**1.QUESTION**

Public class charecterpattern{

 Public static void printcharecterpattern ( int num){

 Int i , j,value = 1;

Char ch = ‘a’;

Char print = ch;

For(j=0;j<=i;j++)

System.out.print( (ch ++ ));

System.out.println(  \*\*  );

}

}

}

**1.(1)TESTCASE 1:**

**Input:**

5

Expected output:

a

ab

abc

abcd

abcde

**testcase 2:**

input:

1

Output:

a

**2.QUESTION**

Public class Maxarrayelement

{

Public static int[] sortArray{int[] arr}

{

Int i=0,j=0,temp=0,index=0;

For(i=0;i<arr.length;i++)

{

For(j=i+1;j<arr.length;j++)

{

If((arr[i]>arr[j])

{

Temp = arr[i];

Arr[i] = arr[j];

Arr[j] = temp;

}

}

}

Return arr;

}

Public static int findMaxElement(int arr1[],int arr2[])

{

Return arr1[];

}

**TESTCASE 1**

Input:

[2,5,1,3,9,8,4,6,5,2,3,11],[11,13,2,4,15,17,67,44,2,100,23]

Expected return value:

100

Input:

[100,22,43,912,56,89,85],[234,123,450,234,890,101]

Expected return value:

912

**3.QUESTION**

The function **findMaxElement(int \*arr1,int len1,int \*arr2,int len2)** accepts two integer arrays arr1,arr2 of length len1,len2 respectively.

It is supposed to return the largest element in both the input arrays.

Another function **sortArray(int \*arr,int len)**sorts the input array arr of length len in ascending order and returns the sorted array.

Your task is to use **sortArray(int \*arr,int len)** function and complete the code in **findMaxElement(int \*arr1,int len1,int \*arr2,int len2)** so that it passes all test cases.

**PROGRAM**

int \*sortArray(int \*arr,int \*len)

{

int i=0,j=0,temp=0,index=0;

for(i=0;i<len;i++)

{

for(j=i+1;j<len;j++)

{

If(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

return arr;

}

**4.QUESTION**

The methods G**etDigitSum(int arr[])**of class **DigitSum**accepts an integers array arr it is supposed to calculate the sum of digits of the even of the smallest elements in the input array it returns 1 if the calculated sum is even and returns 0 otherwise

However  there is a compliation error  in the code your task is to fix it so that the program works for all the input values

**PROGRAM**

Public class Digitsum

{

Public static int getDigitSum(int arr[i])

{

Int result,len=arr.length;

For(int i=0;min=arr[0];i<len;i++)

{

if(arr[i]<min)

min=arr[i];

}

results=getSum(min)

if(results%2==0)

return 1;

else

min==arr[j];

}

}

}

**5.PROGRAM**

Public class Datecomparer

{

  Public static int different\_in\_dates(Date date1,Date date2)

{

   Return 0;

}

}

**5.1 TESTCASE 1:**

Input:

2/5/2013,2/6/2013

Expected Return value

31

Input:

1/6/2012,1/6/2011

Expected Return value

366

**6.QUESTION**

The Function **arrayReverse(int \*arr,int len)** accepts an array arr of length len(len >=0) as an argument.The function is expected to reverse the elements of the input array in-place.

For example, if the input array arr is {20,30,10,40,50} the function is expected to return{50,40,10,30,20}

The function compiles successfully but fails to return the desired result due to logical errors

**PROGRAM:**

int arrayReverse(int \*arr,int len){

int i,temp,originallen=len;

for(i=0;i<=i<=originallen/2;i++){

temp=arr[len-1];

arr[len-1]=arr[i];

arr[i]=temp;

len-=1;

}

Return arr;

}

**7.QUESTION**

This method countOccurance (int arr[], int value) of class occurrence is supposed to return the count of occurrences of a number value in the input array arr. The function compiles successfully but fails to return the desired result due to logical errors.

Your task is to debug the program to pass all test cases.

**PROGRAM**

public class Occurrence{

public static int Occurrence (int arr[], int value)

{

int  i=0, count=0, len=arr.length;

while(i<len){

if(arr[i]==value)

count++;

}

return count;

}

}

**8.QUESTION**

The method countElement(int arr[],int n) of class ElementCount is supposed to return the number of elements in the input array arr which are greater  than twice the input number n

**PROGRAM**

Public class ElementCount

{

Public static int countElement(int arr[], int n)

{

Int I,cou-nt=0, len=arr.length;

For(i=0,i<len,i++)

{

If(arr[i]>2n)

 Cou-nt—1;

}

 Return cou.nt;

}

}

**9.QUESTION**

The method reversearray(int arr[]) of class sortarray accepts an input array arr as an arguments.This function is expected to reverse the elements of the array in-place

**PROGRAM**

int arrayReverse(int \*arr,int len){

int i,temp,originallen=len;

for(i=0;i<=i<=originallen/2;i++){

temp=arr[len-1];

arr[len-1]=arr[i];

arr[i]=temp;

len-=1;

}

Return arr;

}

**10.QUESTION**

Public class charecterpattern{

 Public static void printcharecterpattern ( int num){

 Int i , j,value = 1;

Char ch = ‘a’;

Char print = ch;

For(j=0;j<=i;j++)

System.out.print( (ch ++ ));

System.out.println(  \*\*  );

}

}

}

**10.(1) TESTCASE 1:**

**Input:**

5

Expected output:

a

ab

abc

abcd

abcde

**testcase 2:**

input:

1

Output:

a

**11.PROGRAM**

Int isRighttriangle(point \*p1 , point \*p2,point \*p3)

{

}

**11.(1) TESTCASE 1:**

**Input:**

(2,4)(4,2)(1,2)

Expected output

0

**testcase 2:**

input:

(-4,0)(2,0)(-3,-4)

Output:

1

**12.QUESTION**

The function Manchester(int \*arr,int len)accepts an array arr of legth len (len>0) as an input.Each element of an represents a bit -0 0\=or 1.The output is an array with the following property.

**PROGRAM**

Int  \* Manchester(int\*arr, int len){

Int\* res = (int\*)malloc(sizeof(int)\*len);

For(int i= 0; i< len; i++){

Res[i]= (arr[i]==arr[i-1]);

}

 Return res;

}

**13.QUESTION**

Write a method to determine if a given string contains balanced sequence of parenthesis.The input to the method balanceparentheses of class parentheses is a string str.Each character in the string will be “(“ or “)”.The output is the court of balanced pairs if the sequence is  balanced or -1 otherwise

**PROGRAM**

Public class parentheses

{

 Public static int balancedparentheses(string str)

{

}

}

**14 PROGRAM**

int sameelementcount(int \* arr,  int len)

{

  Int i,count = 0;

 For(i=0;i<len-1;i++)

{

if((arr[i]%2==0)&&(arr[i]==arr[i++]))

count++;

}

return  count;

}

}

**14.(1)TESTCASE 1:**

Input:

[1,5,5,2,2,7,8,6,6,9,10],11

Expected return value:

2

**TESTCASE 2:**

Input:

[13,12,12,13,14],5

Expected return value:

1

**15.PROGRAM**

Public class DateComparer

{

Public static int difference\_in\_dates(Date date1, Date date2)

{

//write your code

return 0;

}

}

**15(1). TestCase 1:**

Input:

3, [7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0], 16

Expected Return Value:

11

**TestCase 2:**

Input:

2, [2,3,1,3,2,1,4,3,2], 9

Expected Return Value:

8

**16.QUESTION**

You are given a pre-defined structure primebank and also a collection of related functions which can be used to perform some basic operations on the structure

**PROGRAM**

Void printprime(int num, int digit)

{

}

**17.PROGRAM**

The function **patternPrint(int n)**supposed to print n numberof lines in the following  pattern

For n=4 the pattern should be:

1

1 1

1 1 1

 1 1 1 1

The function complies successfully but fails to return the desired results due to logical  errors

Your task is to debug the program to pass all the testcases

**Program**

Void patternPrint(int n)

{

Int print=1,i,j;

For(int i=0;i<n;i++)

{

For(j=0;j<=i;j++)

{

Printf(“%d”,print);}

}

Print(“\n”);

}

}

**17(1) TESTCASE 1**

**INPUT 1**

3

**EXPECTED OUTPUT VALUES**

1

1 1

1 1 1

**INPUT 2**

1

**EXPECTED OUTPUT VALUES**

1

**18 PROGRAM**

int multiplyNumber(int a, int b, int c)

{

int result, min, max, mid;

max=(a>b)?a>c)?a:c): (b>c)?b:c;

min(a<b?((a<c)?a:c): ((b<c)?b:c);

mid=(a+b+c)-(min+max)

result=(max\*int mid)

return result;

}

**QUESTION**

The function multiplynumber(int a , int b,int c)accepts three integers a,b and c as inputs,it is supposed to return the multiplication productive  of the maximum two or three input numbers

**19.QUESTION**

The function Manchester(int \*arr,int len)accepts an array arr of legth len (len>0) as an input.Each element of an represents a bit -0 0\=or 1.The output is an array with the following property.

**PROGRAM**

Int  \* Manchester(int\*arr, int len){

Int\* res = (int\*)malloc(sizeof(int)\*len);

For(int i= 0; i< len; i++){

Res[i]= (arr[i]==arr[i-1]);

}

 Return res;

}

**20.QUESTION**

The function sameelementcount(int \*arr,int len)accepts an integer array arr of length len as a input and returns the number of elements in an arr which are even numbers and equal to the element to its right

Int sameelementcount(int \*arr, int len)

{

int  i,count=0;

 for(i=0;i<len-1;i++)

{

if((arr[i]%2==0)&&(arr[i]==arr[i++]))

count++;

}

return  count;

}

**21 PROGRAM**

Int countoccurence(int \*arr, int len,int value){

 Int i=0, count  = 0;

While(i<len){

 If(arr[i]++value)

 Count += I;

}

 Return count;

}

**22.PROGRAM**

public class Median

{

public static int median(int arr[])

{

int start\_index=0;

int len=arr.length;

int end\_index=len-1;

int res=-1;

if(len%2!=0)

{

int median\_order=((len+1)/2);

medianCalculate ob=new medianCalculate(arr,start\_index,end\_index,median\_order);

res=ob.quick\_select();

}

else

{

}

return res;

}

}

**22(1). TestCase 1:**

Input:

[4,2,8,6], 4

Expected Return Value:

[6,8,2,4]

**TestCase 2:**

Input:

[11,20,17], 3

Expected Return Value:

[17,20,11]

**23.QUESTION**

The method sortarray(int arr[]) of class array accepts an integer array arr an input and performs an inplace sort algorithm sort operation on it.the function is expected to return the input array sorted in descending order,but instead it returns the value due to bug in the code

**PROGRAM**

Public  class MinArray

{

Public  static int[] sortArray(int[] arr)

{

int i=0,j=0,temp=0,index=0;

for( i=0;i<arr.length;i++)

{

for(j=i+1;j<arr.length;j++)

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

return arr;

}

**24.QUESTION**

The function replaceValues(int \*arr, int len) accepts an array arr of length len (len>0) as an input and returns an array of the same length.

If the length of arr is off, all the elements of arr are supposed to be replaced by 1st and in case it is even, the elements should be replaced by 0s.

**PROGRAM:**

int \*replaceValues(int \*arr, int len)

{

int i,j;

if(len%2==0)

{for(i=0;i<len;i+=2)

arr[i]=0;

}

else

{

for(j=0;j<len;j+=2)

arr[j]=1;

}

return arr;

}

**25.PROGRAM**

float median(int \* arr,int len)

{

int start\_index=0;

int end\_index=len-1;

float res=-1;

if(len%2!=0)

{

int median\_order=((len+1)/2);

res = (float)quick\_select(arr,start\_index,end\_index,median\_order);

}

else

{

}

return res;

}

**25(1) TESTCASE1:**

Input:

[2,40,23,52,37],5

Output

37.000000

Input:

[2,12,9,5],4

Output:

7.000000

**26.PROGRAM**

Public class arraysum{

  Public static int getarraysum(int arr[1]){

 Int sum = 0;I<len j i\* 1}{

 Sum = arr[i];

}

 Return sum;

}}

**26(1)TESTCASE**

Input

[2,3,5,7,9]

Output:

26

Input:

[-1,-2,-3,-4,-4,-1]

Output:

-15

**27.QUESTION**

The function allexponent(int base,int exponent)accepts two integers base and exponent as [inputs.it](http://inputs.it/) is supposed to calculate and return of exponentation of base raised to power exponent for all input values

**PROGRAM**

 float allExponent( int **base**,int**exponent** )

{

float res=1;

if(exponent >=0)

{

Res = (float)positiveExponent(base,exponent)

}

Else

{

res=0;

}

return res;

}

**28.PROGRAM**

int countElement(int arr,int len,int n)

{

int i,count=0;

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

{

if(arr[i]>(2\*n))

{

count=+1;

}

return count;

}

**28. TESTCASE 1:**  
Input:  
[-2, -4, -3, -5, -6, -7, -8], 7, 3  
Expected Return Value:  
0  
**TESTCASE 2:**  
Input:  
[22, 55, 66, 33, 44, 77], 6,13  
Expected Return Value:

5

**29.PROGRAM**

Public class color

{

Public static  void printcolor(int num)

{

Switch (num)

{

Case 1:

System.out.println(“Red”);

Case 2:

System.out.println(“Black”);

Case 3:

System.out.println(“White”);

Case 4:

System.out.println(“Green”);

default:

System.out.println(“No color”);

Break;

}

}

}

**29(1)TESTCASE 1:**

**Status:**

Wrong

**Expected:**

White

**Returned:**

White

Green

No color

**TESTCASE 2:**

**Status:**

Wrong

**Expected:**

White

**Returned:**

White

Green

No color

**30 QUESTION**

The function matrixsum(int \*\*matrix,int m,int n)is supposed to return the sum of elements  of the input array matrix having m rows and n columns

**PROGRAM**

int calculateMatrixSum(int \*\*matrix, int m, int n)

{

int i,j,sum=0,

for(j=0i<m;j++)

{

For(j=0;j<n;j++)

sum+= matrix[i][j];

}

return sum;

}

**31.QUESTION**

In order to help her,we have a function checkBirthday(char \* month,int day) which takes day and month as inputs and returns 1 if is her birthday and returns a.0 otherwise

**PROGRAM**

Int checkBirthday(char\* month,int day)

{

If(strcmp(month,”july”)||(day-5))

Return 1;

Else

Return 0;

}

**32.PROGRAM**

Public  class MinArray

{

Public  static int[] sortArray(int[] arr)

{

int i=0,j=0,temp=0,index=0;

for( i=0;i<arr.length;i++)

{

for(j=i+1;j<arr.length;j++)

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

return arr;

}

**32(1) TESTCASE 1:**

Input:

[2,5,1,3,9,8,4,6,5,2,3,11],[11,13,2,4,15,17,67,44,2,100,23]

Output:

1

**TESTCASE 2:**

Input:

[100,22,43,912,56,89,85],[235,123,450,234,890,101]

Output:

22

**33.PROGRAM**

Public class Datecomparer

{

  Public static int different\_in\_dates(Date date1,Date date2)

{

   Return 0;

}

}

**33(1) TESTCASE 1:**

Input:

2/5/2013,2/6/2013

Expected Return value

31

Input:

1/6/2012,1/6/2011

Expected Return value

366

**34 QUESTION**

The method **findMinElement(int arr1[i],int arr2[])** of  class**MinArray** accepts two integer array

arr1,arr2

it is supposed to return  the smallest elements in both the input arrays

Another  method **sortArray(int arr[])**  sorts the input  array arr in ascending order  and returns

**PROGRAM**

Public  class MinArray

{

Public  static int[] sortArray(int[] arr)

{

int i=0,j=0,temp=0,index=0;

for( i=0;i<arr.length;i++)

{

for(j=i+1;j<arr.length;j++)

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

return arr;

}

public static int findMinElement(int arr[i],int arr2[j])

{

return 1;

}

}

**35.QUESTION**

The function calculateMatrixSum(int \*\* matrix, int  m, int n) accepts a two dimensional array matrix of dimensions m, n as input and returns the sum of odd elements whose ith and jth index are same.

The function compiles line but falls to return the desired result for some test cases

**PROGRAM**

int calculateMatrixSum(int \*\*matrix, int m, int n)

{

int i,j,sum=0,row=m, column=n;

if((row>0)&&(column>0))

{

for(i=0;i<row;i++)

{

sum=0;

for(j=0;j<column;j++)

{

if(i==j)

{

if(matrix[i][j]/2!=0)

sum+=matrix[i][j];

}

}

}

return sum;

}

else

return sum;

}

**36.PROGRAM**

Public class Arraysum{

Public static int getarraysum(int arr[]){

 Int sum =0 ;i,len.arr.length;

For(i=0;i<len;i=i+1){

Sum =arr[i];

}

Return sum;

}

}

**36(1)TESTCASE**

Input:

[2,3,5,7,9]

Expected Return value

26

Input:

[-1,-2,-3,-4,-1]

Expected Return value

-15

**37.QUESTION**

The function calculate matrixsum(int “matrix ,int m ,int n)accepts a two dimensional array matrix of dimensions m, n as input and returns the sum of odd elements whose I and l index are same

**PROGRAM**

int calculateMatrixSum(int \*\*matrix, int m, int n)

{

int i,j,sum=0,row=m, column=n;

if((row>0)&&(column>0))

{

for(i=0;i<row;i++)

{

sum=0;

for(j=0;j<column;j++)

{

if(i==j)

{

if(matrix[i][j]/2!=0)

sum+=matrix[i][j];

}

}

}

return sum;

}

else

return sum;

}

**38.QUESTION**

The method countdigit(int sum) of class digits is supposed to return the value remainder when the input arguments num(num>0) is divided by the number of digits in num

**PROGRAM**

Public class digits{

  Public static int countDigits ( int num ){

   Int count =0;

While(num !=0)

Num = num / 10;

Count ++;

}

Return (num % count );

}

}

**39.QUESTION**

The function calculateMatrixSum(int \*\* matrix, int  m, int n) accepts a two dimensional array matrix of dimensions m, n as input and returns the sum of odd elements whose ith and jth index are same.

The function compiles line but falls to return the desired result for some test cases.

**PROGRAM**

int calculateMatrixSum(int \*\*matrix, int m, int n)

{

int i,j,sum=0,row=m, column=n;

if((row>0)&&(column>0))

{

for(i=0;i<row;i++)

{

sum=0;

for(j=0;j<column;j++)

{

if(i==j)

{

if(matrix[i][j]/2!=0)

sum+=matrix[i][j];

}

}

}

return sum;

}

**40.PROGRAM**

Public class arrayoperation{

  Public static int[] replacevalues( int arr[] ) {

Int I ,j , len = arr.length;

If (len % 2==0)

{

For(i=0;i==len;i++)

Arr[i] = 0;

}

Else{

 For( j=0;j<=len ;j++)

 Arr[j] = I;

}

 Return arr;

}

}

**40(1) TESTCASE 1:**

Input:

[2,5,3,7,9,5]

Expected Return value

[0,0,0,0,0,0]

Input:

[21,22,3,24,55]

Expected Return value

[1,1,1,1,1]

**41.QUESTION**

The function findminElement(int \* arr1,int len1,int  \*arr2,int len2)accepts two integers  arrays arr1,arr2 of length len1,len2 respectively

**PROGRAM**

int \*sortArray(int \*arr,int \*len)

{

int i=0,j=0,temp=0,index=0;

for(i=0;i<len;i++)

{

for(j=i+1;j<len;j++)

{

If(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

return arr;

}

**42.QUESTION**

The methods G**etDigitSum(int arr[])**of class **DigitSum**accepts an integers array arr it is supposed to calculate the sum of digits of the even of the smallest elements in the input array it returns 1 if the calculated sum is even and returns 0 otherwise

However  there is a compliation error  in the code your task is to fix it so that the program works for all the input values

Note The methods getdigitSum uses another method getSum(int sum) which returns  the sum of the digits of the input number num

**PROGRAM**

Public class Digitsum

{

Public static int getDigitSum(int arr[i])

{

Int result,len=arr.length;

For(int i=0;min=arr[0];i<len;i++)

{

if(arr[i]<min)

min=arr[i];

}

results=getSum(min)

if(results%2==0)

return 1;

else

min==arr[j];

}

}

}

**43.PROGRAM:**

public class AllExponent

{

public static float allExponent( int **base**,int**exponent** )

{

float res=1;

if(exponent >=0)

{

Exponent exp=new  Exponent(base,exponent);

res = (float)exp.positiveExponent();

}

else

res=0;

}

return res;

}

}

**TESTCASE 1:**

**Input:**

5,2

**Expected Return Value:**

25.0

**TestCase 2:**

**Input**

5,-2

**Expected Return Value:**

0.04

**44. Question :**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp; Run anytime to check the compilation/execution status of the

program. You can use System.out.println to debug your code. The submitted

code should be logically/syntactically correct and pass all testcases. Do not

write the main() function as it is not required.

**Code Approach:** For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or incorporate

any additional library methods.

The method printColor(int num) of the class Color is supposed to print

names of color according to given input numbers num

When the values of num equal 1,2,3,4 the function prints

“Red”,”Black”,”White”,”Green” respectively for any other values of num it

should print “No color”.

The method complies fine but fails to return the desired results for some cases

Your task is to fix code so that it passes all the testcases

**PROGRAM:**

Public class color

{

Public static  void printcolor(int num)

{

Switch (num)

{

Case 1:

System.out.println(“Red”);

Case 2:

System.out.println(“Black”);

Case 3:

System.out.println(“White”);

Case 4:

System.out.println(“Green”);

default:

System.out.println(“No color”);

Break;

}

}

}

**45. QUESTION:**

The function **calculateMatrixSum(int matrix[][])** of class**MatrixSum** accepts a two dimensional array matrix of dimensions as input and returns the sum of odd elements whose ith and jth index are same.

The function compiles line but falls to return the desired result for some test cases.

**PROGRAM:**

Public class **MatrixSum**

{

Public static int **calculateMatrixSum**(int **matrix[][]**)

{

int i,j,sum=0,row=m, column=n;

if((row>0)&&(column>0))

{

for(i=0;i<row;i++)

{

sum=0;

for(j=0;j<column;j++)

{

if(i==j)

{

if(matrix[i][j]/2!=0)

sum+=matrix[i][j];

}

}

}

return sum;

}

else

return sum;

}

**46. PROGRAM:**

Public class **Drawpattern**{

 Public static void **printPattern**(int n){

Int i,,j,print = 1;

For(i=i;i<=n;i++)

For(j=1;j<=2 \* i;j++)

System.ou.print( (print ));

System.out.println();

}

}

**TESTCASE 1:**

**Input:**

4

**Expected Return Value:**

11

1111

111111

11111111

**TESTCASE 2:**

**Input:**

1

**Expected Return Value:**

11

**47. QUESTION:**

You are required to fix all logical errors in the given code. You can click on

Compile &amp; Run anytime to check the compilation/execution status of the

program. You can use System.out.println to debug your code. The submitted

code should be logically/syntactically correct and pass all test cases. Do not

write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation  We do not expect you to modify the approach or incorporate

any additional library methods.

The function **sortArray(int \* arr,int len)** accepts an integer array  arr of  length

(len>0) as an input and perform an in place sort operation on it. The function is expected to return the input array sorted in descending order

The function complies successfully but fails to return the desired results due to logical  errors

Your task is to debug the program to pass all the test cases

**PROGRAM:**

int \*sortArray(int \*arr,int \*len)

{

int i=0,j=0,temp=0,index=0;

for(i=0;i<len;i++)

{

for(j=i+1;j<len;j++)

{

If(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

return arr;

}

}

}

**48. TestCase 1:**

**Input:**

[1,2,3,4,5,6,7]

**Expected Return Value:**

[7,6,5,4,3,2,1]

**TestCase 2:**

**Input:**

[2,8,4,6]

**49. TestCases:**

“status”, “Error”, “contains”, in functions “point\_str”.

\r\n.27,24: warning, incompatible implicit declaration of built in function ‘matrix’

**50. QUESTION:**

You are required to fix all logical errors in the given code. You can click on

Compile &amp; Run anytime to check the compilation/execution status of the

program. You can use System.out.println to debug your code. The submitted

code should be logically/syntactically correct and pass all test cases. Do not

write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation  We do not expect you to modify the approach or incorporate

any additional library methods.

The function **maxReplace(int arr[])** of class **MaxArray** is supposed to replace every element of the input array arr with the maximum element of arr.

The function complies successfully but fails to return the desired results due to logical  errors

Your task is to debug the program to pass all the test cases

**PROGRAM:**

public class **MaxArray**

{

Public static int[] **maxReplace**(int arr)

{

int I,len=arr.length;;

if(len>0)

{

int max=arr[0];

for(i=0;i<len;i++)

{

if(max<arr[i])

{

max=arr[i];

}

}

}

**51. TestCase 1:**

**Input:**

[2,5,8,11,3]

**Expected Return Value:**

[11,11,11,11,11]

**TestCase 2:**

**Input:**

[3,2,5,8,9,11,23,45,63]

**Expected Return Value:**

[63,63,63,63,63,63,63,63,63]

**52. PROGRAM:**

int multiplyNumber(int a, int b, int c)

{

int result, min, max, mid;

max=(a>b)?a>c)?a:c): (b>c)?b:c;

min(a<b?((a<c)?a:c): ((b<c)?b:c);

mid=(a+b+c)-(min+max)

result=(max\*int mid)

return result;

}

**53. QUESTION:**

The method manchester (int arr) of class signal accepts an array arr as an input. Each element of arr represents a bit 0 or 1. The output is an array with the following property for each element in the input array arr. If the bit arr[i] is the same as arr[i-1], then the element of the output array is 0,iIf they are different then its 1. For the first bit in the input array , assume its previous bit to be 0. This encoding is stored and returned in a new array.

For e.g if arr is {0, 1, 0, 0, 1, 1, 1, 0}  the function should return  an array {0, 1, 1, 0, 1, 0, 0, 1}.

The function compiles successfully but fails to return the desired result due to logical errors.

Your task is to debug the program to pass all test cases.

**PROGRAM:**

public class Signal

{

public static int[] manchester(int arr[])

{

int len=arr.length;

int res=new int[len];

boolean result;

res[0]=arr[0];

for(int i=1;i<len;i++)

{

result=(arr[i]==arr[i-1]);

res[i]=(result)?1:0;

}

return res;

}

}

**54. TestCase 1:**

**Input:**  
2/5/2013, 2/6/2013

**Expected Return Value:**31

**TestCase 2:**

**Input:**1/6/2011, 1/6/2012

**Expected Return Value:**  
366

**55. QUESTION:**

You are a software developer at ABC technologies. You are supposed to enhance the functionality of a particular module which control a functionality based on date of logging.

Given two date, your task is to write a method which calculates the difference between them. The difference (which shall always be zero o0r a positive number) would then be used by another module to perform checks on logging.

Developer at ABC technologies already use a pre-defined class date containing day, month and year as members and also a collection of method for performing some common operations. You are supposed to make use of these methods to calculate and return the difference.

(Please refer to the Helper Code tab for details regarding the class)

The method difference in dates has to be completed which takes two date instances as inputs and returns the difference.

**PROGRAM:**

int difference\_in\_dates(Date \*date1, Date \*date2)

{

//write your code here

}

**56. TestCase 1:**

**Input:**  
54

**Expected Return Value:**D

**TestCase 2:**

**Input:**  
78

**Expected Return Value:**B

**57. QUESTION:**

This method countOccurance (int arr[], int value) of class occurrence is supposed to return the count of occurrences of a number value in the input array arr. The function compiles successfully but fails to return the desired result due to logical errors.

Your task is to debug the program to pass all test cases.

**PROGRAM:**

public class Occurrence{

public static int Occurrence (int arr[], int value)

{

int  i=0, count=0, len=arr.length;

while(i<len){

if(arr[i]==value)

count++;

}

return count;

}

}

**58. TestCase 1:**

**Input:**  
(0)

**Expected Return Value:**2

**TestCase 2:**

**Input:**  
0{

**Expected Return Value:**-1

**59. TestCase 1:**

**Input:**  
3, [7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0]

**Expected Return Value:**

11

**TestCase 2:**

**Input:**  
2, [2,3,1,3,2,1,4,3,2]

**Expected Return Value:**

8

**60. PROGRAM:**

public class **LeastDecentPage**

{

//Method statement here

public static int iruCount(int num\_cache\_wise,int[] pages)

{

//Insert your code here

}

}

**61. QUESTION:**

The function **removeElement( int \*arr,int len,int index)**takes an array arr of length lenas an input. It is supposed to return an array len-1 after removing the integer at the given index in the input arrayarr.If the given index is out of bounds, then this function should return the input array arr

The function compiles successfully but fails to return the desired result due to logical errors

**PROGRAM:**

int\* removeElement( int \*arr,int len,int index){

int i,j;

if(index<len){

for(i=index;i<len-1;i++){

arr[i]=arr[i++];

}

Int \*rarr =(int\*)malloc(sizeof(int)\*(len-1));

For(i=0;i<len-1;i++)

Rarr[i]=arr[i];

Return rarr;

}

else

return arr;

}

**62. TestCase 1:**

**Input:**

[3,6,4,1,7,9,1,3,12,15]

**Expected Return Value:**

[1,1,3,3,4,6,7,9,12,15]

**TestCase 2:**

**Input:**

[3,3,3,3,3,3,3,3,3,3]

**Expected Return Value:**

[3,3,3,3,3,3,3,3,3,3]

**63. QUESTION:**

The method  sortArray(int arr[]) of class Selection sort an integer array arr as input and perform an in place selection sort .The function an input array sorted as ascending order

The function compiles fine but to return desired results for some cases

Your task to fix  the code so but that it passes at test cases

Assumption:

In this particular implementation sort the smallest elements in the array is swapped with the elements of the next index and so on

**PROGRAM:**

Public class SelectionSort

{

Public static int[] sortArray(int arr[])

{

int x=0,y=0,n=arr.length;

for(x=0;x<n;x++)

{

int index\_of\_min = x;

for(y=x;y<n;y++)

{

If(arr[index\_of\_min]>arr[x])

{

Index\_of\_min=y;

}

}

Int temp=arr[x];

Arr[x]=arr[index\_of\_min];

arr[index\_of\_min]=temp;

}

return  arr;

}

}

**64. QUESTION;**

The function getarraysum(int \* arr,int len)is  supported to calculation and return the sum of elements of the input array arr of length len(len>0)

The function compiles successfully but fails to return the desired result due to logical errors.

**PROGRAM:**

Int getarraysum(int \*arr,int len){

Int sum = 0;

For(i=0;i<len;i-i-1)  
{

Sum = arr[i];

}

Return sum;

}

**65. QUESTION:**

The method median(int arr[]) of class Median accepts an integer array arr. It is supposed to calculate and return the median of elements in the input array.

However, incomplete code in the method median (int arr[]) works only for odd length arrays.

**PROGRAM:**

public class Median

{

public static int median(int arr[])

{

int start\_index=0;

int len=arr.length;

int end\_index=len-1;

int res=-1;

if(len%2!=0)

{

int median\_order=((len+1)/2);

medianCalculate ob=new medianCalculate(arr,start\_index,end\_index,median\_order);

res=ob.quick\_select();

}

else

{

}

return res;

}

}

**66. PROGRAM:**

public class CharacterPattern{

public static void printCharacterPattern(int num)

{

int  i, j, value=1;

char ch= ‘a’;

char print;

for(i=0;i<num;i++)

{

print=ch;

for(j=0;j<=i;j++)

System.out.print((ch++));

System.out.println();

}

}

}

**67. PROGRAM:**

int\* removeElement( int \*arr,int len,int index){

int i,j;

if(index<len){

for(i=index;i<len-1;i++){

arr[i]=arr[i++];

}

Int \*rarr =(int\*)malloc(sizeof(int)\*(len-1));

For(i=0;i<len-1;i++)

Rarr[i]=arr[i];

Return rarr;

}

else

return arr;

}

**68. TestCase 1:**

**Input:**

[1,2,3,4,5,6,7,8,9], 9,3

**Expected Return Value:**

[1,2,3,5,6,7,8,9]

**TestCase 2:**

**Input:**

[11,23,12,34,54,32], 6,6

**Expected Return Value:**

[11,23,12,34,54,32]

**69. PROGRAM:**

 int countElement(int arr,int len,int n)

{

int i,count=0;

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

{

if(arr[i]>(2\*n))

{

Count+=1;

}

return count;

}

**70. TestCase 1:**

**Input:**

[-2,-4,-3,-5,-6,-7,-8], 3

**Expected Return Value:**

0

**TestCase 2:**

**Input:**

[22,55,66,33,44,77], 13

**Expected Return Value:**

5

**71. TestCase 1:**

**Input:**

[2,40,23,52,37], 5

**Expected Return Value:**

37.000000

**TestCase 2:**

**Input:**

[2,12,5,9], 4

**Expected Return Value:**

7.000000

**72. TestCase 1:**

**Input:**

[1,1,0,0,1,0], 6

**Expected Return Value:**

[1,0,1,0,1,1]

**TestCase 2:**

**Input:**

[0,0,0,1,0,1,1,1], 8

**Expected Return Value:**

[0,0,0,1,1,1,0,0]

**73. QUESTION:**

You are required to complete the given code by  reusing existing functions .you can click on complie and run any time to check the compilation / execution status of the program you can use System.out.println to debug your code. The submitted code should be logically/syntactically correct and pass all testcases .do not write the main() function has it is not required

**Code Approach:** For this question you will need to complete the code as in given implementation.we don’t  expect you to modify the approach The method**findMinElement(int arr1[i],int arr2[])** of  class**MinArray** accepts two integer array arr1,arr2 it is supposed to return  the smallest elements in both the input arrays Another  method **sortArray(int arr[])**  sorts the input  array arr in ascending order  and returns It Your task is to use **sortArray(int arr[])** method and complete the code in**FindMinElement(int arr1[],int arr2[]).**so that it possible all test cases.

**PROGRAM:**

Public  class MinArray

{

Public  static int[] sortArray(int[] arr)

{

int i=0,j=0,temp=0,index=0;

for( i=0;i<arr.length;i++)

{

for(j=i+1;j<arr.length;j++)

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

return arr;

}

public static int findMinElement(int arr[i],int arr2[j])

{

return 1;

}

}

**74. TestCase 1:**

**Input:**

3, [7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0], 16

**Expected Return Value:**

11

**TestCase 2:**

**Input:**

2, [2,3,1,3,2,1,4,3,2], 9

**Expected Return Value:**

8

**75. PROGRAM:**

int\* **cellCompete**(int\* **cells,**int **days**)

{

//Write tour code here

}

//Function signature ends

**76. PROGRAM:**

char checkGrade(int score)

{

if(score<=60)

return  ‘D’;

else if((61<=score)&&(score<=75))

return ‘C’;

else if((76<=score)&&(score<=90))

return ‘B’;

else

return ‘A’;

}

**77. QUESTION :**

You are required to fix all logical errors in the given code. You can click on

Compile &amp; Run anytime to check the compilation/execution status of the

program. You can use System.out.println to debug your code. The submitted

code should be logically/syntactically correct and pass all testcases. Do not

write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or incorporate

any additional library methods.

The method **deleteDuplicate(int arr[])** of class DistinctArray takes an array as an input it is supposed to remove duplicates integers from the input array arr such that for each distinct integer the first occurrence is retained and all the duplicates elements following it are removed for Example given input array (2,3,2,2,5,6,6,7) the expected output is (2,3,5,6,7)

The function complies successfully but fails to return the desired results due to logical errors

Your task is debug the program to pass all the test cases

**PROGRAM:**

Public class DistinctArray{

Public static int[] deleteDuplicate (int arr[])

{

int count=0,p,len=arr.length ,i,j,k,originalLength=len;

for(i=0;i<len;i++)

{

for(j=i+1;j<len;j++)

{

if(arr[j]==arr[i])

{

arr[k]=arr[k+1];

}

len=len-1;

count=count+1;

j=i;

}

}

}

return (int\*)arr;

}

**78. QUESTION:**

You are given a pre-defined structure **primebank** and also a collection of related functions which can be used to perform some basic operations on the structure

**PROGRAM:**

Import **java.lang.Math;**

public class **Prime**

{

Public static void **printPrime**(int **num1,**int **n**)

{

//Write your code here

}

}

**79. PROGRAM:**

Int\* cellcompete(int\* cells, int days)

{

}

**TESTCASE 1:**

**Input:**

[1,0,0,0,0,1,0,1], 1

**Expected return value:**

[0,1,0,0,1,0,1,0]

**TESTCASE 2**

**Input:**

[1,1,1,0,1,1,1,1],2

**Expected return value:**

[0,0,0,0,0,1,1,0]

**80. TESTCASE 1:**

**Input:**

[[5,6],[11,14],[23,27]]

**Expected return value:**

5

**TESTCASE 2**

**Input:**

[[3,1,7],[2,5,4],[7,8,9]]

**Expected return value:**

17

**81. QUESTION:**

You are required to fix all logical errors in the given code. You can click on

Compile &amp; Run anytime to check the compilation/execution status of the

program. You can use System.out.println to debug your code. The submitted

code should be logically/syntactically correct and pass all testcases. Do not

write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or incorporate

any additional library methods.

Lisa  always forgets her  birthday which is on **5 Th july**

In order to help her we have function**CheckBirthDay(char \*month,int day)**which takes day and month as inputs and returns  1 if its her birthday and returns a 0 otherwise

The function compiles fine but to return desired results for some cases

Your task to fix  the code so but that it passes at test cases

**PROGRAM:**

int **CheckBirthDay(char \*month,int day)**

**{**

if(strcmp(month,”july”)||(day=5))

{

return 1;

}

else

{

return 0;

}

**82. PROGRAM:**

Public class Digitsum

{

Public static int getDigitSum(int arr[i])

{

Int result,len=arr.length;

For(int i=0;min=arr[0];i&lt;len;i++)

{

if(arr[i]&lt;min)

min=arr[i];

}

results=getSum(min)

if(results%2==0)

return 1;

else

min==arr[j];

 }

}

}

**83. TestCase 1:**

**Input:**

(3,4),(2,1),(1,5)

**Expected  Return Values:**

1

**TestCase 2:**

**Input:**

(1,-1),(0,-1),(1,-1)

**Expected  Return Values:**

0

**84. TestCase 1:**

**Input:**

[2,2,3,3,4,4]

**Expected  return Value:**

[2,3,4]

**TestCase 2:**

**Input :**

[21,21,21,21,21]

**Expected return Value:**

[21]

**85. TestCase 1:**

**Input:**

[1,2,3,4,5,6,7]

**Expected Return values:**

[7,6,5,4,3,2,1]

TestCase 2:

**Input :**

[2,8,4,6]

**Expected Return values:**

[6,4,8,2]

**86.Question**

**TESTCASE**

TestCase 1:

Input :

[1,2,3,4,5,6,7]

Expected  Return Values :

[7,6,5,4,3,2,1]

TestCase 2:

Input:

[2,8,4,6]

Expected Return Values

[6,4,8,2]

**Program**

public class SortArray

{

public static int[] reverseArray(int arr[])

{

int i, temp, orginallen=arr.length;

int len=orginallen;

for(i=0;i&lt;orginallen/2;i++)

{

temp=arr[len-1];

arr[len-1]=arr[i];

arr[i]=temp;

len+=1;

}

return arr;

}

}

**87.Question**

**Problem**

The function countDigits (int num) is supposed  to reurn the remainder when

the input argument num (num&gt;0) is divided by the number of digits in sum.

The function compiles successfully but fails to return the desired result due to

logical errors.

Your task is to debug the program to pass all test cases.

**PROGRAM:**

Public class Digits{

Public static int countDigits(int num){

int countDigits(int num){

int count=0;

while(num!=0){

num=num/10;

count++;

}

return (num%count);

}

**88.Question**

**Problem**

The Function reverseArray(int arr[])of class  sort Array arr of an arguments

For example, if the input array arr is {20,30,10,40,50} the function is expected

to return{50,40,10,30,20}

The function compiles successfully but fails to return the desired result due to

logical errors

**PROGRAM:**

int arrayReverse(int \*arr,int len){

int i,temp,originallen=len;

for(i=0;i&amp;lt;=i&amp;lt;=originallen/2;i++){

temp=arr[len-1];

arr[len-1]=arr[i];

arr[i]=temp;

len-=1;

}

}

**89.Question**

**TestCase**

Testcase 1

Input:

[-2,-4,-3,-5,-6,-7,-8] -3

Expected return values

0

Testcase 2

Input:

[22,55,66,33,44,77] .13

Expected return values

5

**Program**

Public class Elementcount

{

Public static int countelement(int arr[],int n)

{

Int I,count=0,len=arr.length;

For(i=0;i<len;i++)

{

If(arr[i]=2n)

Count=-1;

}

return count ;

}

}

**90.Question**

**TESTCASES**

TESTCASE 1:

Input:

2/5/2013, 2/6/2013

Expected Return Value:

31

TESTCASE 2:

Input:

1/6/2011, 1/6/2012

Expected Return Value:

366

**91.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click

onCompile &amp;amp;amp; Run anytime to check the compilation/execution statusof the program. You can use System.out.println to debug your code. The

submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or incorporate

any additional library methods.

The method printCharacterPattern(int n) of class characterPattern is expected to print the first n (26>n>0)   lines of the pattern

For examples if n=4 the pattern is

a

ab

abc

abcd

**Program**

public class CharacterPattern{

public static void printCharacterPattern(int num)

{

int  i, j, value=1;

char ch= ‘a’;

char print;

for(i=0;i&lt;num;i++)

{

print=ch;

for(j=0;j&lt;=i;j++)

System.out.print((ch++));

System.out.println();

}

}

}

**92.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click

onCompile &amp;amp;amp; Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The

submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach:** For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or incorporate

any additional library methods.

The Least-Recently-Used(LRU) cache algorithm exists the element from the cache(when it&#39;s full)

that was least-recently-used. After an element is requested from the cache, it should be added to

the cache (if not already there) and considered the most-recently-used element in the cache.

Given the maximum size of the cache and a list of integers(to request from the

cache), calculate the number of cache misses using the LRU cache algorithm. A

cache miss occur when the requested integer does not exist in the cache.

Initially, the cache is empty.

The input to the function lruCountMiss shall consist of an integer

max\_cache\_size, an array pages and its length len.

The function should return an integer for the number of cache misses using

the LRU cache algorithm.

Assume that the array pages always has pages numbered from 1 to 50.

**93.Question**

**TESTCASE**

TestCase  1

Input:

[1,1,0,0,1,1,1],6

Expected Return Values

[1,0,1,0,1,1]

TestCase 2

Input:

[0,0,0,1,0,1,1,1],8

Expected return Values

[0,0,0,1,1,1,0,0]

**Program**

int \* manchester(int\* arr,int len){

int\*res=(int\*)malloc(sizeof(int)\*len);

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

{

res[i]=(arr[i]==arr[i-1]);

}

Return res;

}

**94.Question**

**Problem**

 You are required to complete the given code by reusing existing functions youcanclick on complie and run any time to check the compilation / executionstatus of the program you can use System.out.println to debug your code. Thesubmitted code should be logically/syntactically correct and pass all testcases.do not write the main() function has it is not required

**Code Approach** :For this question you will need to complete the code as in

given implementation.we don’t expect you to modify the approach

You are given a pre-defined structure Point and also a collection of related

functions which can be used to perform some basic operations on the structure.

You will have to implement the function **isRightTriangle(Ppoint \*P1, Point**

**\*P2,Point \*P3)** which accepts 3 points as input and checks whether the given 3

points can make a right angle triangle or not.

If they make a right angle triangle the function returns 1 else if returns 0

You are supposed to use point structure and associated fuction for the task

**95.Question**

**Program**

Public class Multipation

{

  Public ststic int multiplynumber(int a,int b,int c)

{

Int result,min,max,mid;

Max=(a&gt;b)?a&gt;c?a:cLb&gt;c)?b:c);

Min=(a&lt;b)?((a&lt;c)?a:c((b&lt;c)?b:c);

Mid=(a+b+c)-(min+max);

Result=(max\*int mid);

Return result;

}

}

**95.TESTCASES**

TESTCASE 1

Input

5,7,4

Expected return value:

35

TESTCASE 2

Input

11,12,13

Expected return value:

156

**96.Question**

**TESTCASES**

TestCase 1

Input:

4

Expected Return value:

11

1111

111111

11111111

TestCase 2:

Input:

1

Expected Return Value:

11

**96.Program**

Void drawPrintPattern(int n)

{

int i,j,print=1;

for(i=1;i<=n;i++){

for(j=1;j<=2\*I;j++)

printf(“%d”,print);

printf(“\n”);

}

}

**97.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp; Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or

incorporateany additional library methods.

 The function drawprintPattern(int n) is expected to print the first n(n&gt;0) lines of the pattern shown below

For Example ,if n=3 the pattern should be

11

1111

111111

The function complies successfully but fails to return the desired results due to

logical  errors

Your task is to debug the program to pass all the testcases

**Program**

Void drawPrintPattern(int n)

{

int i,j,print=1;

for(i=1;i<=n;i++){

for(j=1;j<=2\*I;j++)

printf(“%d”,print);

printf(“\n”);

}

**98.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp; Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or

incorporateany additional library methods.

The function countOccurrence(int \*arr,int len,int value) is supposed to return the count of occurences  of number value in the input array arr of length (len>0)

The function compiles successfully but fails to return the desired resukts due to logical errors

**Program**

Int countOccurrence(int \*arr,int len,int value){

Int i=0;count=0;

While(i=len){

If(arr[i]==value)

Count=1;

}

return  count;

}

**99.Question**

**TESTCASES**

TestCase 1 :

Input:

[3,6,4,1,7,9,1,3,12,15]

Expected  Return values

[15,12,9,7,6,4,3,3,1,1]

TestCase 2

Input:

[3,3,3,3,3,3,3,3,3]

Expected return Values :

[3,3,3,3,3,3,3,3,3]

**Program**

Public class SelectionSort

{

Public static int[] sortArray(int arr[])

{

int x=0,y=0,n=arr.length;

for(x=0;x&lt;n;x++)

{

int index\_of\_min = x;

for(y=x;y&lt;n;y++)

{

If(arr[index\_of\_min]&gt;arr[x])

{

Index\_of\_min=y;

}

}

Int temp=arr[x];

Arr[x]=arr[index\_of\_min];

arr[index\_of\_min]=temp;

}

return  arr;

}

}

**100.Question**

**Problem**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp; Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or

incorporateany additional library methods.

The method median(int arr[]) of class Median accepts an integer array arr. It is

supposed to calculate and return the median of elements in the input array.

However, incomplete code in the method median (int arr[]) works only for odd

length arrays.

**Program**

Float  median(int \* arr,int len)

{

int start\_index=0;

int end\_index=len-1;

float res = -1;

if(len%2!=0)

{

Int median\_order = ((len+1)/2);

Res=(float)quick\_select(arr,start\_index,end\_index,median-order)

}

Else

{

}

Return res;

}

}

**101.Question**

**PROGRAM**

public calss AllExponent

{

public static float allExponent( int base,int exponent )

{

float res=1;

if(exponent &gt;=0)

{

Exponent exp=new  Exponent(base,exponent);

res = (float)exp.positiveExponent();

}

else

res=0;

}

return res;

}

}

**TESTCASE**

TestCase 1;

Input

5,2

Expected Return Value:

25.000000

TestCase 2:

Input

5,-2

Expected Return Values:

0.040000

**102.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp; Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or

incorporateany additional library methods.

The method  sortArray(int arr[]) of class Selection sort an integer array arr as

input and perform an in place selection sort .The function an input array sorted

as ascending order

The function compiles fine but to return desired results for some cases

Your task to fix  the code so but that it passes at test cases

Assumption:

In this particular implementation sort the smallest elements in the array is

swapped with the elements of the next index and so on

**Program**

Public class SelectionSort

{

Public static int[] sortArray(int arr[])

{

int x=0,y=0,n=arr.length;

for(x=0;x&lt;n;x++)

{

int index\_of\_min = x;

for(y=x;y&lt;n;y++)

{

If(arr[index\_of\_min]&gt;arr[x])

{

Index\_of\_min=y;

}

}

Int temp=arr[x];

Arr[x]=arr[index\_of\_min];

arr[index\_of\_min]=temp;

}

return  arr;

}

}

**103.Question**

**TESTCASES**

TestCase 1

Input:

[[3,2,1],[4,6,5],[7,8,9],3,3

Expected return Values:

45

TestCase2

Input:

[[3,12,10,[14,61,51],[21,84,95],3,3

Expected return Values:

357

**Program**

Int matrixsum( int &amp;matrix, int m, int n )

{

Int i,j,sum=0;

For(i=0;i&lt;m;i++)

{

For(j=0;j&lt;n;j++)

Sum + =matrix(i)(j);

}

Return sum;

}

**104.question**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp; Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We do not expect you to modify the approach or incorporateany additional library methods.

The  function patternPrint(int n) is supposed to print  n number of lines in the following pattern

For n=4 the pattern should be

1

11

111

1111

The function compiles line by to return the desired results for some testcases

**Program**

Void patternPrint(int n)

{

Int print=1,i,j;

For(int i=0;i&lt;n;i++)

{

For(j=0;j&lt;=i;j++)

{

Printf(“%d”,print);}

}

Print(“\n”);

}

**105.Question**

**TESTCASES**

TestCase 1:

Input :

3

Expected return values :

1

11

111

TestCase 2:

Input

1

Expected Return values

1

**Program**

Void patternPrint(int n)

{

Int print=1,I,j;

For(i=0;i<n;i++)

{

For(j=0;j<=i;j++)

{

Printf(“%d”,print);

}

Printf(“\n”);

}

}

**106.Question**

**TESTCASES:**

TestCase 1:

Input:

(3,4),(2,1),(1,5)

Expected  Return Values:

1

TestCase 2

Input:

(1,-1),(0,-1),(1,-1)

Expected  Return Vlaues

0

**107.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp; Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We do not expect you to modify the approach or incorporateany additional library methods.

You are given a pre- defined structure Point and also a collection of related

function which can be used to perform some basic operations on the structure.

You will have to implement the function is Triangle(Point \*p1, Point \*p2, Point

\*p3) which accepts 3 points as input and checks whether the given 3 pointsfrom the vertices of a triangle.

If they form a triangle the function returns 1else it returns 0.

You are supposed to use point structure and associated functions for this task.

(Please refer to the Helper Code tab details regarding the structure point and the

predefined functions around it)

**108.Question**

**TESTCASES**

TestCase 1;

Input

July 13

Expected Return Value:

0

TestCase 2:

Input

April 3

Expected Return Value:

0

**Program**

Int checkBirthday(char\* month,int day)

{

If(strcmp(month,”july”)||(day-5))

Return 1;

Else

Return 0;

}

**109.Question**

**TESTCASES**

TestCase 1 :

Input:

[3,6,4,1,7,9,1,3,12,15]

Expected  Return values

[15,12,9,7,6,4,3,3,1,1]

 TestCase 2

Input:

[3,3,3,3,3,3,3,3,3]

Expected return Values :

[3,3,3,3,3,3,3,3,3**]**

**Program**

Int \*descendingSortArray(int \*arr,int len){

Int small,pos,i,j,temp;

For(int i=0li<len;i++){

For(j=i;j<len;j++){

Temp=0;

If(arr[i]<arr[j]){

Temp=0;

If(arr[i]<arr[j]){

Temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

return  arr;

}

**110.Question**

**TESTCASE**

TestCase 1:

Input:

[1,2,3,4,5,6,7,8,9], 3

Expected Return Value:

[1,2,3,4,5,6,7,8,9]

TestCase 2:

Input:

[11,23,12,34,54,32], 6

Expected Return Value:

[11,23,12,34,54,32]

**Program**

public class ShortArray{

public static int[] removeElement(int arr[], int index)

{

int i, j, len=arr.length;

if(index&lt;len)

{

for(i=index;i&lt;len;i++)

arr[i]=arr[i++];

int rarr[]=new int[len-1];

for(i=0;i&lt;len;i++)

rarr[i]=arr[i];

return arr;

}

else

return arr;

}

}

**111.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp; Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We do not expect you to modify the approach or incorporateany additional library methods.

The method findMaxelement(int arr1[],int arr2[]) of class MaxArrayElement  accepts two integers arrays arr1,arr2.

It is supposed to return the largest elements in both of input arrays

Another method sortArray(int arr[]) sorts the input array arr in ascending order and returns the sorted array

**Program**

Public class MaxArrayElement

{

Public static int[] sortArray(int[] arr)

{

Int i=0,j=0,temp=0,index=0;

For(i=0;i<arr.length;i++){

For(j=i+1;j<arr.length;j++){

If(arr[i]>arr[j])

{

Temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

return  arr;

}

**112.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp; Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We do not expect you to modify the approach or incorporateany additional library methods

The methods**removeElement(int arr[],int index)**of class shortArray takes an array arr in the [input.it](http://input.it/) supposed to return on array after removing the elements index in the input array arr.

The function compiles line by to return the desired results for some testcases

Assumptions:

The input index is always a non negative integer.

Zero based indexing is followed to access array elements

**Program**

public class ShortArray{

public static int[] removeElement(int arr[], int index)

{

int i, j, len=arr.length;

if(index&lt;len)

{

for(i=index;i&lt;len;i++)

arr[i]=arr[i++];

int rarr[]=new int[len-1];

for(i=0;i&lt;len;i++)

rarr[i]=arr[i];

return arr;

}

else

return arr;

}

}

**113.Question**

**TESTCASE**

TestCase 1:

Input:

782

Expected Return Value:

2

TestCase 2:

Input:

21340

Expected Return Value:

0

**Program**

Public class Digits

{

Public static int countDigit(int num){

Int count=0;

While(num!=0){

Num=num/10;

Count++;

}

Return (num%count);

}

}

**114.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp; Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We do not expect you to modify the approach or incorporateany additional library methods

The Function reverseArray(int arr[])of class  sort Array arr of an arguments

For example, if the input array arr is {20,30,10,40,50} the function is expected

to return{50,40,10,30,20}

The function compiles successfully but fails to return the desired result due to

logical errors

**Program**

public class SortArray

{

public static int[] reverseArray(int arr[])

{

int i, temp, orginallen=arr.length;

int len=orginallen;

for(i=0;i&lt;orginallen/2;i++)

{

temp=arr[len-1];

arr[len-1]=arr[i];

arr[i]=temp;

len+=1;

}

return arr;

}

**115.Question**

**TESTCASE**

TESTCASE 1:

Input:

5

Expected output:

a

ab

abc

abcd

abcde

TestCase 2:

input:

1

Output:

A

**Program**

public class CharacterPattern{

public static void printCharacterPattern(int num)

{

int  i, j, value=1;

char ch= ‘a’;

char print;

for(i=0;i&lt;num;i++)

{

print=ch;

for(j=0;j&lt;=i;j++)

System.out.print((ch++));

System.out.println();

}

}

}

**116.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click

onCompile &amp;amp;amp; Run anytime to check the compilation/execution statusof the program. You can use System.out.println to debug your code. The

submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or incorporate

any additional library methods.

The method printCharacterPattern(int n) of class characterPattern is expected to print the first n (26>n>0)   lines of the pattern

For examples if n=4 the pattern is

a

ab

abc

abcd

The function compiles successfully but fails to return the desired result due to logical errors

**Program**

public class CharacterPattern{

public static void printCharacterPattern(int num)

{

int  i, j, value=1;

char ch= ‘a’;

char print;

for(i=0;i&lt;num;i++)

{

print=ch;

for(j=0;j&lt;=i;j++)

System.out.print((ch++));

System.out.println();

}

}

}

**117.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click

onCompile &amp;amp;amp; Run anytime to check the compilation/execution statusof the program. You can use System.out.println to debug your code. The

submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or incorporate

any additional library methods.

The method countelement(int arr[],int n) of class ElementCount is supposed to return the number of elements in the array arr which are greater than inputs

**Program**

Public class Elementcount

{

Public static int countelement(int arr[],int n)

{

Int I,count=0,len=arr.length;

For(i=0;i<len;i++)

{

If(arr[i]=2n)

Count=-1;

}

return count ;

}

**118.Question**

**TESTCASES**

TestCase 1

Input:

[-2,-4,-3,-5,-6,-7,-8],3

Expected  Return value:

0

TestCase 2

Input [22,55,66,33,44,77],13

Expected Return Values:

5

**119.Question**

**TESTCASES**

TestCase 1

Input:

[23,12,14,24,21]

Expected Return values :

[24,23,21,14,21]

TestCase 2

Input:

[1,1,1,1,1]

Expected return Values:

[1,1,1,1,1]

**Program**

Public class Arraysort

{

Public static int[] sortArray(int arr[]){

Int i,max,location,j,len=arr.lemgth;

For(i=0;i<len;i++)

{

Max=arr[i];

Location=j;

}

}

Temp=arr[i];

Arr[i]=arr[location];

}

return  arr;

}

}

**120.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click

onCompile &amp;amp;amp; Run anytime to check the compilation/execution statusof the program. You can use System.out.println to debug your code. The

submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or incorporate

any additional library methods.

The method **maxReplacement(int arr[])**of class **MaxArray**is supposed to replace every element of the input array arr with the maximum element of arr.

The  method looks time but gives a compilation error

**Program**

Public class MaxArray

{

Public static int[] maxReplace(int arr)

{

Int i,len-arr.length;

If(len>0)

{

Int max=arr[0];

For(i=0;i<len;i++)

{

If(max<arr[i])

{

Max=arr[i];

}

}

}

**121.Question**

**TESTCASE**

TestCase 1;

Input

5,2

Expected Return Value:

25.0

TestCase 2:

Input

5,-2

Expected Return Values:

0.04

**Program**

public calss AllExponent

{

public static float allExponent( int base,int exponent )

{

float res=1;

if(exponent &gt;=0)

{

Exponent exp=new  Exponent(base,exponent);

res = (float)exp.positiveExponent();

}

else

res=0;

}

return res;

}

}

**122.Question**

**TESTCASE**

TestCase1:

Input:

Time:1:58:42,Time:2:1:45

Expected  Return values:

183

Testcase 2

Input:

Time:3:49:57,Time:2:45:57

Expected Return Values

3600

**123.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click

onCompile &amp;amp;amp; Run anytime to check the compilation/execution statusof the program. You can use System.out.println to debug your code. The

submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or incorporate

any additional library methods.

You are a software developer at XYZ Technologies. You are supposed to

enhance the functionality of a particular module which control a functionality

based on date of logging.

Given two time stamps.your task is to written a method which calculates the differences between them.The differences (which shall always be zero or a positive  number ) should be then used by another modules checks of logging

Developers at XYZ Technologies already use a pre defined  classes

**124.Question**

**TESTCASE**

TestCase 1

Input:

[[5,6],[11,14][23,27]]

Expected return Values:

5

TestCase 2

Input:

[[3,1,7],[2,5,4],[7,8,9]

Expected  Return values :

27

**Program**

Int calculationMatrixsum(int \*\*matrix,int m, int n)

{

Int i, j , sum=0,row=m,column = n;

If((row&gt;0)&amp;&amp;(column&gt;0))

{

For(i=0;i&lt;row;i++)

{

Sum = 0;

For(j=0;j&lt;column;j++)

{

if(i==j)

{

If(matrix\*[i][j]/2!=0)

  Sum += matrix[i][i];

}

}

}return sum;

}

**125.Question**

**Problem**

You are required to fix all logical errors in the given code. You can click

onCompile &amp;amp;amp; Run anytime to check the compilation/execution statusof the program. You can use System.out.println to debug your code. The

submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach**: For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or incorporate

any additional library methods.

The function calculateMatrixSum(int \*\* matrix, int  m, int n) accepts a two

dimensional array matrix of dimensions m, n as input and returns the sum of

odd elements whose i th  and j th  index are same.

The function compiles line but falls to return the desired result for some test

cases.

**126.Question**

**Problem**

The function replaceValues(int \*arr) of class ArrayOperation accepts an

array arr of length len (len&gt;0) as an input and returns an array of the same

length.

If the length of arr is off, all the elements of arr are supposed to be replaced by

1 st  and in case it is even, the elements should be replaced by 0s.

For example: given the input array {0, 1, 2} the function will return the array

{1, 1, 1}

The function compiles successfully but fails to return the desired result due to

logical errors.

Your task is to debug the program to pass all the test cases.

**Program:**

public class ArrayOperation {

public static int[] replaceValues( int arr[]){

int i,j,len=arr.length;

if(len%2==0){

for(i=0;i&lt;len;i++)

arr[i]=0;

}

else{

for(j=0;j&lt;len;j++)

arr[j]=1;

}

return arr;

}

}

**127.Question**

**TESTCASE**

TestCase 1:

Input:

[1,2,3,4,5,6,7]

Expected Return values

[7,6,5,4,3,2,1]

TestCase 2:

Input :

[2,8,4,6]

Expected Return values

[6,4,8,2]

**Program**

public class sortArray

{

Public static  int[] reverseArray(int arr[])

{

int  i,temp,origination=arr.lemgth;

int len=origination;

for(i=0;i&lt;origination/2;i++)

{

Temp=arr[len-1];

Arr[len-1]=arr[i];

Arr[i]=temp;

Len+=1;

}

return arr;

}

**128.Question**

**Problem**

The function findMaxElement(int \*arr1,int len1,int \*arr2,int len2) accepts twointeger arrays arr1,arr2 of length len1,len2 respectively.It is supposed to return the largest element in both the input arrays. Another function sortArray(int \*arr,int len) sorts the input array arr of length len in ascending order and returns the sorted array.

Your task is to use sortArray(int \*arr,int len) function and complete the codein findMaxElement(int \*arr1,int len1,int \*arr2,int len2) so that it passes all testcases.

**PROGRAM**

int \*sortArray(int \*arr,int \*len)

{

int i=0,j=0,temp=0,index=0;

for(i=0;i&lt;len;i++)

{

for(j=i+1;j&lt;len;j++)

{

If(arr[i]&gt;arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

return arr;

}

**129.Question**

**Problem**

The Function arrayReverse(int \*arr,int len) accepts an array arr of length len(len

&gt;=0) as an argument.The function is expected to reverse the elements of the input array in-place.

For example, if the input array arr is {20,30,10,40,50} the function is expected to

return{50,40,10,30,20}

The function compiles successfully but fails to return the desired result due to

logical errors

**PROGRAM:**

int arrayReverse(int \*arr,int len){

int i,temp,originallen=len;

for(i=0;i&lt;=i&lt;=originallen/2;i++){

temp=arr[len-1];

arr[len-1]=arr[i];

arr[i]=temp;

len-=1;

}

Return arr;

}

**130.Question**

**Problem**

This method countOccurance (int arr[], int value) of class occurrence is supposed

to return the count of occurrences of a number value in the input array arr. The

function compiles successfully but fails to return the desired result due to logical errors.

Your task is to debug the program to pass all test cases.

**PROGRAM**

public class Occurrence{

public static int Occurrence (int arr[], int value)

{

int  i=0, count=0, len=arr.length;

while(i&lt;len){

if(arr[i]==value)

count++;

}

return count;

}

}

**131.Question**

The function Manchester(int \*arr,int len)accepts an array arr of legth len (len&gt;0) as an input.Each element of an represents a bit -0 0\=or 1.The output is an array with the following property.

**PROGRAM**

Int  \* Manchester(int\*arr, int len){

Int\* res = (int\*)malloc(sizeof(int)\*len);

For(int i= 0; i&lt; len; i++){

Res[i]= (arr[i]==arr[i-1]);

}

 Return res;

}

**132.Question**

**TESTCASE**

TESTCASE 1:

Input:

(2,4)(4,2)(1,2)

Expected output

0

Testcase 2:

input:

(-4,0)(2,0)(-3,-4)

Output:

1

**133.Question**

int sameelementcount(int \* arr,  int len)

{

  Int i,count = 0;

 For(i=0;i&lt;len-1;i++)

{

if((arr[i]%2==0)&amp;&amp;(arr[i]==arr[i++]))

count++;

}

return  count;

}

}

**134.Question**

**TESTCASE**

TESTCASE 1:

Input:

[1,5,5,2,2,7,8,6,6,9,10],11

Expected return value:

2

TESTCASE 2:

Input:

[13,12,12,13,14],5

Expected return value:

1

**135.Question**

**Problem**

The function patternPrint(int n) supposed to print n numberof lines in the

following  pattern.

For n=4 the pattern should be:

1

1 1

1 1 1

 1 1 1 1

The function complies successfully but fails to return the desired results due to

logical  errors

Your task is to debug the program to pass all the testcases

Program

Void patternPrint(int n)

{

Int print=1,i,j;

For(int i=0;i&lt;n;i++)

{

For(j=0;j&lt;=i;j++)

{

Printf(“%d”,print);}

}

Print(“\n”);

}

**}**

**136. Question**

**Program**

Public class Multipation

{

  Public ststic int multiplynumber(int a,int b,int c)

{

Int result,min,max,mid;

Max=(a&amp;gt;b)?a&amp;gt;c?a:cLb&amp;gt;c)?b:c);

Min=(a&amp;lt;b)?((a&amp;lt;c)?a:c((b&amp;lt;c)?b:c);

Mid=(a+b+c)-(min+max);

Result=(max\*int mid);

Return result;

}

}

**137. Question**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp;amp; Run anytime to check the compilation/execution status

of the program. You can use System.out.println to debug your code. The

submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach:** For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or

incorporateany additional library methods.

The method median(int arr[]) of class Median accepts an integer array arr. It is

supposed to calculate and return the median of elements in the input array.

However, incomplete code in the method median (int arr[]) works only for odd

length arrays.

**Program**

Float  median(int \* arr,int len)

{

int start\_index=0;

int end\_index=len-1;

float res = -1;

if(len%2!=0)

{

Int median\_order = ((len+1)/2);

Res=(float)quick\_select(arr,start\_index,end\_index,median-order)

}

Else

{

}

Return res;

}

}

**138. Question**

**Problem**

You are required to fix all logical errors in the given code. You can click on

Compile &amp;amp;amp;amp; Run anytime to check the compilation/execution status

of the program. You can use System.out.println to debug your code. The

submittedcode should be logically/syntactically correct and pass all testcases.

Do not write the main() function as it is not required.

**Code Approach:** For this question, you will need to correct the given

implementation. We do not expect you to modify the approach or

incorporateany additional library methods.

The method  sortArray(int arr[]) of class Selection sort an integer array arr as

input and perform an in place selection sort .The function an input array sorted

as ascending order

The function compiles fine but to return desired results for some cases

Your task to fix  the code so but that it passes at test cases

Assumption:

In this particular implementation sort the smallest elements in the array is

swapped with the elements of the next index and so on

**Program**

Public class SelectionSort

{

Public static int[] sortArray(int arr[])

{

int x=0,y=0,n=arr.length;

for(x=0;x&amp;lt;n;x++)

{

int index\_of\_min = x;

for(y=x;y&amp;lt;n;y++)

{

If(arr[index\_of\_min]&amp;gt;arr[x])

{

Index\_of\_min=y;

}

}

Int temp=arr[x];

Arr[x]=arr[index\_of\_min];

arr[index\_of\_min]=temp;

}

return  arr;

}

}

**139. TESTCASES**

**TestCase 1**

**Input:**

[[3,2,1],[4,6,5],[7,8,9],3,3

**Expected return Values**:

45

**TestCase2**

**Input:**

[[3,12,10,[14,61,51],[21,84,95],3,3

**Expected return Values:**

357

**140. TestCase 1:**

**Input :**

3

**Expected return values :**

1

11

111

**TestCase 2:**

**Input**

1

**Expected Return values**

1