METHODS

EX.NO:6a

AIM:

Write a program to find the minimum number of days or which Mathew will have a rest.

ALGORITHM:

1. Start the program.
2. Declare the class
3. Declare scanner to get the value.
4. Declare the values.
5. Apply while condition
6. Store the result.
7. Terminate the program.

PROGRAM:

```
Import java.util.*;
Public class class 3169383780Z9
{
 Public static void main [string()args]
 {
  Scanner str = new scanner(system.in);
  int P=3; x,s=0,n=str.nextInt();
 While (n!=0)
 {
  x=str.nextInt();
```

INPUT:

4

1 3 20

1 0

0 0 11 0000 10

OUTPUT:

2

8

```
if (x==0 || x==P && x!=3))
{
  s++;
  P=0;
}
else
p=(x==3)? 3-P:x;
n--;
}
System.out.println(s);
if(n!=0)
System.out.println(" static Scanner sc = new scanner
(system.in); public static in gotogym() system.out.
println(obj.gotogym());
classAdmin obj = new classAdmin(); ");
}
}
```

RESULT:

Thus the program to find the minimum number of days on which Mathew will have rests.

EX.NO: 6.b

AIM:

To write a program to find the minimum possible data consumption factor of the resulting network.

ALGORITHM:

1. Start the program
2. Declare the class
3. Declare scanner to get the values.
4. Apply the if and else condition.
5. Store the result.
6. Terminate the program.

PROGRAM:

```
import java.io.*;
import java.util.*;
Public class class 3469383780229
{
Public static void network computs (int N, int M)
{
if(M< N-1 || M>((N)*(N+1))/2)
System.out.println("-1");
else if(N==1 && M<=1)
System.out.println(M);
else if(N==2 && M==1)
System.out.println(M);
else{
```

INPUT:

5 6 25 40

OUTPUT:

2 3

```
if (M<=N+1)
System.out.println("2");
else if((M-(N+1)<=N-2)
system.out.println("3");
else{
long ans = (2*M+N-1)/N-1;
system.out.println(ans);
}
}
}
Public static Void Main (string [] args)
{
int x,y;
scanner sc = new scanner(system.in);
x = sc.nextInt();
y = sc.nextInt();
network compute (x,y);
}
}
```

RESULT:

Thus the program to find the minimum possible data consumption factor for the resulting network was verified.

EX.NO: 6C

AIM:

Write a program to find the sum of digits in a number using functions.

ALGORITHM:

1. Start the program.
2. Declare the class
3. Declare the scanner to get values.
4. Apply the for condition
5. Store the result.
6. Terminate the program.

PROGRAM:

```
import java.io.*;
import java.util.*;
Public class class 3769383780229
{
 Public Static void main (string [] args)
 {
  scanner sc = new scanner (system.in);
  int n, sum = 0;
   n = sc.nextInt();
  for (sum = 0 ; r != 0, r = r/10)
  {
   sum = sum + r%10;
  }
  system.out.println(sum);
```

INPUT:

1 2 3 4 5 1 2 4 4 2 1

OUTPUT:

15 14

```
String s="public static prnts(int n)
return sum;
System.out.println(s(n));";
}
}
```

RESULT:

Thus the program to find the sum of digits in a number using functions was verified.

STRING

EX.NO: 7a

AIM:

To write a program to find the number to counting only digits 0's and 1's.

ALGORITHM:

1. Start the program.
2. Declare the class.
3. Declare the scanner to get the values.
4. Apply if else condition.
5. Store the program.
6. Teminate the program.

PROGRAM:

```
import java.io.*;
import java.util.scanner;
Public class class 3769383f8029
{
Public static void Main (string [] args)
{
 scanner sc = new scanner(system.in);
 string str;
 str.sc nextline();
 char [] an = str.tocharArray();
 int c1 = 0, c2 = 0;
 for(int i=0; i<str.length(); i++)
 {
   if(arr[i]=='1')
 {
```

INPUT :

101 11001

OUTPUT:

Yes No

```
c1++;
}
else
{
c2++;
}
}
if(c2==str.length()-1||c1==str.length()-1)
{
system.out.println("Yes");
}
else
{
system.out.Println("No");
}
}
}
```

RESULT:

Thus the program to find the number to counting only digits 0's and 1's was verified.

EX.NO: 7b

AIM:

To write a program to find the resulting string.

ALGORITHM:

1. Start the program
2. Declare the class
3. Declare scanner to get values.
4. Store the result
5. Terminate the program.

PROGRAM:

```
import java.util.*;
Public class class 3169383718029 {
Public static void main (string[] args)
{
scanner sc = new scanner (system.in);
string s;
  s= sc.nextline()
string st = s. replace ("Travis", "Batman");
system. out. println(st);
}
}
```

RESULT:

Thus the program to find the result of the string was verified.

INPUT:

Travis scottmeal is awesome

OUTPUT:

Batman scott meal is awesome

EX:NO:7C

AIM:

Write a program to find out the minimum number of moves required to make the given string s a good string.

ALGORITHM:

1. Start the program.
2. Declare the class.
3. Declare the scanner
4. Apply for condition.
5. Apply string condition
6. Store the result.
7. Terminate the program.

PROGRAM:

```
import java.util.*;
Public class class 3769383?8029
{
Public static void main (string[] args)
{
scanner sc = new scanner (system.in);
string s = sc.next(); int count=0;
char[] str = s to charArray();
char[] str = s to charArray();
Array s. sort (str);
for(int j-0 ; j<str.length-1 ;j++)
{
  if(str[j] = str[j+1])
{
```

INPUT:

Plausible

tantalizingly

OUTPUT:

1

5

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

RESULT:

Thus the program to find out the maximum number of moves required to make the given string s a good string.

INHERITANCE AND INTERFACE

EX:NO:8a

AIM:

Write a program to find that the skill level of ivan and joseph became exactly PQ respectively or some day.

ALGORITHM:

1. start the program
2. Declare the class.
3. Declare the scanner.
4. store the result.
5. Terminate the program.

PROGRAM:

```
import java.util.*;
Public class class 3769383780 29
{
Public static Void main (string [] args)
{
 Scanner sc = new scanner (system.in);
 Int A = sc.nextInt ();
 Int B = sc.nextInt();
 Int P = sc.nextInt();
 Int Q = sc.nextInt();
 system.out.println(solve A,B,P,Q)? "YES!" "NO");
 String s = "static scanner sc = new scanner (system.in);
      static class work int a, b, P, q; static class
      schedule extends work schedule obj - newschedule;
```

INPUT:

1 2 1 2 1 2 3 2

OUTPUT:

YES NO

```
obj.input(); obj.planc(); "";

}
static boolean solve(int A, int B, int P, int Q)
{
return P%.A ==0 && Q%.B ==0 && Math.abs(P/A - Q/B)<=1;

}
}
```

RESULT:

Thus the program to find the skill level of Ivan and Joseph become exactly PQ respectively or some day was verified successfully.

EX.NO: 8b

AIM:

Write a program to determine whether he can go to cell t using the transportation system.

ALGORITHM:

1. Start the program.
2. Declare the class.
3. Declare the scanner.
4. Apply if and else condition
5. Store the result.
6. Terminate the program.

PROGRAM:

```
Import java.util.*;
Public class class 3769383T8029
{
Public static void main (String[] args)
{
Scanner Input = new scanner (System.in);
String str = " interface Transport void transportation(intt,
    long[]a); class Portal implements Transport public void
    transportation (int t, long[] a) portal travel = new
    portal (); travel. transportation (travel.t, travel.a);";
Int n, t;
n = Input.nextInt();
t = Input.nextInt();
If((n*t)>30)
```

INPUT:

8 4

1 2 1 2 1 2 1

10 3

8 3 5 4 2 3 2 2 1

OUTPUT:

YES

NO

```
System.out.println("YES");
else
System.out.println("No");
}
}
```

RESULT:

Thus the program to determine whether he can go to call a using the transportation system was verified.

EX.NO: 8C

AIM:

Write a program to find out exactly what two errors he corrected.

ALGORITHM:

1. Start the program.
2. Declare the class.
3. Declare the scanner.
4. Apply for condition.
5. Store the result.
6. Terminate the program.

PROGRAM:

```
import java.util.*;
Public class class 3769383F8029
{
Public static void Main (string[] args)
{
scanner sc = new scanner (system.in);
Int n = sc.nextInt();
Int a=0, b=0, c=0;
Int[] arr1 = new Int[n];
Int[] arr2 = new Int[n-1];
Int[] arr3 = new Int[n-2];
for (Int i=0; i<n; i++)
{
  arr1[i] = sc.nextInt();
  a+ = arr1[i];
}
```

```
for(int i=0; i<n-1; i++)
{
arr2[i] = sc.nextInt();
b+= arr2[i];
}
for(int i=0; i<n-2; i++)
{
arr3[i] = sc.nextInt();
c+= arr3[i];
}
system.out.println(a-b);
system.out.println(b-c);
String s = "static scanner sc= new scanner (system.in); interface
Prob void solver (int n); static class complication implements
prob public void solver (int n) complication obj= new
complation(); obj.solver (obj.n);";
}
}
```

RESULT:

Thus the program to find out exactly what two errors he corrected was executed successfully.

INPUT:

5

1 5 8 123 7

123 7 5 1

OUTPUT: 5 1 7

8

123

3

84 30 9

9 84

9

30

84

EXCEPTIONAL HANDLING

Ex. No: 9a

AIM:

To write a program to find the distribute all beans.

ALGORITHM:

1. Start the program.
2. Declare the class.
3. Declare the scanner.
4. Apply if and else condition.
5. Store the result.
6. Terminate the program.

PROGRAM:

```
import java.util.*;
Public class class 3169383780229
{
 Public static Void Main (string[] args)
{
 Scanner sc = new Scanner (System.in);
 try{
 Int r = sc.nextInt();
 Int b = sc.nextInt();
 Int d = sc.nextInt();
 If (Math.max(r,b) <= Math.min(r,b) * (d+1))
 {
  System.out.println("YES");
 }
```

```
else
system.out.println("No");
}
catch(Exception e)
{
system.out.println("Incomplete Details");
}
string s="|| catch (org r,b,d);";
}
```

RESULT:

Thus the program to find the distribute all beans was verified successfully.

INPUT:

2 4 3

10 3

OUTPUT:

Yes

Incomplete Details

EX.NO: 9b

AIM:

Write a program to place w+b dominoes on the board if you can place dominoes both horizontally and vertically.

ALGORITHM:

1. Start the program.
2. Declare the class.
3. Declare scanner to get values.
4. Apply try condition.
5. Store the result.
6. Terminate the program.

PROGRAM:

```
import java.util.*;
Public class class 3169383780299
{
Public static Void main(String[] args)
{
Scanner sc = new scanner (System.in);
String s= "Int n, k1, k2, w, b;";
try {
Int n = 2*sc.nextInt();
Int ws = sc.nextInt() + sc.nextInt();
Int w = 2* sc.nextInt(), b= 2* sc.nextInt();
System.out.println(w<=ws && b<=n-ws? "YES": "NO");}
catch (NoSuchElement Exception e){
System.out.println("Few Inputs Missing");
```

}

}

}

RESULT:

Thus the program to place w*h dominoes on the board if you can place dominoes both horizontally and vertically was verified successfully.

INPUT:

2 2

5 4 3

3 4

2

OUTPUT:

NO

Few Inputs Missing.

EX.NO:9C

AIM:

To write a program to find the number of stacks always remains n stacks don't disappear when they have 0 blocks.

ALGORITHM:

1. Start the program.
2. Declare the class.
3. Declare the scanner.
4. Apply try condition.
5. Apply if and else condition.
6. Store the result.
7. Terminate the program.

PROGRAM:

```
Import java.util.*;
Public class class 3769383780229
{
Public static void main (string [] args)
{
 Scanner Input = new Scanner (System.in);
 try {
   Int n= Input.nextInt();
   Int [] ar = new Int [n];
   Int tag=0;
   for(Int i=0 ;i<n; i++){
    ar[i] = Input.nextInt();
```

```
}
if (arr[0] >= arr[1])
  tag++;
if (tag>0)
  system.out.println ("YES");

else
  system.out.println ("NO");

}
catch (Exception e)

{
system.out.println ("Few Details missing");

}
}
```

RESULT:

Thus the program to find the number of stacks always remains n stacks don't disappear when they have 0 blocks was verified successfully.

INPUT:

4

1000000000 1000000000 1000000000

1000000000

10

1 1 2 25 4 6 8

OUTPUT:

YES

Few details

GENERIC AND COLLECTIONS

EX.NO: 10a

AIM:

Write a program to determine if a number n is legit.

ALGORITHM:

1. Start the program.
2. Declare the public class.
3. Declare the scanner.
4. Declare the static boolean.
5. Apply if and else condition.
6. Store the result.
7. Terminate the program.

PROGRAM:

```
Import java.util.*;
Public class class 376938378029
{
Public static void main (string [] args)
{
scanner sc = "Import new scanner (system.in);
int n = sc.nextInt()
String s = " Import java.util.Hashset; Public static
    int sum (int n) Public static boolean is Legit (int n)
    Hashset < Integer > set = new Hashset <Integer > (); ";
system.out.println (isHappy (n));
}
Public static boolean isHappy (int n)
{
```

```
if (n==1 || n==7)
    return true;
if (n<10)
{
return false;

}
int sum=0;
while (n>0)
{
sum+= (n%10)*(n%10);
n=n/10;
}
return isHappy(sum);
}
}
```

RESULT:

Thus the program to determine if a number n is Legit was executed successfully.

INPUT:

19 270

OUTPUT:

true false

EX.NO: 10 b

AIM:

To write a program to determine an integer denoting the minimum number of plates.

ALGORITHM:

1. Start the program.
2. Declare the string function.
3. Declare the scanner
4. Store the result.
5. Terminate the program.

PROGRAM:

```
import java.util.*;
Public class class 3769383780029
{
 Public static void main (string [] args)
{ string j = "ArrayList < Long > li = new ArrayList < Long >();
   Collections.sort(li); for (int j = li.size()-1; j>=0; j--)";
 Scanner s = new scanner (system.in);
 Int n = S.nextInt();
 long m = S.nextlong();
 long aaa[] = new long [n];
 for(int P=0; i< n; P++)
 aaa[i] = S.nextLong();
 Array.sort (aaa);
 long cnt = 0;
 long tot = 0;
```

```
for(int i=n-1 ; i>=0 ; i--)
{
tot += arr[i];
cnt++;
if(tot >m)
break;
}
if (tot<m)
cnt=-1;
System.out.println(cnt);
}
}
```

RESULT:

Thus the program to determine an integer denoting the minimum number of plates was executed successfully.

INPUT:

4 7

1 2 3 4

OUTPUT:

2

5 9

1 2 4 11

5

EX.NO: 10 c

AIM:

To write a program to determine Timon is happy or not.

ALGORITHM:

1. Start the program.
2. Declare the scanner.
3. Apply the condition.
4. Store the result.
5. Terminate the program.

PROGRAM:

```
import java.util.*;
Public class class 3169383718099
{
Public static Void Main (String [] args)
{
scanner sc = new scanner (System.in);
Int n = sc.nextInt();
Int [] a = new Int [n];
Int b,c, flag = 0;
for (Int i = 0; i<n; i++) {
 a[i] = sc.nextInt();
}
for (Int i=0; i<n-1; i++)
 for (Int j = i+1; j<n; j++)
{
 If (a[i] != a[j])
 {
   b = a[i]-1;
   c = a[j]-1;
```

```
if(a[b] == a[c])
{
flag =1;
break;
}
}
}
if(flag==0)
System.out.println("Timon sad");
else {
System.out.println("Truly Legit");}
String s = "HashMap<Integer, Arraylist<Integer>>> mp; Arraylist
<Integer>list=mp.get(val2); list=new ArrayList<>();
if(list.contains(val1)==false)";
}
}
```

RESULT:

Thus the program to determine whether Timon is Happy or Not was verified successfully.

INPUT:

4

1 1 2 3

4

2 1 3 3

OUTPUT:

Truly Legit

Timon Sad.