

Question1: Given n you have to find sum of first n natural numbers

Constraints

$1 \leq t \leq 10^5$

$1 \leq n \leq 10^9$

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

```
int main()
```

```
{
```

```
    int t;
```

```
    long long int n,ans;    //Here we have to use long long int because the resultant
```

```
//answer will not fit in the range of int
```

```
    scanf("%d",&t);
```

```
    while(t--)
```

```
    {
```

```
        scanf("%lld",&n);
```

```
        ans=(n*(n+1))/2;
```

```
        printf("%lld\n",ans);
```

```
    }
```

```
    return 0;
```

```
}
```

```
//Range of Int -2,147,483,648 to 2,147,483,647
```

```
//Range of long long int : approx  $10^{18}$ 
```

Time Complexity: $O(1)$ per testcase and $O(t)$ overall.

Space Complexity: $O(1)$

Question2. Linear Search

Constraints

$t \leq 10$;

$n \leq 10^5$

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

```
int main()
```

```
{
```

```
    int arr[100005],n,t,i;
```

```

scanf("%d",&t);      //No. of Test Cases
scanf("%d",&n); // No. of elements

for(i=0;i<n;i++)
{
    scanf("%d",&arr[i]);
}
while(t--)
{

    scanf("%d",&x); //Element to be searched

    int flag=0;
    for(i=0;i<n;i++)
    {
        if(arr[i]==x) //if element is found set flag and break the loop
        {
            flag=1;
            break;
        }
    }

    if(flag)
    {
        printf("Element Found\n");
    }
    else
    {
        printf("Element not Found\n");
    }

}
return 0;
}

```

Time Complexity $O(t \cdot n)$

Space Complexity $O(n)$

Question3: Given Sorted array, we have to search an element in that array

Constraints

$t \leq 10^5$

$n \leq 10^6$

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

```
int main()
```

```
{
```

```
    int t,n,i,low,mid,high,arr[1000000];
```

```
    scanf("%d",&t);
```

```
    scanf("%d",&n);
```

```
    for(i=0;i<n;i++)
```

```
        scanf("%d",&arr[i]);
```

```
    while(t--)
```

```
    {
```

```
        scanf("%d",&x);
```

```
        low=0;
```

```
        high=n-1;
```

```
        int flag=0;
```

```
        while(low<=high)
```

```
        {
```

```
            mid=low+(high-low)/2;
```

```
            if(arr[mid]==x)
```

```
            {
```

```
                flag=1;
```

```
                break;
```

```
            }
```

```
            else if(arr[mid]>x)
```

```
            {
```

```
                high=mid-1;
```

```
            }
```

```
            else
```

```
                low=mid+1;
```

```
        }
```

```
        if(flag)
```

```
            printf("Element Found\n")
```

```
        else
```

```
            printf("Element Not Found\n");
```

```
    }
```

```
    return 0;
```

```
}
```

Time Complexity: $O(t \cdot \log(n))$

Space Complexity: $O(n)$

Question : Given a string of lower cases only, you have to find the frequency of each character in string .

Constraints

Length of String $\leq 10^6$

String consists of lower cases only

```
#include<stdio.h>
int main()
{
    char str[1000005];
    int i,len,arr[26]={0};

    scanf("%s",str);
    len=strlen(str);
    for(i=0;i<len;i++)
    {
        arr[str[i]-'a']++; // This will store the frequency of each character
    }

    for(i=0;i<26;i++)
    {
        printf("Character %c has occurred %d times\n",'a'+i,arr[i]);
    }
    return 0;
}
```

Note ASCII values: a to z : 97 to 122

A to Z : 65 to 90

0 to 10 (character): 48 to 57

Time Complexity: $O(n)$;

Space Complexity: $O(26)$ extra space complexity as array of size 26 is being declared and $O(n+26)$ overall.