

BUBT Intra University Programming Contest

Half yearly Preliminary- 2022 Editorial (Junior Division)

Problem Name: Wish You Eid Mubarak

Problem Setter: Jahin Hossain

Problem Type: Basic

Solution: Just copy the given code and change the year.

```
#include<stdio.h>
int main() {
    printf("Eid Mubarak 2022\n");
    return 0;
}
```

Problem Name: Good number

Problem Setter: All Moon Tasir

Problem Type: Basic

Approach: Check if the modulus of the given number is zero when divided by 2.

Solution:

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    if(n%2==0) {
        printf("Good\n");
    }
    else{
        printf("Bad\n");
    }
    return 0;
}
```

Problem Name: Winning the Game

Problem Setter: Maruf Ahmed Rahad

Problem Type: Condition

Solution: check if the number is 0, print Jahin, else if it is 1 print Tasir

```
#include<stdio.h>

int main() {
    int n;
    scanf("%d",&n);
    if(n==0) {
        printf("Jahin\n");
    }
    else if(n==1) {
        printf("Tasir\n");
    }
    return 0;
}
```

Problem Name: Division selection

Problem setter: Maruf Billah

Problem Type: Condition

Approach: Keep one condition to check if the number is between 43 and 46, another one to check if it is between 47 and 49, if the is in none of them, print out of contest

Solution:

```
#include<stdio.h>

int main() {
    int n;
    scanf("%d",&n);
    if(n>=47 && n<=49)
        printf("junior division\n");
    else if(n>=43 && n<=46)
        printf("senior division\n");
    else
        printf("out of contest\n");
    return 0;
}
```

Problem Name: Climbing the mountain!**Problem Setter:** Habibullah**Tester :** Jahin Hossain**Problem Type:** Complex conditions

Approach: Choose a number and check if there is one number that is greater than this number and other two are less than the number. Do this check for every number, and print the one that satisfies all conditions. From a,b,c,d a will be the third highest number,

if $b > a$ and $a > c$ and $a > d$,

Else if $c > a$ and $a > b$ and $a > d$,

Else if $d > a$ and $a > b$ and $a > c$.

Check these same conditions for b,c and d, and find the answer.

Authors solution:

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

```
int main() {
    int a,b,c,d;
    scanf("%d %d %d %d",&a,&b,&c,&d);
    if((a<b && a<c && a>d) || (a<c && a<d && a>b) || (a<b && a<d && a>c)) {
        printf("%d\n",a);
    }
    else if((b<a && b<c && b>d) || (b<c && b<d && b>a) || (b<a && b<d && b>c)) {
        printf("%d\n",b);
    }
    else if((c<b && c<a && c>d) || (c<a && c<d && c>b) || (c<b && c<d && c>a)) {
        printf("%d\n",c);
    }
    else{
        printf("%d\n",d);
    }
    return 0;
}
```

Problem Name: Counting mango

Problem Setter: Maruf Billah

Tester : Jahin Hossain, ALL MOON TASIR

Problem Type: Loops or Equations

Approach: Use the formula for the summation of first N natural numbers. You can also run a loop from 1 to N, and keep adding the numbers to another variable "sum".

Author Solution :

Using Loop:

```
#include<stdio.h>

int main ()
{
    int n,i,sum=0;
    scanf("%d",&n);
    for (i=1;i<=n;i++)
    {
        sum=sum+i;
    }
    printf("%d\n",sum);
}
```

Using formula:

```
#include<stdio.h>

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

Problem Name: Birthday Party!

Problem Setter: Habibullah

Tester : Jahin Hossain and All Moon Tasir

Problem Type: Basic array traversal

Approach: Just iterate through the array and keep adding the values to a variable 'sum'. Don't forget to use "long long int", since there can be overflow if 'int' is used. Since there can be a maximum 10^{10} number in the array.

Author Solution :

```
#include<stdio.h>
int main() {
    int n;
    long long int arr[11];
    scanf("%d", &n);
    for(int i=0; i<n; i++) {
        scanf("%lld", &arr[i]);
    }
    long long int sum=0;
    for(int i=0; i<n; i++) {
        sum+=arr[i];
    }
    printf("%lld\n", sum);
    return 0;
}
```

Problem Name: Less or equal even odds

Problem Setter: Jahin Hossain

Problem Type: Nested loops

Approach: First run a loop of i, from 1 to N, then run another loop of j inside it. The second loop will run from 1 to i. In the second loop, check if $i\%2==j\%2$, this is checking they are both even or both odd. If yes, then print j and a space. Don't forget to print the value of i in the given format before running the second loop.

Author Solution:

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

```
int main() {
    int t, tk, n, i, j;
    scanf("%d", &t);
    for(tk=1; tk<=t; tk++) {
        scanf("%d", &n);
        for(i=1; i<=n; i++) {
            printf("%d -> ", i);
            for(j=1; j<=i; j++) {
                if(i%2==j%2) {
                    printf("%d ", j);
                }
            }
            printf("\n");
        }
    }
    return 0;
}
```

Problem Name: ANOTHER GOOD NUMBER**Problem Setter:** All Moon Tasir**Problem Type:** String to integer

Approach: Look at the constraints $0 < N \leq 10^{100}$, you can not take the input into an integer. You have to take the input into a string.

Use the basic concept you learnt at school. "A number is divisible by 3, if the sum of its digits is divisible by 3".

So iterate through the string and for every index, take the character and convert it into an integer digit, by subtracting the ascii value of '0' from it, that is $\text{digit} = s[i] - '0'$. Now, keep adding these digits to a variable "sum", which was initially 0. After the iteration, just check if the sum is divisible by 3 or not. Then print your answer.

Author Solution:

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

```
int main() {
    char str[30];
    int sum=0;
    scanf("%s",str);
    for(int i=0;str[i]!='\0';i++) {
        sum+=(str[i]-'0');
    }
    if(sum%3==0) {
        printf("Good\n");
    }
    else{
        printf("Bad\n");
    }
    return 0;
}
```

Problem Name: Distance**Problem setter:** Jahin Hossain**Problem Type:** BFS graph algorithm

Approach: This is a problem with the basic BFS algorithm. First you have to build the graph, since the number of nodes is huge, you have to use Vector to store it. You can choose either 'a' as the source node or even 'b', since it is an bi-directional graph. That is, if you can go from a to b, you can go from b to a using the same path. Now, if you choose 'a' as the source, after running BFS, just check if the node 'b' was visited. If yes, then print the distance, else it is not reachable from 'a'. Also you could have kept -1 into the dis array for all nodes, in that case you don't have to check anything at the end, just print the value on dis[b].

Learn BFS algorithm - <http://www.shafaetsplanet.com/?p=604>

Author Solution:

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

const int MM = 1e5+6;
vector<int> grph[MM];
int dis[MM],vis[MM];

void bfs(int s){
    queue<int> qq;
    qq.push(s);
    dis[s]=0;
    vis[s]=1;
    while(!qq.empty()){
        int u = qq.front();
        qq.pop();
        vis[u]=1;
        for(int i=0;i<grph[u].size();i++){
            int v = grph[u][i];
            if(!vis[v]){
                qq.push(v);
                dis[v]=dis[u]+1;
                vis[v]=1;
            }
        }
    }
}
```



```
}  
int main() {  
    int n,m,u,v,a,b;  
    cin>>n>>m;  
    for(int i=0;i<=n;i++) {  
        dis[i]=-1;  
    }  
    for(int i=0;i<m;i++) {  
        cin>>u>>v;  
        grph[u].push_back(v);  
        grph[v].push_back(u);  
    }  
    cin>>a>>b;  
    bfs(a);  
    cout<<dis[b]<<"\n";  
}
```

Problem Name: Weird stone sum**Setter:** Jahin Hossain**Problem Type:** Dynamic Programming

Approach: This is a variation of coin change problem. Think of the numbers written on the stones as the values of coins. Think that you have to determine the number of ways you can give change to give a change of the number N using these 8 coins, now just solve the problem using coin change algorithm and print the answer modulo 10^9+7 . Note that you have to apply the modulus after every addition or multiplication operations, and don't forget to use 'long long int'.

Know more about coin change algorithm - <http://www.shafaetsplanet.com/?p=3638>

Author's Solution:

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

int main() {
    int mod = 1e9+7;
    int t,n;
    vector<int> dp(5001,0);
    dp[0] = 1;
    for (int i = 1; i <= 5001; i++) {
        for (int j = 1; j <= 8 && i-j >= 0; j++) {
            (dp[i] += dp[i-j]) %= mod;
        }
    }
    cin>>t;
    while(t--){
        cin >> n;
        cout << dp[n] << endl;
    }
    return 0;
}
```

Editorial Written By:

Jahin Hossain

Intake: 41

Dept. of CSE, BUBT