

Q20) Eight persons A, B, C, D, E, F, G, and H are sitting around a square table. All of them are facing towards the center, Only one person sits on each of the sides and only one person sits on each of the corners. Four of them sit on the sides and four of them sit on the corners. C is sitting 3rd to the left of B, who sits on a side. D is sitting opposite to H, who is sitting 3rd to the right of B. A is sitting 2nd to the right of G. G is not sitting opposite to either E or F. F is not an immediate neighbor of H. Who among the following is sitting opposite to A?

A. F

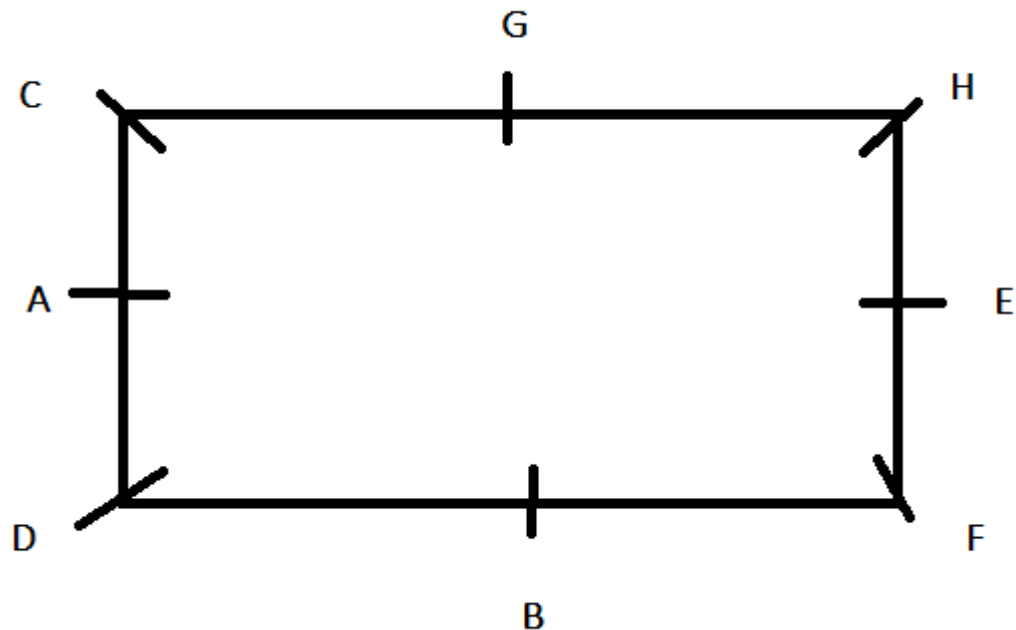
B. B

C. E

D. C

Answer: Option C

Explanation:



TCS NQT Advanced Coding Previous Year Questions

Q1) Given an array of integers where every element appears even number of times except one element which appears odd number of times, write a program to find that odd occurring element in $O(\log n)$ time. The equal elements must appear in pairs in the array but there cannot be more than two consecutive occurrences of an element.

For example :

3

2 3 2

It doesn't have equal elements appear in pairs

7

1 1 2 2 2 3 3

It contains three consecutive instances of an element.

5

2 2 3 1 1

It is valid and the odd occurring element present in it is 3.

Enter only valid inputs.

Sample Input :

5

2 2 3 1 1

Sample Output :

3

Solutions:

C Language

```
#include <stdio.h>
```

```
int main()
{
    int n; scanf("%d",&n);
    int arr[n];
    for(int i=0 ; i<n ; i++){
        scanf("%d",&arr[i]);
    }
    int left=0,right=n-1,mid,pre,nxt;
    if(arr[0] != arr[1]){
        printf("%d ",arr[0]);
    }
    else if(arr[n-1] != arr[n-2]){
        printf("%d ",arr[n-1]);
    }
    else{
        while(left <= right){
            mid = ((right-left)/2)+left;
            pre = mid-1;
            nxt = mid+1;
            if((arr[pre] != arr[mid]) && (arr[nxt] != arr[mid])){
                printf("%d ",arr[mid]);
            }
        }
    }
}
```

```

        break;
    }
    else if(mid%2==0){
        if(arr[pre] == arr[mid])
            right = mid - 1;
        else
            left = mid + 1;
    }
    else{
        if(arr[pre] == arr[mid])
            left = mid + 1;
        else
            right = mid - 1;
    }
}
}
return 0;
}

```

C++

```

#include <iostream>
#include <vector>
using namespace std;

int main()
{
    int n; cin >> n;
    vector<int> v(n);
    for(int i=0 ; i<n ; i++){
        cin >> v[i];
    }
    int left=0,right=n-1,mid,pre,nxt;
    if(v[0] != v[1]){
        cout << v[0];
    }
    else if(v[n-1] != v[n-2]){
        cout << v[n-1];
    }
    else{
        while(left <= right){
            mid = ((right-left)/2)+left;
            pre = mid-1;
            nxt = mid+1;
            if((v[pre] != v[mid]) && (v[nxt] != v[mid])){
                cout << v[mid];
                break;
            }
            else if(mid%2==0){
                if(v[pre] == v[mid])
                    right = mid - 1;
                else

```

```

        left = mid + 1;
    }
    else{
        if(v[pre] == v[mid])
            left = mid + 1;
        else
            right = mid - 1;
    }
}
}
return 0;
}

```

JAVA

```

import java.util.*;
public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int arr[] = new int[n];
        for(int i=0 ; i<n ; i++){
            arr[i] = sc.nextInt();
        }
        int left=0,right=n-1,mid,pre,nxt;
        if(arr[0] != arr[1]){
            System.out.print(arr[0]);
        }
        else if(arr[n-1] != arr[n-2]){
            System.out.print(arr[n-1]);
        }
        else{
            while(left <= right){
                mid = ((right-left)/2)+left;
                pre = mid-1;
                nxt = mid+1;
                if((arr[pre] != arr[mid]) && (arr[nxt] != arr[mid])){
                    System.out.print(arr[mid]);
                    break;
                }
                else if(mid%2==0){
                    if(arr[pre] == arr[mid])
                        right = mid - 1;
                    else
                        left = mid + 1;
                }
                else{
                    if(arr[pre] == arr[mid])
                        left = mid + 1;
                    else
                        right = mid - 1;
                }
            }
        }
    }
}

```

```

    }
  }
}
}
}

```

PYTHON

```

n = int(input())
a = list(map(int,input().split()))
left=0
right=n-1
if a[0] != a[1]:
    print(a[0])
elif a[n-1] != a[n-2]:
    print(a[n-1])
else:
    while left <= right:
        mid = ((right-left)//2)+left
        pre = mid-1
        nxt = mid+1
        if (a[pre] != a[mid]) and (a[nxt] != a[mid]):
            print(a[mid])
            break
        elif mid%2==0 :
            if a[pre] == a[mid] :
                right = mid - 1
            else :
                left = mid + 1
        else :
            if a[pre] == a[mid] :
                left = mid + 1
            else :
                right = mid - 1;

```

Join the TCS NQT Cracker Course

➤ Includes **Ninja + Digital** training with Mock Interviews

[Click to Join Now](#)

We assure you to prepare you with the best material in the simplest way possible!



Q2) Given an array of integers and a sum, the task is to count all subsets of given array with sum equal to given sum.

Input :

The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. Each test case contains an integer n denoting the size of the array. The next line contains n space separated integers forming the array. The last line contains the sum.

Output :

Count all the subsets of given array with sum equal to given sum.

NOTE: Since result can be very large, print the value modulo 109+7.

Constraints :

1<=T<=100

1<=n<=103

1<=a[i]<=103

1<=sum<=103

Example :

Input :

2

6

2 3 5 6 8 10

10

5

1 2 3 4 5

10

Output :

3

3

Explanation :

Testcase 1: possible subsets : (2,3,5) , (2,8) and (10)

Testcase 2: possible subsets : (1,2,3,4) , (2,3,5) and (1,4,5)

Solutions:

C++

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
void printBool(int n, int len)
```

```
{
```

```
while (n) {
```

```
    if (n & 1)
```

```
        cout << 1;
```

```
    else
```

```
        cout << 0;
```

```
    n >>= 1;
```

```
    len--;
```

```
}
```

```
while (len) {
```

```
    cout << 0;
```

```
    len--;
```

```
}
```

```
cout << endl;
```

```
}
```

```
void printSubsetsCount(int set[], int n, int val)
```

```
{
```

```

int sum;
int count = 0;
for (int i = 0; i < (1 << n); i++) {
    sum = 0;

    for (int j = 0; j < n; j++)

        if ((i & (1 << j)) > 0) {
            sum += set[j];
        }

        if (sum == val) {

            count++;
        }
    }
    if (count == 0) {
        cout << ("No subset is found") << endl;
    }
    else {
        cout << count << endl;
    }

}

int main()
{
    int t,n,sum;
    cin>>t;
    while(t--)
    {
        cin>>n;
        int set[n];
        for(int i=0;i<n;i++)
            cin>>set[i];
        cin>>sum;
        printSubsetsCount(set, n, sum);
    }
}

```

JAVA

```

import java.io.*;
import java.util.*;
class Main {

    static void printBool(int n, int len)
    {

        while (n>0) {

```

```

        if ((n & 1) == 1)
            System.out.print(1);
        else
            System.out.print(0);

        n >>= 1;
        len--;
    }

    while (len>0) {
        System.out.print(0);
        len--;
    }
    System.out.println();
}

static void printSubsetsCount(int set[], int n, int val)
{
    int sum;
    int count = 0;
    for (int i = 0; i < (1 << n); i++) {
        sum = 0;
        for (int j = 0; j < n; j++)

            if ((i & (1 << j)) > 0) {
                sum += set[j];
            }

        if (sum == val) {

            count++;
        }
    }

    if (count == 0) {
        System.out.println("No subset is found");
    }
    else {
        System.out.println(count);
    }
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    int t = sc.nextInt();
    int n,sum;
    while(t>0)
    {
        n=sc.nextInt();
        int set[] = new int[n];

```



```

        for(int i=0;i<n;i++)
        set[i] = sc.nextInt();
        sum = sc.nextInt();
        printSubsetsCount(set, n, sum);
        t--;
    }

}
}

```

PYTHON

```

def printBool(n, len):
    while n:
        if n & 1:
            print("1 ")
        else:
            print("0 ")
        n = n >> 1
        len -= 1

    while len:
        print("0 ")
        len -= 1

    print()

def printSubsetsCount(set, n, val):
    sum = 0
    count = 0
    for i in range(0, 1 << n):
        sum = 0

        for j in range(0, n):

            if (i & 1 << j) > 0:
                sum += set[j]

        if (sum == val):

            count += 1

    if (count == 0):

        print("No subset is found")

    else:

        print(count)

t=int(input())
set = []
while t>0:
    n = int(input())

```

```

set = list((map(int,input().strip().split()))[:n])
Sum = int(input())
printSubsetsCount(set, n, Sum)
t = t-1

```

Q3) Before the outbreak of corona virus to the world, a meeting happened in a room in Wuhan. A person who attended that meeting had COVID-19 and no one in the room knew about it! So everyone started shaking hands with everyone else in the room as a gesture of respect and after meeting unfortunately every one got infected! Given the fact that any two persons shake hand exactly once, Can you tell the total count of handshakes happened in that meeting?

Input Format :

The first line contains the number of test cases T, T lines follow.

Each line then contains an integer N, the total number of people attended that meeting.

Output Format :

Print the number of handshakes for each test-case in a new line.

Constraints :

1 <= T <= 1000

0 < N < 106

Sample Input :

2

1

2

Output :

0

1

Explanation :

Case 1 : The lonely board member shakes no hands, hence 0.

Case 2 : There are 2 board members, 1 handshake takes place.

Solutions:

C Language

```
#include <stdio.h>
```

```

int main()
{
    int t,n; scanf("%d",&t);
    long int sum = 0;
    while(t--){
        scanf("%d",&n);
        n--;
        sum = (n*(n+1))/2;
        printf("%ld\n",sum);
    }
    return 0;
}

```

C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```

{
    int t,n,sum; cin >> t;
    long long sum;
    while(t--){
        cin >> n;
        n--;
        sum = (n*(n+1))/2;
        cout << sum << endl ;
    }
    return 0;
}

```

JAVA

```

import java.util.*;
public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int t,n;
        long sum;
        t = sc.nextInt();
        while(t > 0){
            n = sc.nextInt();
            n--;
            sum = (n*(n+1))/2;
            System.out.println(sum);
            t--;
        }
    }
}

```

PYTHON

```

t = int(input())
while t > 0 :
    n = int(input())
    n = n-1;
    print((n*(n+1))/2)
    t = t-1

```

Q4) For enhancing the book reading, the school distributed story books to students as part of the Children's Day celebrations. To increase the reading habit, the class teacher decided to exchange the books every week so that everyone will have a different book to read. She wants to know how many possible exchanges are possible.

If they have 4 books and students, the possible exchanges are 9. Bi is the book of i-th student and after the exchange, he should get a different book, other than Bi.

B1 B2 B3 B4 – first state, before exchange of the books

B2 B1 B4 B3

B2 B3 B4 B1

B2 B4 B1 B3

B3 B1 B4 B2

B3 B4 B1 B2

B3 B4 B2 B1
B4 B1 B2 B3
B4 B3 B1 B2
B4 B3 B2 B1

Find the number of possible exchanges, if the books are exchanged so that every student will receive a different book.

Constraints

$1 \leq N \leq 1000000$

Input Format

Input contains one line with N, indicates the number of books and number of students.

Output Format

Output the answer modulo 100000007.

Refer the sample output for formatting

Sample Input :

4

Sample Output :

9

Solutions:

C Language

```
#include <stdio.h>
```

```
int main()
{
    int mod = (int)1e7+7;
    long int n,a=0,b=1,c=2,d,i;
    scanf("%ld",&n);
    if(n==1 || n==2)
        printf("%ld",n-1);
    else{
        for(i=3 ; i<=n ; i++){
            d = (c * (a + b)) % mod ;
            a = b;
            b = d;
            c++;
        }
        printf("%ld",d);
    }
    return 0;
}
```

C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
{
    int mod = (int)1e7+7;
    long int n,a=0,b=1,c=2,d,i;
    cin >> n;
    if(n==1 || n==2)
        cout << (n-1);
}
```

```

else{
    for(i=3 ; i<=n ; i++){
        d = (c * (a + b)) % mod ;
        a = b;
        b = d;
        c++;
    }
    cout << d;
}
return 0;
}

```

JAVA

```

import java.util.*;
public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int mod = (int)1e8+7;
        long n,a=0,b=1,c=2,d=0;
        n = sc.nextLong();
        if(n==1 || n==2)
            System.out.println(n-1);
        else{
            for(int i=3 ; i<=n ; i++){
                d = (c * (a + b)) % mod ;
                a = b;
                b = d;
                c++;
            }
            System.out.println(d);
        }
    }
}

```

PYTHON

```

mod = 100000007;
a=0
b=1
c=2
d=0
n = int(input())
if n==1 or n==2:
    print(n-1)
else :
    for i in range(1,n) :
        d = (c * (a + b)) % mod
        a = b
        b = d
        c = c + 1
    print(d)

```

Q5) You are given a string A and you have to find the number of different sub-strings of the string A which are fake palindromes.

Note:

1. Palindrome: A string is called a palindrome if you reverse the string yet the order of letters remains the same. For example, MADAM.
2. Fake Palindrome: A string is called as a fake palindrome if any of its permutations is a palindrome. For example, AAC is fake palindrome, but ACD is not.
3. Sub-string: A sub-string is a contiguous sequence (non-empty) of characters within a string.
4. Two sub-strings are considered same if their starting indices and ending indices are equal.

Input Format:

First line contains a string S

Output Format:

Print a single integer (number of fake palindrome sub-strings).

Constraints:

$1 \leq |S| \leq 2 * 10^5$

The string will contain only Upper case 'A' to 'Z'

Sample Input 1:

ABAB

Sample Output 1:

7

Explanation:

The fake palindrome for the string ABAB are A, B, A, B, ABA, BAB, ABAB.

Sample Input 2:

AAA

Sample output 2:

6

Explanation:

The fake palindrome for the string AAA are A, A, A, AA, AA, AAA

Solutions:

C++

```
#include <bits/stdc++.h>
using namespace std;
```

```
void countSubstring(string s)
{
    int res = 0;

    for (int i = 0;
        i < s.length(); i++) {

        int x = 0;

        for (int j = i;
            j < s.length(); j++) {

            int temp = 1 << s[j] - 'a';

            x ^= temp;
            if ((x & (x - 1)) == 0)
                res++;

        }

    }
    cout << res;
}

int main()
{
    string str;
    getline(cin, str);
    countSubstring(str);
    return 0;
}
```

JAVA

```
import java.util.*;
class Main{
    static void countSubString(String s)
    {
        int res = 0;
        for (int i = 0; i < s.length(); i++)
        {
            int x = 0;
            for (int j = i; j < s.length(); j++)
            {
                int temp = 1 << s.charAt(j) - 'a';
                x ^= temp;
                if ((x & (x - 1)) == 0)
                    res++;
            }
        }
    }
}
```

```

    }

    }
    System.out.print(res);

}
public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    String str = sc.nextLine();
    countSubString(str);

}
}

```

PYTHON

```

def countSubString(s):

    res = 0;

    for i in range(len(s)):
        x = 0;
        for j in range(i, len(s)):

            temp = 1 << ord(s[j]) - ord('a');

            x ^= temp;
            if ((x & (x - 1)) == 0):
                res += 1;

    print(res);

if __name__ == '__main__':
    str = input();

    countSubString(str);

```
