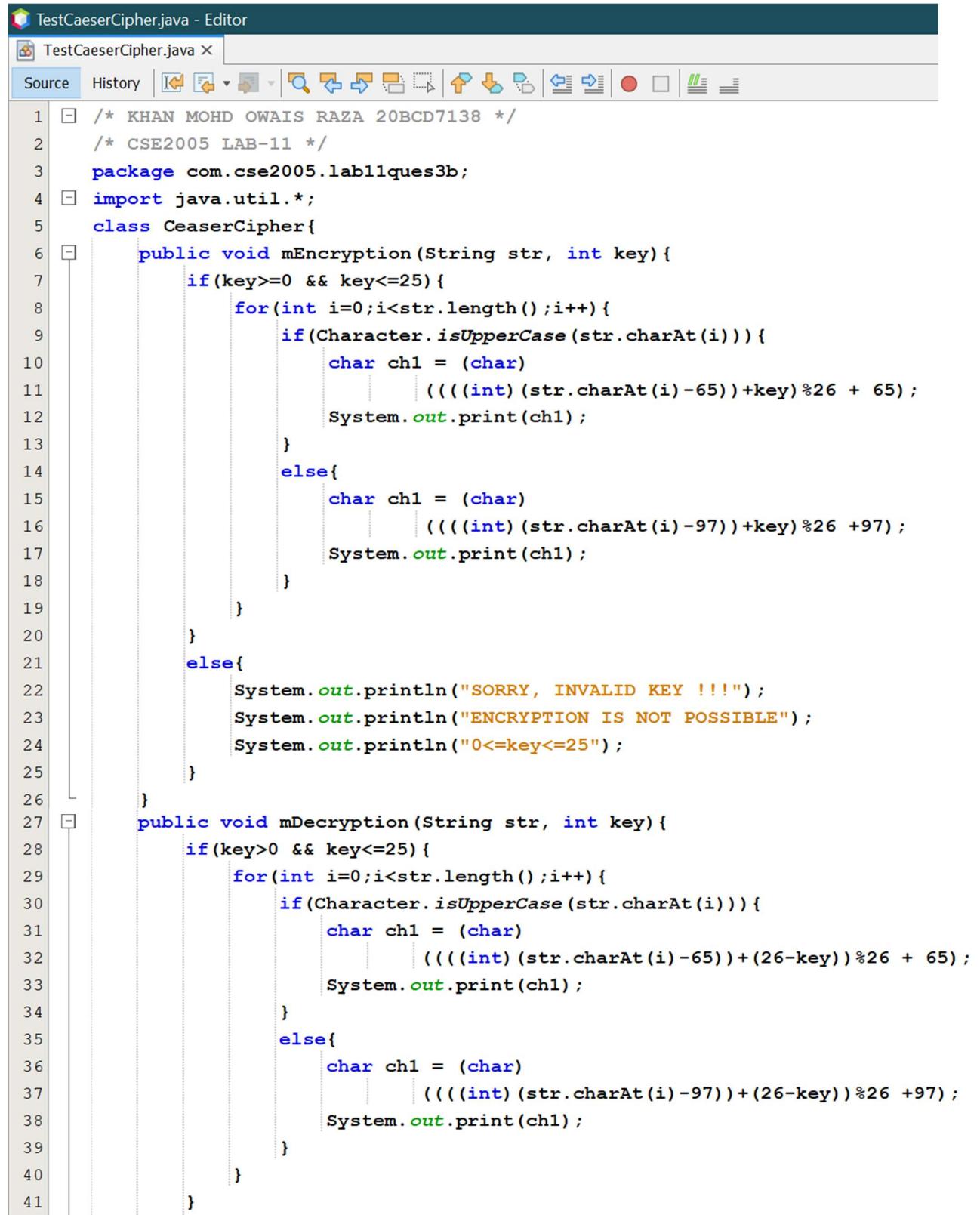
-----< com.cse2005:Lab11Ques2 >-----
[Building Lab11Ques2 1.0-SNAPSHOT]
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab11Ques2 ---
4
ARRAY CANNOT BE TAKEN WITH THE GIVEN SIZE

-----BUILD SUCCESS-----
Total time: 11.892 s
Finished at: 2021-12-20T23:25:33+05:30
-----
```

Q.3]

A)

TestCaesarCipher.java –



The screenshot shows a Java code editor window titled "TestCaesarCipher.java - Editor". The tab bar has "TestCaesarCipher.java x" selected. Below the tabs is a toolbar with various icons. The main area contains the Java code for a Caesar cipher implementation.

```
1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-11 */
3  package com.cse2005.lab11ques3b;
4  import java.util.*;
5  class CeaserCipher{
6      public void mEncryption(String str, int key){
7          if(key>=0 && key<=25){
8              for(int i=0;i<str.length();i++){
9                  if(Character.isUpperCase(str.charAt(i))){
10                     char ch1 = (char)
11                         (((int)(str.charAt(i)-65))+key)%26 + 65);
12                     System.out.print(ch1);
13                 }
14                 else{
15                     char ch1 = (char)
16                         (((int)(str.charAt(i)-97))+key)%26 + 97);
17                     System.out.print(ch1);
18                 }
19             }
20         }
21         else{
22             System.out.println("SORRY, INVALID KEY !!!");
23             System.out.println("ENCRYPTION IS NOT POSSIBLE");
24             System.out.println("0<=key<=25");
25         }
26     }
27     public void mDecryption(String str, int key){
28         if(key>0 && key<=25){
29             for(int i=0;i<str.length();i++){
30                 if(Character.isUpperCase(str.charAt(i))){
31                     char ch1 = (char)
32                         (((int)(str.charAt(i)-65)+(26-key))%26 + 65);
33                     System.out.print(ch1);
34                 }
35                 else{
36                     char ch1 = (char)
37                         (((int)(str.charAt(i)-97)+(26-key))%26 + 97);
38                     System.out.print(ch1);
39                 }
40             }
41         }
42     }
43 }
```

```

42     }
43     else{
44         System.out.println("SORRY, INVALID KEY !!!");
45         System.out.println("ENCRYPTION IS NOT POSSIBLE");
46         System.out.println("0<=key<=25");
47     }
48 }
49 public class TestCaeserCipher{
50     public static void main(String args[]){
51         Scanner in = new Scanner(System.in);
52         System.out.println("ENTER TEXT TO ENCRYPT");
53         String message1 = in.nextLine();
54         System.out.println("ENTER KEY:");
55         int key1 = in.nextInt();
56         CeaserCipher cc = new CeaserCipher();
57         System.out.print("ENCRYPTED MESSAGE: ");
58         cc.mEncryption(message1, key1);
59         System.out.println();
60         System.out.println("ENTER TEXT TO DECRYPT:");
61         String message2 = in.next();
62         System.out.println("ENTER KEY:");
63         int key2 = in.nextInt();
64         System.out.print("DECRYPTED MESSAGE: ");
65         cc.mDecryption(message2, key2);
66     }
67 }

```

Output - Run (TestCaeserCipher) - Editor

Output - Run (TestCaeserCipher) X

cd C:\Users\Owais\Documents\NetBeansProjects\Lab11Ques3B; "JAVA_HOME=C:\\\\Running NetBeans Compile On Save execution. Phase execution is skipped ar Scanning for projects...

-----< com.cse2005:Lab11Ques3B >-----

Building Lab11Ques3B 1.0-SNAPSHOT

-----[jar]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab11Ques3B ---

ENTER TEXT TO ENCRYPT

rama

ENTER KEY:

25

ENCRYPTED MESSAGE: qzlz

ENTER TEXT TO DECRYPT:

qzlz

ENTER KEY:

25

DECRYPTED MESSAGE: rama

BUILD SUCCESS

Total time: 18.927 s

Finished at: 2021-12-21T00:11:42+05:30

B)

CaesarCipher.java –

The screenshot shows a Java code editor with the file "CaeserCipher.java" open. The code implements a Caesar cipher using three threads: Thread1 for encryption, Thread2 for decryption, and Thread3 for generating a key. The CaesarCipher class contains a synchronized method for encryption that iterates through each character of the input string, applying the key to both uppercase and lowercase letters.

```
1  /* KHAN MOHD OWAIS RAZA 20BCD7138*/
2  /* CSE2005 LAB-11 */
3  package com.cse2005.lab11ques3;
4  class Thread1 extends Thread{
5      CaeserCipher encryp;Thread1(CaeserCipher encryp){
6          this.encryp=encryp;
7      }
8      @Override
9      public void run(){
10         encryp.mEncryption("rama",25);
11     }
12 }
13 class Thread2 extends Thread{
14     CaeserCipher decryp;
15     Thread2(CaeserCipher decryp){
16         this.decryp=decryp;
17     }
18     @Override
19     public void run(){
20         decryp.mDecryption("qzlx",25);
21     }
22 }
23 class Thread3 extends Thread{
24     CaeserCipher ukey;
25     Thread3(CaeserCipher ukey){
26         this.ukey=ukey;
27     }
28     @Override
29     public void run(){
30         ukey.unknownkey("gdhrzfnnncanx");
31     }
32 }
33 public class CaeserCipher {
34     synchronized public StringBuffer mEncryption(String Ptext,int key){
35         StringBuffer crypt= new StringBuffer();
36         for(int i=0;i<Ptext.length();i++)
37             if(Character.isUpperCase(Ptext.charAt(i)))
38             {
39                 char ch = (char)
40                     (((int)Ptext.charAt(i) + key- 65) % 26 + 65);
41                 crypt.append(ch);
42             }
43             else{char ch = (char)
44                 (((int)Ptext.charAt(i) + key- 97) % 26 + 97);
45                 crypt.append(ch);
46             }
47         }
48     }
49 }
```

```
43     }
44 }
45 System.out.println("PLAIN TEXT INPUT TO mENCRYPTION IS: "+Ptext+
46         " AND "+key+
47         " AND THE ENCRYPTION IS "+crypt)
48 ;return crypt;
49 }
50 synchronized public StringBuffer mDecryption(String cpt,int key){
51     StringBuffer crypt= new StringBuffer();
52     for(int i=0;i<cpt.length() ;i++){
53         if(Character.isUpperCase(cpt.charAt(i))){
54             char ch = (char)
55                 (((int)cpt.charAt(i) +(26-key)-65) % 26 + 65);
56             crypt.append(ch);
57         }
58         else{
59             char ch = (char)
60                 (((int)cpt.charAt(i) +(26-key)-97) % 26 +97);
61             crypt.append(ch);
62         }
63     }
64 System.out.println("ENCRYPTED TO mDECRYPTION IS: "+cpt+
65         " AND "+key+
66         " AND DECRYPTION IS: "+crypt);
67 return crypt;
68 }
69 synchronized public int unknownkey(String cyptxt){
70     int key=0;
71     for(int i=0;i<26;i++){
72         mDecryption(cyptxt,i);
73         key=i;
74         System.out.println("THE KEY VALUE IS: "+key);
75     }
76     return key;
77 }
78 }
```

BruteForceCaeserCipher.java –

The screenshot shows the NetBeans IDE interface with the title bar "BruteForceCaeserCipher.java - Editor". The code editor window contains the following Java code:

```
1  /* KHAN MOHD OWAIS RAZA 20BCD7138*/
2  /* CSE2005 LAB-11 */
3  package com.cse2005.lab11ques3;
4  public class BruteForceCaeserCipher{
5      public static void main(String[] args) {
6          CaeserCipher cipher=new CaeserCipher();
7          Thread1 x1=new Thread1(cipher);
8          Thread2 x2=new Thread2(cipher);
9          Thread3 x3=new Thread3(cipher);
10         try {x1.start();
11             x1.join();
12             x2.start();
13             x2.join();
14             x3.start();
15             x3.join();
16         }
17         catch (InterruptedException e){
18             e.printStackTrace();
19         }
20     }
21 }
```

Output console –

The screenshot shows the NetBeans IDE Output console window titled "Output - Run (BruteForceCaeserCipher) - Editor". The console output is as follows:

```
cd C:\Users\Owais\Documents\NetBeansProjects\Lab11Ques3; "JAVA_HOME=C:\\Program
Running NetBeans Compile On Save execution. Phase execution is skipped and output
Scanning for projects...

-----< com.cse2005:Lab11Ques3 >-----
Building Lab11Ques3B 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab11Ques3 ---
PLAIN TEXT INPUT TO mENCRYPTION IS: rama AND 25 AND THE ENCRYPTION IS qzrz
ENCRYPTED TO mDECRYPTION IS: qzrz AND 25 AND DECRYPTION IS: rama
ENCRYPTED TO mDECRYPTION IS: gdhrzfnnncanx AND 0 AND DECRYPTION IS: gdhrzfnnncanx
THE KEY VALUE IS: 0
ENCRYPTED TO mDECRYPTION IS: gdhrzfnnncanx AND 1 AND DECRYPTION IS: fcgqyemmbzmw
THE KEY VALUE IS: 1
ENCRYPTED TO mDECRYPTION IS: gdhrzfnnncanx AND 2 AND DECRYPTION IS: ebfpwdxllaylv
THE KEY VALUE IS: 2
ENCRYPTED TO mDECRYPTION IS: gdhrzfnnncanx AND 3 AND DECRYPTION IS: daeowckkzxku
THE KEY VALUE IS: 3
ENCRYPTED TO mDECRYPTION IS: gdhrzfnnncanx AND 4 AND DECRYPTION IS: czdnvbjjywjt
THE KEY VALUE IS: 4
ENCRYPTED TO mDECRYPTION IS: gdhrzfnnncanx AND 5 AND DECRYPTION IS: bycmuaiixvis
THE KEY VALUE IS: 5
```

```
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 6 AND DECRYPTION IS: axbltzhhwuh  
THE KEY VALUE IS: 6  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 7 AND DECRYPTION IS: zwaksyggvtgq  
THE KEY VALUE IS: 7  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 8 AND DECRYPTION IS: yvzjrxffusfp  
THE KEY VALUE IS: 8  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 9 AND DECRYPTION IS: xuyiqweetreo  
THE KEY VALUE IS: 9  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 10 AND DECRYPTION IS: wtxhpvddsqdn  
THE KEY VALUE IS: 10  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 11 AND DECRYPTION IS: vswgouccrpcm  
THE KEY VALUE IS: 11  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 12 AND DECRYPTION IS: urvfntbbqobl  
THE KEY VALUE IS: 12  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 13 AND DECRYPTION IS: tquemsaapnak  
THE KEY VALUE IS: 13  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 14 AND DECRYPTION IS: sptdlrzomzj  
THE KEY VALUE IS: 14  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 15 AND DECRYPTION IS: rosckqyylnyi  
THE KEY VALUE IS: 15  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 16 AND DECRYPTION IS: qnrbjpxxmkxh  
THE KEY VALUE IS: 16  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 17 AND DECRYPTION IS: pmqaiowwljwg  
THE KEY VALUE IS: 17  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 18 AND DECRYPTION IS: olpzhnvvkivf  
THE KEY VALUE IS: 18  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 19 AND DECRYPTION IS: nkoygmujhue  
THE KEY VALUE IS: 19  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 20 AND DECRYPTION IS: mjnxflttigtd  
THE KEY VALUE IS: 20  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 21 AND DECRYPTION IS: limweksshfsc  
THE KEY VALUE IS: 21  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 22 AND DECRYPTION IS: khlvdjrrgerb  
THE KEY VALUE IS: 22  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 23 AND DECRYPTION IS: jgkuciqqfdqa  
THE KEY VALUE IS: 23  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 24 AND DECRYPTION IS: ifjtbhppecpz  
THE KEY VALUE IS: 24  
ENCRYPTED TO mDECRYPTION IS: gdhrzfnncanx AND 25 AND DECRYPTION IS: heisagoodboy  
THE KEY VALUE IS: 25
```

```
-----  
BUILD SUCCESS
```

```
-----  
Total time: 1.474 s  
Finished at: 2021-12-21T00:54:17+05:30
```



BruteForceCaeserCipher.java (using runnable interface) –

```
49     Decryption text = new Decryption();
50     CaesarCipher obj0 = new CaesarCipher(message,0,text);
51     CaesarCipher obj1 = new CaesarCipher(message,1,text);
52     CaesarCipher obj2 = new CaesarCipher(message,2,text);
53     CaesarCipher obj3 = new CaesarCipher(message,3,text);
54     CaesarCipher obj4 = new CaesarCipher(message,4,text);
55     CaesarCipher obj5 = new CaesarCipher(message,5,text);
56     CaesarCipher obj6 = new CaesarCipher(message,6,text);
57     CaesarCipher obj7 = new CaesarCipher(message,7,text);
58     CaesarCipher obj8 = new CaesarCipher(message,8,text);
59     CaesarCipher obj9 = new CaesarCipher(message,9,text);
60     CaesarCipher obj10 = new CaesarCipher(message,10,text);
61     CaesarCipher obj11 = new CaesarCipher(message,11,text);
62     CaesarCipher obj12 = new CaesarCipher(message,12,text);
63     CaesarCipher obj13 = new CaesarCipher(message,13,text);
64     CaesarCipher obj14 = new CaesarCipher(message,14,text);
65     CaesarCipher obj15 = new CaesarCipher(message,15,text);
66     CaesarCipher obj16 = new CaesarCipher(message,16,text);
67     CaesarCipher obj17 = new CaesarCipher(message,17,text);
68     CaesarCipher obj18 = new CaesarCipher(message,18,text);
69     CaesarCipher obj19 = new CaesarCipher(message,19,text);
70     CaesarCipher obj20 = new CaesarCipher(message,20,text);
71     CaesarCipher obj21 = new CaesarCipher(message,21,text);
72     CaesarCipher obj22 = new CaesarCipher(message,22,text);
73     CaesarCipher obj23 = new CaesarCipher(message,23,text);
74     CaesarCipher obj24 = new CaesarCipher(message,24,text);
75     CaesarCipher obj25 = new CaesarCipher(message,25,text);
76     obj0.t.start();
77     obj1.t.start();
78     obj2.t.start();
79     obj3.t.start();
80     obj4.t.start();
81     obj5.t.start();
82     obj6.t.start();
83     obj7.t.start();
84     obj8.t.start();
85     obj9.t.start();
86     obj10.t.start();
87     obj11.t.start();
88     obj12.t.start();
89     obj13.t.start();
90     obj14.t.start();
91     obj15.t.start();
92     obj16.t.start();
93     obj17.t.start();
94     obj18.t.start();
95     obj19.t.start();
96     obj20.t.start();
97     obj21.t.start();
98     obj22.t.start();
99     obj23.t.start();
100    obj24.t.start();
101    obj25.t.start();
102}
103}
```

Output console –

```
Output - Run (BruteForceCaeserCipher) - Editor
Output - Run (BruteForceCaeserCipher) X

cd C:\Users\Owais\Documents\NetBeansProjects\Lab11Ques3; "JAVA_HOME=C:\\\\F
Running NetBeans Compile On Save execution. Phase execution is skipped an
Scanning for projects...

-----< com.cse2005:Lab11Ques3 >-----
Building Lab11Ques3B 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab11Ques3 ---
gdhrzfnncanx Thread 0
heisagoodboy Thread 25
ifjtbhppecpz Thread 24
jgkuciqqfdqa Thread 23
khldvjrrgerb Thread 22
limweksshfsc Thread 21
olpzhnvvkivf Thread 18
mijnxflltigd Thread 20
nkoygmuujhue Thread 19
pmqaiowwljwg Thread 17
qnrbjpxxmxh Thread 16
rosckqyynlyi Thread 15
sptdlrzzomzj Thread 14
tquemsaapnak Thread 13
urvfntbbqobl Thread 12
vswgouccrpcm Thread 11
wxhpvddsqdn Thread 10
xuyiqweetreo Thread 9
yvzjrxffusfp Thread 8
zwaksyggvtgq Thread 7
czdnvbjjywjt Thread 4
bycmuaiixvis Thread 5
axbltzhhwuh Thread 6
daeowckkzxku Thread 3
ebfpxdllaylv Thread 2
fcgqyemmbzmw Thread 1
-----
BUILD SUCCESS
-----
Total time: 1.505 s
Finished at: 2021-12-21T01:16:10+05:30
-----
```

Q.4]

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-11 */
package com.cse2005.lab11ques4;
import java.util.*;
interface MyMap<Key,Value>{
    abstract void add(Key K, Value V);
    abstract Value remove(Key K);
    abstract int size();
    abstract boolean isEmpty();
    abstract ArrayList<Key>keys();
    abstract void print();
}
class MyMapImp<Key, Value>implements MyMap<Key,Value>{
    ArrayList<Key> A1;
    ArrayList<Value> A2;
    MyMapImp(ArrayList<Key> A1, ArrayList<Value> A2){
        this.A1=A1;
        this.A2=A2;
    }
    public void add(Key K, Value V){
        System.out.println("KEY: "+K+
            "VALUE: " +V+
            "ARE ADDED TO THE MAP");
        A1.add(K);
        A2.add(V);
    }
    public Value remove(Key K){
        Value V=null;
        for(int i=0;i<A1.size();i++){
            if(A1.get(i)==K){
                V=A2.get(i);
                System.out.println("KEY: "+K+
                    "VALUE: " +V+
                    "ARE REMOVED FROM THE MAP");
                A1.remove(i);
                A2.remove(i);
            }
        }
        return V;
    }
    public int size(){
        return A1.size();
    }
    public boolean isEmpty(){
        if(A1.size()==0){
            return true;
        }
        else{
            return false;
        }
    }
    public ArrayList<Key>keys(){
        return A1;
    }
    public void print(){
        for(int i=0;i<A1.size();i++){
            System.out.println("KEY: "+A1.get(i));
        }
    }
}
```

```

        System.out.println("VALUE: "+A2.get(i));
    }
}
}

public class MyMapTest{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        Map<String, Integer> M1 = new HashMap<String, Integer>();
        Map<Integer, Double> M2 = new HashMap<Integer, Double>();
        for(int i=0;i<5;i++){
            System.out.println("ENTER THE DATA FOR MAP-1: ");
            String s = sc.nextLine();
            int n = sc.nextInt();
            M1.put(s,n);
            System.out.println("ENTER THE DATA FOR MAP-2: ");
            int n1 = sc.nextInt();
            double n2 = sc.nextDouble();
            String s1 = sc.nextLine();
            M2.put(n1,n2);
        }
        ArrayList<Integer> A1 = new ArrayList<Integer>(M1.values());
        ArrayList<String> A2 = new ArrayList<String>(M1.keySet());
        MyMapImpl<String, Integer> MyMapImpl = null;
        MyMapImpl<String, Integer> String = null;
        MyMapImpl<String, Integer> MMI = MyMapImpl<String, Integer>(A1,A2);
        System.out.println("ENTER THE VALUES TO BE ADDED TO MAP-1: ");
        String s = sc.nextLine();
        int n3 = sc.nextInt();
        MMI.add(s,n3);
        MMI.remove(s);
        System.out.println(MMI.size());
        System.out.println(MMI.keys());
        if(MMI.isEmpty()){
            System.out.println("MAP DO NOT HAVE ANY ELEMENT");
        }
        else{
            System.out.println("MAP HAS"+A1.size()+"ELEMENTS");
        }
        for(int i=0;i<A1.size();i++){
            MMI.remove(A2.get(i));
        }
        if(MMI.isEmpty()){
            System.out.println("MAP DO NOT HAVE ANY ELEMENT");
        }
        else{
            System.out.println("MAP HAS"+A1.size()+"ELEMENTS");
        }
        ArrayList<Integer> A3 = new ArrayList<Integer>(M2.keySet());
        ArrayList<Double> A4 = new ArrayList<Double>(M2.values());
        MyMapImpl<Integer, Double> MMI2;
        MMI2 = new MyMapImpl<Integer, Double>(A3,A4);
        System.out.println("ENTER THE VALUES TO ADDED TO MAP-2: ");
        int n7 = sc.nextInt();
        double n8 = sc.nextDouble();
        MMI2.add(n7,n8);
        MMI2.remove(n7);
        System.out.println(MMI2.size());
        System.out.println(MMI2.keys());
        if(MMI2.isEmpty()){

```

```
        System.out.println("MAP DO NOT HAVE ANY ELEMENT");
    }
    else{
        System.out.println("MAP HAS "+A1.size()+" ELEMENTS");
    }
    for(int i=0;i<A3.size();i++){
        MMI2.remove(A3.get(i));
    }
    if(MMI2.isEmpty()){
        System.out.println("MAP DO NOT HAVE ANY ELEMENT");
    }
    else{
        System.out.println("MAP HAS "+A1.size()+" ELEMENTS");
    }
}
```

Output console –

```
Output - Run (MyMapTest) - Editor
Output - Run (MyMapTest) X

cd C:\Users\Owais\Documents\NetBeansProjects\Lab11Ques4; "JAVA_HOME=C:\\\\
Scanning for projects...

-----< com.cse2005:Lab11Ques4 >-----
Building Lab11Ques4 1.0-SNAPSHOT
-----[ jar ]-----

--- maven-resources-plugin:2.6:resources (default-resources) @ Lab11Ques4 ---
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory C:\Users\Owais\Documents\NetBeansPro

--- maven-compiler-plugin:3.1:compile (default-compile) @ Lab11Ques4 ---
Changes detected - recompiling the module!
Compiling 2 source files to C:\Users\Owais\Documents\NetBeansProjects\La

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab11Ques4 ---
ENTER THE DATA FOR MAP-1:
Raza
125
ENTER THE DATA FOR MAP-2:
23
45.40
ENTER THE DATA FOR MAP-1:
Michael
58
ENTER THE DATA FOR MAP-2:
21
22.67
ENTER THE DATA FOR MAP-1:
Brock
67
ENTER THE DATA FOR MAP-2:
82
182.93
ENTER THE DATA FOR MAP-1:
Ronaldo
7019
```

```
ENTER THE DATA FOR MAP-2:  
8291  
638.21  
ENTER THE DATA FOR MAP-1:  
Alen  
7910  
ENTER THE DATA FOR MAP-2:  
721  
829.89  
ENTER THE VALUES TO BE ADDED TO MAP-1:  
John  
0829  
KEY: John VALUE: 829 ARE ADDED TO THE MAP  
KEY: John VALUE: 829 ARE REMOVED FROM THE MAP  
5  
[Alen, Brock, Michael, Ronaldo, Raza]  
MAP HAS 5 ELEMENTS  
KEY: Alen VALUE: 7910 ARE REMOVED FROM THE MAP  
KEY: Michael VALUE: 58 ARE REMOVED FROM THE MAP  
KEY: Raza VALUE: 125 ARE REMOVED FROM THE MAP  
MAP HAS 2 ELEMENTS  
ENTER THE VALUES TO BE ADDED TO MAP-2:  
198  
KEY: 198 VALUE: 3821.21 ARE ADDED TO THE MAP  
6  
[721, 82, 8291, 21, 23, 198]  
MAP HAS 2 ELEMENTS  
KEY: 721 VALUE: 829.89 ARE REMOVED FROM THE MAP  
KEY: 8291 VALUE: 638.21 ARE REMOVED FROM THE MAP  
KEY: 23 VALUE: 45.4 ARE REMOVED FROM THE TOP  
MAP HAS 2 ELEMENTS
```

```
BUILD SUCCESS
```

```
Total time: 5.717 s  
Finished at: 2021-12-21T16:13:50+05:30
```

```
<
```

Output console using gdb compiler –

```
ENTER THE DATA FOR MAP-1:  
Raza  
125  
ENTER THE DATA FOR MAP-2:  
23  
45.40  
ENTER THE DATA FOR MAP-1:  
Michael  
58  
ENTER THE DATA FOR MAP-2:  
21  
22.67  
ENTER THE DATA FOR MAP-1:  
Brock  
67  
ENTER THE DATA FOR MAP-2:  
82  
182.93  
ENTER THE DATA FOR MAP-1:  
Ronaldo  
7019  
ENTER THE DATA FOR MAP-2:  
8291  
638.21  
ENTER THE DATA FOR MAP-1:  
Alen  
7910  
ENTER THE DATA FOR MAP-2:  
721  
829.89  
ENTER THE VALUES TO BE ADDED TO MAP-1:  
John  
0829  
KEY: John VALUE: 829 ARE ADDED TO THE MAP  
KEY: John VALUE: 829 ARE REMOVED FROM THE MAP  
5  
[Alen, Brock, Michael, Ronaldo, Raza]  
MAP HAS 5 ELEMENTS  
KEY: Alen VALUE: 7910 ARE REMOVED FROM THE MAP  
KEY: Michael VALUE: 58 ARE REMOVED FROM THE MAP  
KEY: Raza VALUE: 125 ARE REMOVED FROM THE MAP  
MAP HAS 2 ELEMENTS  
ENTER THE VALUES TO BE ADDED TO MAP-2:  
198  
KEY: 198 VALUE: 3821.21 ARE ADDED TO THE MAP  
6  
[721, 82, 8291, 21, 23, 198]  
MAP HAS 2 ELEMENTS  
KEY: 721 VALUE: 829.89 ARE REMOVED FROM THE MAP  
KEY: 8291 VALUE: 638.21 ARE REMOVED FROM THE MAP  
KEY: 23 VALUE: 45.4 ARE REMOVED FROM THE TOP  
MAP HAS 2 ELEMENTS  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```