

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

[*www.MyPlacementPrep.com*](http://www.MyPlacementPrep.com)

Pro Material Series

Join Telegram Channel for more updates:

<https://t.me/MyPlacementprepApp>

Visit www.MyPlacementPrep.com .

India's No1 website for Placement Materials and Free mock test Series

Free Placement Learning path now available with free video course.

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
int calculationmatrixsum(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][i];
```

```
  } }
```

```
}
```

```
return sum; }
```

```
else return sum; }
```

```
/*
```

The method printpattern(int) of class drawpattern is expected to expected to print the first n (n > 0) Lines of the pattern

TESTCASES

TestCase 1

Input:

4

Expected Return value:

11

1111

111111

11111111

TestCase 2:

Input:

1

Expected Return Value:

11

```
*/
```

```
public class drawpattern {
```

```
public static void printpattern(int n) {
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
int i,j,print = 1; for(i=i;i<=n;i++) for(j=1;j<=2 * i;j++) System.out.print( (print ));  
System.out.println();
```

```
 } }
```

```
int matrixsum( int &matrix, int m, int n ) {
```

```
int i,j,sum=0; for(i=0;i<m;i++) {
```

```
for(j=0;j<n;j++)  
sum +=matrix(i)(j);
```

```
}
```

```
return sum; }
```

```
/*
```

```
TESTCASE 1
```

```
Input
```

```
5,7,4
```

```
Expected return value: 35
```

```
TESTCASE 2
```

```
Input
```

```
11,12,13
```

```
Expected return value: 156
```

```
*/
```

```
public class Multipation {
```

```
public static int multiplynumber(int a,int b,int c) {
```

```
int result,min,max,mid; max=(a>b)?a>c?a:cLb>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; }
```

```
}
```

```
/*
```

The method allExponent(int base,int exponent) of class All Exponent Accepts two integers base and exponents as inputs it is supposed

to calculate and return the results of exponents of base raised to power exponent for all the input values

However incomplete code in the method allExponent works only

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

positive values of exponent you will have to complete the code and make it works it works for negative values of exponents as well

Another method positiveExponent of class Exponent uses an efficient way for exponentiation but accepts only positive exponents values .you are supposed to use this method to complete the code i allExponent function

TESTCASE

TestCase 1;

Input

5,2

Expected Return Value:

25.0

TestCase 2:

Input

5,-2

Expected Return Values: 0.04

*/

```
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//odd {
```

```
            res=0; }
```

```
        return res;
```

```
    } }
```

/*

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

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

compiles fine but to return desired results for some cases Your task to fix the code so but that it passes at

test cases

15(1) TestCase 1;

Input

July 13

Expected Return Value: 0

TestCase 2:

Input

April 3

Expected Return Value: 0

```
*/
```

```
int checkBirthday(char* month,int day) {
```

```
if(strcmp(month,"july")||(day-5)) return 1;
```

```
else
```

```
return 0; }
```

```
int main() {
```

```
char inp[]="june";
```

```
int day=5; if(checkBirthday(inp,day)==1)
```

```
printf("Yes"); else
```

```
printf("No"); return 0 ;
```

```
}
```

```
/*
```

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

15(1) TestCase 1;

Input

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

July 13

Expected Return Value: 0

TestCase 2:

Input

April 3

Expected Return Value: 0

```
*/
```

```
int checkBirthDay(char* month,int day) {
```

```
if(strcmp(month,"july")||(day-5)) return 1;
```

```
else
```

```
return 0;
```

```
}
```

```
int main() {
```

```
char inp[]="june";
```

```
int day=5; if(checkBirthDay(inp,day)==1)
```

```
printf("Yes"); else
```

```
printf("No"); return 0 ;
```

```
}
```

```
/*
```

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 odd, 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.

```
*/
```

```
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;
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
} return arr;
```

```
}
```

```
/*
```

Input:

5

Expected output:

a

ab

abc

abcd

abcde

testcase 2:

input:

1

Output:

a

```
*/
```

```
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( ** );
```

```
}
```

```
} }
```

```
/*
```

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.

```
*/
```

```
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)
```

```
{
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
int median_order=((len+1)/2);
medianCalculate ob=new medianCalculate(arr,start_index,end_index,median_order);
res=ob.quick_select();
```

```
    } else {
```

```
    }
```

```
return res; }
```

```
}
```

```
/*
```

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.

TESTCASE 1:

Input:

[2, 5, 1, 3, 9, 8, 4, 6, 5, 2, 3, 11], 12,

[11, 13, 2, 4, 15, 17, 67, 44, 2, 100, 0, 23]11 Expected Return Value:

100

TESTCASE 2:

Input:

[100, 22, 43, 912, 56, 89, 85], 7, [234, 123, 456, 234, 890, 101], 6 Expected Return Value:

912

```
*/
```

```
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]) {
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
temp=arr[i]; arr[i]=arr[j]; arr[j]=temp;
```

```
} }
```

```
}
```

```
return arr; }
```

```
findMaxElement(int *arr1,int len1,int *arr2,int len2) {
```

```
//write down the code }
```

```
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");
```

```
} }
```

```
/*
```

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) */

```
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++) {
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
if(arr[j]==arr[i]) {  
  
arr[k]=arr[k+1]; }  
  
len=len-1; count=count+1; j=i;  
  
}  
int newlength = originalLength-count;  
int newArr[]=new int[newlength];  
for(i=0;i<newLength;i++)  
newArr[i]=arr[i];  
return newArr;  
}  
}  
/*
```

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, if 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.

TESTCASE 1:

Input:

[1, 1, 0, 0, 1, 0] Expected Return Value: [1, 0, 1, 0, 1, 1] TESTCASE 2:

Input:

[0, 0, 0, 1, 0, 1, 1, 1] Expected Return Value: [0, 0, 0, 1, 1, 1, 0

*/

```
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++) {
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

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

```
 } }
```

```
 }
```

```
return res; }
```

```
 }
```

/*You are required to fix all logical errors in the given code. You can click on Compile & 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 TESTCASE 1:

Input:

[23, 12, 14, 24, 21], 5

Expected Return Value:

[24, 23, 21, 14, 12]

TESTCASE 2:

Input:

[1, 1, 1, 1, 1], 5 Expected Return Value: [1, 1, 1, 1, 1]

```
*/
```

```
int * sortArray(int *arr,int len ) {
```

```
int i,max,location,j,temp; for(i=0;i<len;i++)
```

```
{
```

```
max=arr[i]; location=i; for(j=i+1;j<len;j++) {
```

```
if(max<arr[j]) {
```

```
max=arr[j];
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
location=j; }
```

```
    } temp=arr[location]; arr[location]=arr[i]; arr[i]=temp;
```

```
    }
```

```
    return arr;
```

```
    } }
```

```
/*
```

TESTCASE 1:

Input:

[2, 5, 8, 11, 3], 5 Expected Return Value: [11, 11, 11, 11, 11]

TESTCASE 2:

Input:

[3, 2, 5, 8, 9, 11, 23, 45, 63], 9 Expected Return Value:

[63, 63, 63, 63, 63, 63, 63, 63, 63] */

```
int* maxReplace(int *arr, int len) {
```

```
    int i; if(len>0) {
```

```
        int max=arr[0]; for(i=0;i<len;i++) {
```

```
            if(max<arr[i]) max=arr[i];
```

```
        } }
```

```
        for(i=0;i<len;i++) arr[i]=max; return arr;
```

```
    }
```

```
/*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. */
```

```
public class Occurrence
```

```
{
```

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

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
int i=0, count=0, len=arr.length; while(i<len)
{
    if(arr[i]==value) count++;
}

return count; }
```

```
}
/*
```

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.

```
*/
int getarraysum(int *arr,int len)
```

```
{
    int sum = 0; for(i=0;i<len;i-i-1) {
```

```
        sum = arr[i]; }
```

```
    return sum; }
```

/* The Least-Recently-Used(LRU) cache algorithm exists the element from the cache(when it'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.

37. 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

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

TESTCASE 2:

Input:

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

Expected Return Value:

8

Explanation:

The following page numbers are missed one after the other 2,3,1,2,1,4,3,2. This results in 8 page misses.

*/

/*You are a software developer at ABC technologies. You are supposed to enhance the functionality of a particular module which controls a functionality based on date of logging. Given two dates, your task is to write a method which calculates the difference between them. The difference (which shall always be zero or 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 methods 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.

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

*/

Public class DateComparer

{

Public static int difference_in_dates(Date date1, Date date2)

{

//write your code

return 0;

}

}

/*

The methods GetDigitSum(int arr[]) of class DigitSum accepts an integer 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

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

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

```
*/
public class Digitsum {

public static int getDigitSum(int arr[]) {

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]; }

} }

int getsum(int num) {

int sum=0;

while (num) {

sum=sum+(num%10);

num=num/10; }

return sum; }

/*The function patternPrint(int n) supposed to print n number of lines in

the following pattern
For n=4 the pattern should be:
1
11
111
1111
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
*/
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
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"); }
```

```
}
```

/*The function countDigits (int num) is supposed to return the remainder when the input argument num (num>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

```
*/
```

```
int countDigits(int num)
{
```

```
int count=0; while(num!=0) {
```

```
num=num/10;
```

```
count++; }
```

```
return (num%count); }
```

```
/*
```

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

```
*/
```

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

```
int i,count=0;
```

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

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
if(arr[i]>(2*n)) {  
count+=1;
```

```
}  
return count;
```

```
}  
/*
```

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]

Expected Return Value: [6, 4, 8, 2]

```
*/
```

```
public class SortArray {
```

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

```
int i, temp, orginallen=arr.length; int len=orginallen; for(i=0;i<orginallen/2;i++)  
{
```

```
temp=arr[len-1]; arr[len-1]=arr[i]; arr[i]=temp; len+=1;
```

```
}
```

```
return arr; }
```

```
}  
/*
```

The function removeElement(int *arr,int len,int index)takes an array arr of length len as 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

```
*/
```

```
int* removeelement( int *arr,int len,int index)
```

```
{  
int i,j;
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
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; }
```

/*Problem

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[y])  
{  
Index_of_min=y;  
}  
}  
Int temp=arr[x];  
Arr[x]=arr[index_of_min];  
arr[index_of_min]=temp;
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
}
return arr;
}
}
Public class arrayCount
{
Public static int sameElementCount(int arr[])
{
int len=arr.length,i,count=0;

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

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

return count;
}
}
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; }

} }

public calss Array {

public static int[] sortArray(int arr[]) {
int len=arr.length;
int small,pos,i,j,temp; for(i=0;i<=len-1;i++)
```

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
{ for(j=i;j<len;j++) {  
Temp=0; if(arr[i]>arr[j])  
{  
temp=arr[i]; arr[i]=arr[j];  
}  
}  
}  
return arr;  
}  
}
```

```
public class Array {
```

```
public static int[] sortArray(int arr[]) {  
int len=arr.length;  
int small,pos,i,j,temp;
```

```
for(i=0;i<=len-1;i++) {  
for(j=i;j<len;j++)  
{
```

```
Temp=0; if(arr[i]>arr[j]) {  
temp=arr[i]; arr[i]=arr[j];  
}  
}  
}  
return arr;  
}  
}  
/*
```

Problem

You are required to complete the given code by reusing existing functions .you can click on compile 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 as 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(Point *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.

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

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

*/

/*You are required to complete the given code by reusing existing functions.click on the helper code tab to find out the details of functions/classes provided for reuse you can click on compile&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 testcase . . 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 given a pre defined class PrimeBank containing of methods to perform some basic operations

You will have to implement the methods PrintPrime(int num,int n)which accepts initial number num and number of digits n as inputs and prints all the numbers of n digits starting from the initial number num

you supposed to use Prime Bank class and associated methods of task

*/

/*

The function findMinElement(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 smallest 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 findMinElement(int *arr1,int len1,int *arr2,int len2)

so that it passes all test cases.

*/

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;

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

```
} }

} return arr;

}
public static int findMinElement(int arr1[],int len1 , int arr2[],int len2) {

//your code

} }

#include<stdio.h> Int main()
{
int num,count=0; scanf("%d",&num); while(num)

{
Num=num/10;
Count++;
}
Printf("%d",count);
}
/*Input
[1,1,1,0,1,1,1,1],2
Expected return value: [0,0,0,0,0,1,1,0]
*/
int* cellcompete(int* cells, int days) {

}

int main() {

int cells[8]={ 1,1,1,0,1,1,1,1 };
int days=2,index; cellcompete(cells,days); for(index=0 ; index < 8 ; index++)

printf("%d ",cells[index]); }

/*:
```

You are required to complete the given code by reusing existing functions.click on the helper code tab to find out the details of functions/classes provided for reuse you can click on compile&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 testcase . . 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.

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

The function countElement(int *arr,int len,int n) is supposed to return the numbers of elements in the inputs array arr of length len,which are greater than input number n

The function looks fine but given a compilation error

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

PROGRAM:

```
*/
```

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

```
    int i,count=0;
```

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

```
        if(arr[i]>2n)
```

```
        {
```

```
            count++;
```

```
        }
```

```
    return count; }
```

www.MyPlacementPrep.com

*Free Mock Test and Video
Tutorial*

Visit www.MyPlacementPrep.com .

India's No1 website Placement and Mock Test series

Free Placement Learning Path with Free Video Course.

Join

Join

p

Free Placement Video Tutorial with Mock Test visit:

www.MyPlacementPrep.com

Join Telegram Channel: <https://t.me/MyPlacementprepApp>

Join Telegram Group: <https://t.me/myPlacementPrep>