

Question no.: 01

Question name: Multiples

Algorithm:

- i. Input a and b, two numbers.
- ii. Check which number is greater.
- iii. If a is greater than b then find if a is multiple of b.
- iv. If a is multiple of b then print “Multiples”.
- v. If not then print “No Multiples”.
- vi. Repeat previous two steps if b is greater than a.
- vii. Exit.

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    int a,b;
    cin>>a>>b;
    if(a>b) {
        if(a%b==0)
            cout<<"Multiples\n";
        else
            cout<<"No Multiples\n";
    }
    else{
        if(b%a==0)
            cout<<"Multiples\n";
        else
            cout<<"No Multiples\n";
    }
    return 0;
}
```

Analysis: In this question, the task is to find if number a is multiple of b or vice-versa. It was simply solved by checking if $a\%b=0$ or $b\%a=0$. And then “Multiples” or “No Multiples” was printed based on the result of the condition. The time complexity of this problem is $O(1)$.

Acceptance:

215710728	Jul/26/2023 15:55 ^{UTC+6}	sajal2002	1 - Multiples	GNU C++17	Accepted	15 ms	0 KB
---------------------------	------------------------------------	-----------	-------------------------------	-----------	----------	-------	------

Question no.: 02

Question name: Sort Numbers

Algorithm:

- i. Take input of an array a size of 3.
- ii. Copy the array to a new array c, size of 3.
- iii. Sort array a.
- iv. Print array a.
- v. Print array c.
- vi. Exit.

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    int a[3], c[3];
    for (int i=0; i<3; i++) {
        cin>>a[i];
        c[i]=a[i];
    }
    sort(a, a+3);
    for (int i=0; i<3; i++) {
        cout<<a[i]<<endl;
    }
    cout<<endl;
    for (int i=0; i<3; i++) {
        cout<<c[i]<<endl;
    }
    return 0;
}
```

Analysis: The task of this question is to print an array two times. First time as sorted array and second time as it was inputted. So, the array can be copied to another array and sorted one and kept another one as it was. Then both of the array was printed. Each element was printed in a new line. The time complexity of the problem is $O(n)$.

Acceptance:

215711343	Jul/26/2023 15:59 ^{UTC+6}	sajal2002	T - Sort Numbers	GNU C++17	Accepted	15 ms	0 KB
---------------------------	------------------------------------	-----------	----------------------------------	-----------	----------	-------	------

Question no.: 03

Question name: Multiplication Table

Algorithm:

- i. Take input of n.
- ii. Repeat a loop from i=1 to i is equal to 12 by 1.
- iii. Print n and ' * ' after that i, then ' = '.
- iv. Then in the same line print the multiplication of n and i.

v. Exit.

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    int n;
    cin>>n;
    for(int i=1;i<=12;i++){
        cout<<n<<" * "<<i<<" = "<<n*i<<endl;
    }
    return 0;
}
```

Analysis: In this problem, the task is to print a multiplication table of a number n. The problem is solved using a simple loop. The loop was run i=1 to is less or equal to 12. The time complexity of this problem is $O(n)$.

Acceptance:

215712368	Jul/26/2023 16:07 ^{UTC+6}	sajal2002	F - Multiplication table	GNU C++17	Accepted	15 ms	0 KB
---------------------------	------------------------------------	-----------	--	-----------	----------	-------	------

Question no.: 04

Question name: Even, Odd, Positive and Negative

Algorithm:

- i. Set n, the size of array.
- ii. Set a, an array size of n.
- iii. Set i:=0, e:=0, o:=0, p:=0, ne:=0.
- iv. Repeat step iv, v, vi, vii while i is less than n.
- v. If a[i] is divisible by 2 increase e by one.
- vi. If a[i] is not divisible by 2 increase o by one.
- vii. If a[i] is greater than 0 increase p by one.
- viii. If a[i] is less than 0 increase ne by one.
- ix. Print the total number of even, odd, positive and negative number.
- x. Exit.

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    int n,e=0,o=0,p=0,ne=0;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
}
```

```

    }
    for(int i=0;i<n;i++){
        if(a[i]%2==0)
            e++;
        if(a[i]%2!=0)
            o++;
        if(a[i]>0)
            p++;
        if(a[i]<0)
            ne++;
    }
    cout<<"Even: "<<e<<endl;
    cout<<"Odd: "<<o<<endl;
    cout<<"Positive: "<<p<<endl;
    cout<<"Negative: "<<ne<<endl;
    return 0;
}

```

Analysis: In this problem, the task is to find the total number of odd, even, positive and negative number. For this a loop was used. Each of the element was checked if it was odd or even, then if it was positive or negative. In the end the total amount was printed. The time complexity of this problem is $O(n)$.

Acceptance:

215713442	Jul/26/2023 16:16 ^{UTC+6}	sajal2002	C - Even, Odd, Positive and Negative	GNU C++17	Accepted	31 ms	0 KB
---------------------------	------------------------------------	-----------	--	-----------	----------	-------	------

Question no.: 05

Question name: Three numbers

Algorithm:

- i. Set k and s.
- ii. Set d:=0.
- iii. Repeat i=0 to i is less or equal to k by one.
- iv. Repeat j=0 to j is less or equal to k by one
- v. If s-i-j is greater or equal to 0 and less or equal to k then increase d by one.
- vi. Print d.
- vii. Exit.

Code:

```

#include<bits/stdc++.h>
using namespace std;
int main() {
    int k,s,d=0;
    cin>>k>>s;
    for(int i=0;i<=k;i++){
        for(int j=0;j<=k;j++){
            if(s-i-j>=0 && s-i-j<=k){
                d++;
            }
        }
    }
    cout<<d<<endl;
    return 0;
}

```

```

    }
}
cout<<d<<endl;
return 0;
}

```

Analysis: The task of this problem is to find three numbers, whose sum is not greater than s. And also the each number will not be greater than k. To solve the problem a nested loop was used. Then it was checked if s-i-j is between 0 to k, if it is then d was increased by one. And this is the ans.

Acceptance:

217898933	Aug/08/2023 22:36 UTC+6	sajal2002	Z - Three Numbers	GNU C++17	Accepted	31 ms	0 KB
---------------------------	----------------------------	-----------	-----------------------------------	--------------	----------	-------	------

Question no.: 06

Question name: Beautiful Matrix

Algorithm:

- i. Set a 2D array a, size of 5 and 5.
- ii. Repeat i=0 to i is less than 5 by 1.
- iii. Repeat j=0 to j is less than 5 by 1.
- iv. If a[i][j] is equal to 1,
- v. Set x:= i+1.
- vi. Set y:= j+1.
- vii. Print the summation of difference between x and 3, and y and 3.
- viii. Exit.

Code:

```

#include<bits/stdc++.h>
using namespace std;
int main() {
    int a[5][5], x, y;
    for(int i=0; i<5; i++) {
        for(int j=0; j<5; j++) {
            cin>>a[i][j];
        }
    }
    for(int i=0; i<5; i++) {
        for(int j=0; j<5; j++) {
            if(a[i][j]==1) {
                x=i+1;
                y=j+1;
            }
        }
    }
    cout<<abs(x-3)+abs(y-3)<<endl;
    return 0;
}

```

```
}
```

Analysis: In this problem, any algorithm can be implemented. In order to solve this, the location of 1 was determined. Then, just the difference of current row and column number and row and column number 3 was calculated and the sum of the difference was printed. Time complexity of this problem is $O(n^2)$.

Acceptance:

215716139	Jul/26/2023 16:35 ^{UTC+6}	sajal2002	A - Beautiful Matrix	GNU C++17	Accepted	30 ms	0 KB
---------------------------	------------------------------------	-----------	--------------------------------------	-----------	----------	-------	------

Question no.: 07

Question name: One Prime

Algorithm:

- i. Set n, the number.
- ii. Set d:=0.
- iii. Repeat i=2 to i is less than n by 1.
- iv. If n is a multiple of i increase d by one and break the loop.
- v. If d is equal to zero print “YES”.
- vi. Else print “NO”.

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    int n,d=0;
    cin>>n;
    for(int i=2;i<n;i++) {
        if(n%i==0) {
            d++;
            break;
        }
    }
    if(d==0)
        cout<<"YES\n";
    else
        cout<<"NO\n";
    return 0;
}
```

Analysis: Prime number is a number which is not divisible by any number except 1 and itself. So, if a number is found which is greater than 1 and a multiplication of n, then n is not a prime number. Else n is a prime number. This logic was used in a loop to check if n is prime or not. Then the output “YES” and “NO” was printed. Time complexity is $O(n)$.

Acceptance:

215716502	Jul/26/2023 16:37 ^{UTC+6}	sajal2002	H - One Prime	GNU C++17	Accepted	15 ms	0 KB
---------------------------	------------------------------------	-----------	-------------------------------	-----------	----------	-------	------

Question no.: 08

Question name: Increasing Array

Algorithm:

- i. Set n, the size of array.
- ii. Set an array a.
- iii. Set d:=0.
- iv. Repeat i=0 to i<n by one,
- v. If a[i] is greater than a[i+1],
- vi. Set d:=d+a[i]-a[i+1].
- vii. Set a[i+1]=a[i].
- viii. Print d.
- ix. Exit.

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    long long int n,d=0;
    cin>>n;
    long long int a[n];
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    for(int i=0;i<n-1;i++){
        if(a[i]>a[i+1]){
            d=d+a[i]-a[i+1];
            a[i+1]=a[i];
        }
    }
    cout<<d<<endl;
    return 0;
}
```

Analysis: The task of the question is to make every element make as large as the previous element, and calculate the steps to do that. For that, it was checked if an element is larger than it's next one, if it was then the difference between two elements was calculated and was added to d. Then the elements were interchange. The time complexity of this problem is O(n).

Acceptance:

Task:	Increasing Array
Sender:	Msajal2002
Submission time:	2023-07-26 16:57:24 +0300
Language:	C++
Status:	READY
Result:	ACCEPTED

Question no.: 09

Question name: Missing Number

Algorithm:

- i. Set n.
- ii. Set an array size of n-1.
- iii. Sort the array.
- iv. Repeat i=0 to i<n by one.
- v. If $i+1 \neq a[i]$ print i+1 and break the loop.
- vi. Exit.

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    int n;
    cin>>n;
    int a[n-1];
    for(int i=0;i<n-1;i++){
        cin>>a[i];
    }
    sort(a,a+(n-1));
    for(int i=0;i<n;i++){
        if(i+1!=a[i]){
            cout<<i+1<<endl;
            break;
        }
    }
    return 0;
}
```

Analysis: The goal of this problem is to find a missing number in a array. The size of array is n. And the elements are 1 to n, but one element is missing. In order to find it, first the array was sorted and then in a loop it was checked which number was missing by using condition. If $i+1$ is not equal to $a[i]$ the $i+1$ was missing. Time complexity is $O(n)$.

Acceptance:

Task:	Missing Number
Sender:	Msajal2002
Submission time:	2023-08-07 14:09:41 +0300
Language:	C++11
Status:	READY
Result:	ACCEPTED

Question no.: 10

Question name: Magic Elements

Algorithm:

- i. Set t.
- ii. Repeat a loop while t is not equal to 0.
- iii. Set n, k.
- iv. Set d:=0,s:=0.
- v. Set an array a size of n.
- vi. Calculate the sum of every element of array and set it to s.
- vii. Repeat i=0 to i is less than n by one.
- viii. Set x:=k+a[i]
- ix. If x is greater than s-a[i] increase d by one.
- x. Print d.
- xi. Exit.

Code:

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

using namespace std;

int main() {

    int t;

    cin>>t;

    while(t--){

        int n,k,d=0,s=0;

        cin>>n>>k;

        int a[n];

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

            cin>>a[i];
```

```

        s=s+a[i];
    }
    for(int i=0;i<n;i++){
        int x=k+a[i];
        if(x>(s-a[i])){
            d++;
        }
    }
    cout<<d<<endl;
}
return 0;
}

```

Analysis:

Acceptance:

1012153593	✓	0.02s	3M	C++17	5 days ago	→
------------	---	-------	----	-------	------------	---

Question no.: 12

Question name: Word Capitalization

Algorithm:

- i. Set a string s.
- ii. Check if s[0]>=97 or not.
- iii. If it is set s[0]:=s[0]-32.
- iv. Print s.
- v. Exit.

Code:

```

#include<bits/stdc++.h>
using namespace std;
int main() {
    string s;
    cin>>s;
    if(s[0]>=97){
        s[0]=s[0]-32;
    }
    cout<<s<<endl;
    return 0;
}

```

Analysis: In this problem, the task is to make first character of string s capital letter if it is in smaller letter. For this it was checked if the character is in smaller letter or not by using ASCII number of alphabet. If the character is greater than 96 then it was in smaller letter. To get capital letter 32 was subtracted from the ASCII value of the character. And the answer was found. If the letter was in capital letter no operation needs to be done. Time complexity is O(n).

Acceptance:

215717449	Jul/26/2023 16:45 ^{UTC+6}	sajal2002	A - Word Capitalization	GNU C++17	Accepted	30 ms	0 KB
---------------------------	------------------------------------	-----------	---	-----------	----------	-------	------

Question no.: 13

Question name: Way Too Long words

Algorithm:

- i. Set t, the total test case.
- ii. Repeat while t is greater than 0.
- iii. Set string s.
- iv. Set l, size of string s.
- v. Check if l is greater than 10 or not.
- vi. If l is greater than 10, print s[0], then l-2 and in the end s[l-1].
- vii. If l is less than 10, print the string.
- viii. Exit.

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    int t;
    cin>>t;
    while(t--){
        string s;
        cin>>s;
        int l=s.size();
        if(l>10){
            cout<<s[0]<<l-2<<s[l-1]<<endl;
        }
        else{
            cout<<s<<endl;
        }
    }
    return 0;
}
```

Analysis: In this problem the main concept is to determine the size of a string. For this a built in function, size() is used. If the size is greater than 10, then the first character and the last

character was printed and in the middle of these two character 1-2 was printed. If the size is less or equal to 10 just the string was printed.

Acceptance:

217847898	Aug/08/2023 16:30 ^{UTC+6}	sajal2002	71A - Way Too Long Words	GNU C++17	Accepted	15 ms	0 KB
---------------------------	------------------------------------	---------------------------	--	-----------	----------	-------	------

Question no.: 14

Question name: Float or Int

Algorithm:

- i. Set string s.
- ii. Set l, the size of string.
- iii. Set d:=0.
- iv. Repeat i=0 to i is less than l by 1.
- v. If s[i] is equal to '.', set k:=i.
- vi. Repeat j=i+1 to j is less than l by 1.
- vii. If s[j] is not equal to '0', increase d by one.
- viii. If d is equal to 0,
- ix. Print "int" and every character which index is less than k.
- x. If d is not equal to 0,
- xi. Print "float".
- xii. Print every character which index is less than k, and after a space print "0." and rest of the element.
- xiii. Exit.

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    string s;
    cin>>s;
    int l=s.size(),d=0,k;
    for(int i=0;i<l;i++){
        if(s[i]=='.'){
            k=i;
            for(int j=i+1;j<l;j++){
                if(s[j]!='0'){
                    d++;
                }
            }
        }
    }
    if(d==0){
        cout<<"int ";
        for(int i=0;i<k;i++){
            cout<<s[i];
        }
    }
```

```

        cout<<endl;
    }
    else{
        cout<<"float ";
        for(int i=0;i<k;i++){
            cout<<s[i];
        }
        cout<<' '<<'0';
        for(int i=k;i<l;i++){
            cout<<s[i];
        }
        cout<<endl;
    }
    return 0;
}

```

Analysis: In order to solve this problem, first it was determined if there are any numbers after '.', if there are not then the number is an integer type number, else floating type number. Time complexity of this problem is $O(n^2)$.

Acceptance:

217022717	Aug/03/2023 23:06 ^{UTC+6}	sajal2002	U - Float or int	GNU C++17	Accepted	15 ms	0 KB
---------------------------	------------------------------------	-----------	----------------------------------	-----------	----------	-------	------

Question no.: 15

Question name: Petya and String

Algorithm:

- i. Set two string s and c.
- ii. Convert every character of both string lowercase.
- iii. Set sn:=0 and cn:=0.
- iv. Repeat i=0 to i is less than the size of string s by one.
 - v. If s[i] is greater than c[i], increase sn by one and break.
 - vi. If s[i] is less than c[i], increase cn by one and break.
- vii. If sn is greater than cn, print 1.
- viii. If sn is less than cn, print -1.
- ix. If both are same print 0.
- x. Exit.

Code:

```

#include<bits/stdc++.h>
using namespace std;
int main() {
    string s,c;
    cin>>s>>c;
    for(int i=0;i<s.size();i++){
        s[i]=tolower(s[i]);
    }
    for(int i=0;i<c.size();i++){
        c[i]=tolower(c[i]);
    }
    int sn=0,cn=0;
    for(int i=0;i<s.size();i++){
        if(s[i]>c[i]) sn++;
        if(s[i]<c[i]) cn++;
        if(sn>cn) break;
        if(cn>sn) break;
    }
    if(sn>cn) cout<<1<<endl;
    else if(cn>sn) cout<<-1<<endl;
    else cout<<0<<endl;
    return 0;
}

```

```

        c[i]=tolower(c[i]);
    }
    int sn=0,cn=0;
    for(int i=0;i<s.size();i++){
        if(s[i]>c[i]){
            sn++;
            break;
        }
        if(c[i]>s[i]){
            cn++;
            break;
        }
    }
    if(sn>cn){
        cout<<1<<endl;
    }
    else if(cn>sn){
        cout<<-1<<endl;
    }
    else{
        cout<<0<<endl;
    }
    return 0;
}

```

Analysis: In this problem, which of the two string is lexicographically larger than other has to be determined. For this, it was check that if the lowest index character's ASCII number was bigger. The string with bigger character is larger than other one. Time complexity of this problem is $O(n)$.

Acceptance:

217853525	Aug/08/2023 17:29 ^{UTC+6}	sajal2002	112A - Petya and Strings	GNU C++17	Accepted	30 ms	0 KB
---------------------------	------------------------------------	-----------	--	-----------	----------	-------	------