

C++

Introduction

C++ is an object-oriented programming language which was developed by **Bjarne Stroustrup** at Bell Labs in 1979. It was developed as an extension to C programming language. It is a cross platform language which allows developers to create high performing application. It gives programmers a high level of control of system resources and memory.

This language has three major updates in 2011, 2014, 2017 which are C++ 11, C++ 14, C++ 17.

C++ is one of the fastest languages in the world so it is quite famous with competitive programmers since in it time limits are there and fast compilation of C++ makes it a better choice for usage. C++ also has a vast library of pre-defined functions which helps programmers in many ways. So, it makes C++ an ideal choice for coders.

Benefits

- One of the fastest languages for coding.
- Has a vast library of pre-defined functions to help coders.
- Is not difficult to learn.
- High level of usages as it can build high performing applications.

Basic Structure/Syntax

In it we will be learning the basic structure on how to write C++ programs. Hope, you read the Basic Knowledge track provided before starting on with its structure to understand it better. We will not be repeating those again as those are common for all languages. So, if not checked out till now, go first check out the basic track and then start with C++.

```
1. #include<bits/stdc++.h>
2. int main()
3. {
4.     int a,b;
5.     cout<<"Enter two numbers \n";
6.     cin>>a>>b;
7.     cout<<"\n The sum of the two numbers are"<<a+b;
8.     return 0;
9. }
10.
11. /* OUTPUT
12. Enter two numbers
13. 23 45
14. The sum of the two numbers are 68
15. */
```

In the above code, you can see the basic structure of a C++ program which basically takes two integers from user and display their sum.

I will now explain each and every line of the code for your understanding.

The first line of the code is called the header files. It tells the compiler which libraries to include for using in the program. There are many header files available and you will learn about them later. For competitive coding or for any programmer, best choice is **#include<bits/stdc++.h>** as it contains all the necessary library functions.

The next line states the start of main function which is an important part of any C++ program. No C++ program goes without it. All other functions which you define (you will learn how to create functions in basic knowledge track) are called from main function.

The next line basically is defining of two integers a and b. Then next, we display the line to user for telling him/her to give two integers.

Then next line we take two integers which user provides. Then next we display the sum of integers with a message to clarify.

Then we use return keyword and close the braces to complete the program.

Now, I know many of you might have doubts in the above code, well let me clarify them now.

cin – This keyword is used to take any input from the user. It is accompanied by >> symbol. cin is basically a mixture of c and in which means to take input.

cout – This keyword is used to display output on screen. Anything we want to print on screen we put it in cout keyword followed by << symbol. The lines or sentence we want to display as it is on screen, we put them inside double quotation marks (e.g.: cout<<“The sum is”);). The result we want to print goes without quotation marks (e.g.: cout<<a+b;). For printing multiple outputs with same cout we use << symbol that many times between outputs to be displayed.

e.g.: cout<<“The sum is \t”<<a+b;

return – This is basically used to describe the return type of the main function. Since we are using int main() thus our return value is 0.

Header Files

While writing a program in C++ we need to include some header files at beginning of program. It allows you to include or call various library functions to be used in our program. So, we will discuss about many such header files and what function each of those header files serve.

There are two types of header files: -

1. **Standard Library Header files** – These are the standard libraries which are already available for use in C++ compiler. #include is used to include them in program.

2. **User Defined Header files** – These are the libraries which users define for their convenience. `#define` is used to create them.

Syntax:

The name of header file is always enclosed within angular brackets (<>) or double quotation marks (""), though we prefer the former method. In C++ it is not important for the header files to end with .h extension. While in C it is compulsory to be writing .h extension each time you define a header file.

e.g.: `#include<iostream.h>` or `#include<iostream>` or `#include"iostream"` or `#include"iostream.h"`

While defining user defined header files, the format is a little different as you need to start with `#define` and then write what you need to define. It usually makes tasks easier for a coder while writing the program.

e.g.: `#define ll long long int`

In the above example we are defining `ll` to be used in place of writing `long long int` in our program to make it easier and shorter while writing code as we need to write `ll` only and it makes the compiler understand that we are referring to `long long int`.

Defining a header file

When you need to define a header file on our own to be used in any program you write in C++, you need to create a new file, write your code for header in C++ program, and save it with extension .h as it's a header file.

A sample header file is here to show how to define a header is: -

```
1. int factorial(int number)
2. {
3.     int iteration, factorial=1;
4.     for(iteration=1; iteration<=number; iteration++)
5.     {
6.         factorial=factorial*iteration;
7.     }
8.     return factorial;
9. }
```

Then you will be able to use this header file by using `include<name of header>` and then it can be used in the code.

Various header files and their functions

1. **`#include<stdio.h>`** - It is the standard input – output header file. It handles input output operations such as `cin` and `cout`.

2. **#include<conio.h>** - It is used the console input output header file which is used to perform operations such as clrscr() which is used to clear the screen or getch() to get a character from keyboard.
3. **#include<math.h>** - It is used to use Math related functions in our program such as sqrt() which finds out square root or like pow() which calculates power of numbers.
4. **#include<string.h>** - This header file is used to use all string related functions like strlen() to get length of string and others.
5. **#include<errno.h>** - It is used to handle error related operations such as errno().

There are many other header files for our use.

Pro-tip

As a competitive coder, I would recommend everyone to use **#include<bits/stdc++.h>** header file because it contains all necessary functions and libraries so we don't need to define multiple header files which makes our work easier.

Hope you got how to use header files in any C++ program, well then we can move to main part of program.

Before moving forward, I would recommend to go through Basic Knowledge track to get some knowledge of functions, pointers and other necessary stuff. If you have read that, its great and get ready to move forward. Hope you are enjoying the learning journey! Now, lets move forward to learn how to take different types of input from the user.

Taking Input

Now we will learn how to take input from user. For this you need to be thorough with the data types given in Basic Knowledge section.

So, to take any type of input we first need to declare which data type are we using and then declare a variable name.

e.g.: int a, b, sum;

In the above example int is the data type and a, b, sum are the variable names. Variable names should not be keywords.

Then to ask user for input we need to use cin keyword.

e.g.: cin>>a>>b;

Here in the above example, cin tells the system to take input from the user and then >> is used then we put the name of variable in which we need to store the input which user gives. Every variable is separated by >>.

Few examples are: -

1. string s;
2. double rt;
3. char c;
4. bool e;
5. int a[8];
6. vector<int>a;

There are many more such examples. The 5th one shows input of an array while 6th one denotes a vector input (you will read about vectors later).

Giving Output

It is quite an easy task to print or give output on screen. We use cout keyword followed by << symbol and then variables whose value are to be printed or if we need to print some sentence or words as it is then we write them inside double quotation marks.

Each variable is separated by << symbol.

e.g.: `cout<<"\n The sum is\t"<<sum<<endl;;`

In the above example the first to come after << is a printable statement. \n in the statement denotes starting in a new line then it displays the sentence then \t to get some spaces. After that we put << symbol so that we can put next variable and then we display the value of sum there. After that we use endl keyword to get our cursor on a new line.

endl – This statement is used to take our cursor to a new line after the line is displayed. It comes at last part of cout. Its not a necessary keyword as \n basically does the same task. endl marks the end of a line.

Arrays in C++ (applicable for all languages)

Arrays are collection of items which are stored at contiguous memory positions and elements can be accessed randomly using indices of array. Arrays can store both primitive data types and composite data types. Arrays can even store arrays!!

Arrays are like boxes in which data is filled.

67	12	56	29	58	78	90	11
----	----	----	----	----	----	----	----

Above is an example of an integer array of 8 elements. The first element is at index 0 and so last one has an index of 7.

Note: Generally, arrays start from 0 index unless its mentioned that its 1 indexed array.

Declaration:

Data type array name[size];

e.g.:

int ser[n];

Then we take input by using loop as in the same way as in vector (not using push_back things) but by using a[i] type of input. a[i] means to take input in ith variable.

Vectors in C++

Vectors are a very important tool in C++. It is a better version of arrays, to say at least.

Vectors serve the same purpose as what an array does but has several advantages in memory management. If few don't know the exact number of inputs the array needs to take from user, we need to define the maximum possible boxes (positions) in the array so that it does not overflow. But if user does not give those number of inputs the empty boxes are not destroyed but are wasting the memory.

While on other hand, in vectors you don't need to specify how many boxes or positions you need. The vector just creates a position when it gets input so it saves a lot of space/memory. Vectors are quite easy to use and so is a better choice to be used.

Benefits of vectors over array

- Size of vector need not be defined. It changes dynamically.
- Elements can be deleted from a vector without wasting the position as when we delete an element the next one takes its space and reduces the size of vector while in array it just leaves a blank space.
- Multiple objects can be stored in a vector.
- Size of vector can be determined in $O(1)$ time while size of a dynamic allocated array can't be determined.
- While passing array to a function we need to pass an additional variable too to represent size of array while there is no need to do so in vector as it maintains variable which keeps track of its size.
- Arrays cannot be returned from a function unless dynamically allocated while vectors can easily be returned from a function.

Format of vector

A vector is defined in the following way: -

vector <data type> variable name;

e.g.: vector <int> scoder;

The above example shows the creation of a vector named scoder which is in integer data type.

e.g.:

```
vector<int> scoder;  
for(int i=0; i<10; i++)  
{  
    cin>>scoder[i];  
}
```

Here in above example we have shown how to take input in vector. Its just the same like array.

There is another way to take input in vectors like `scoder.push_back()`.

There are many such functions to use in C++. Let's have a look at some of those used for vectors.

1. `scoder.push_back()` – It takes the value and stores it in the vector. The value stored goes at end of vector.
2. `scoder.push_front()` – It takes the value and store it in front position in vector.
3. `scoder.begin()` – It returns an iterator pointing to first element of vector.
4. `scoder.end()` – It returns an iterator pointing to last element of vector.
5. `scoder.size()` – It returns the size of vector.
6. `scoder.empty()` – Returns whether container is empty or not.
7. `scoder.at(i)` – Returns the reference to ith vector element.
8. `scoder.pop_back()` – It removes the last element of vector.
9. `scoder.clear()` – It clears all elements of vector.
10. `scoder.erase()` – It removes elements from vectors from a specified position to another specified position. E.g.: `scoder.erase(i,j)` – erases elements from ith position to jth position.

Pre-defined functions for use

There are various pre-defined functions or library functions which have already been defined for a specific purpose when used in the program.

We will be seeing few of them now.

1. `Sqrt()` – This function is used to find out square root of any number written within () after `sqrt`.
2. `Pow()` – This function is used to calculate any power of any number . e.g.: `pow(2,4)=2^4=16`.

3. Strlen() – This function is used to get the length of string.
4. Size() – This function returns the size of any string.
5. Substr() – It is used to get substrings from a string. E.g.: substr(5,9) means taking the substring from position 5 to 8.

There are many more such functions and you will get to know more and more when you practice questions.

So, we have prepared for you a great set of questions to practice. No, we won't be giving you very simple questions like adding numbers and printing your name but we would take a more practical approach to things as I present to you our great set of questions prepared by me. These questions try to make you feel motivated and its fun solving them as they are real life based. So, lets jump in the problems now. For very simple beginner questions, visit the C track and solve them using C++ now.

Practice Problems (For Algo & DS practice too)

1) Ansh was going crazy while writing Python section. He wanted to set an unusual question as the starter one, so he proposed the idea to Shubham doing C++. They together made a new table question which was weird and crazy. They formulated something crazy like a multiplication table but was actually named ASA table. In that you will be given a number and your task is to print its ASA table.

Format of ASA table – number # digit (being ASAed) = number + digit² (# being ASA symbol)

e.g.:

$$1 \# 1 = 1 + 1^2 = 2$$

$$1 \# 2 = 1 + 2^2 = 5$$

And so on till 10th term.

Solution:

```

1. #include<bits/stdc++.h>
2. int main()
3. {
4.     int a;
5.     cin>>a;
6.     for(int i=1;i<10;i++)
7.     {
8.         cout<<"\n"<<a<<" # "<<i<<" = "<<(a+i*i);
9.     }
10. }
11.
12. /* OUTPUT if a=3
13. 3 # 1 = 4
14. 3 # 2 = 7
15. 3 # 3 = 12
16. ...
17. */

```


2) Shubham wanted to make you practice a good pattern program which makes all your concepts clear so he is asking this problem to print the required pattern:

```
1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

Solution:

```
1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5.     int rows, count = 0, count1 = 0, k = 0;
6.     cout << "Enter number of rows: ";
7.     cin >> rows;
8.     for(int i = 1; i <= rows; ++i)
9.     {
10.        for(int space = 1; space <= rows-i; ++space)
11.        {
12.            cout << " ";
13.            ++count;
14.        }
15.        while(k != 2*i-1)
16.        {
17.            if (count <= rows-1)
18.            {
19.                cout << i+k << " ";
20.                ++count;
21.            }
22.            else
23.            {
24.                ++count1;
25.                cout << i+k-2*count1 << " ";
26.            }
27.            ++k;
28.        }
29.        count1 = count = k = 0;
30.        cout << endl;
31.    }
32.    return 0;
33. }
```

3) Anmol was feeling tired while preparing the Java content for you all. So, he devises a strange resting system for himself. He will rest only when certain conditions are met. For that purpose, he takes any sentence from Shubham and states that if number of spaces in the sentence is even then only, he takes rest. So, your task is to write code on behalf of Anmol as he is already tired and help him decide. Take the sentence and write the code to figure out whether he should rest or not.

Note: There are no extra spaces. Only 1 space is present between 2 words.

e.g.:

Input: Vidya Bhushan is going to inter IIT contest in Aeromodelling.

Output: No rest

Explanation: Since there are 9 spaces, it's an odd number so Anmol won't rest.

Solution:

```
1. #include<bits/stdc++.h>
2. int main()
3. {
4.     string s;
5.     int c=0;
6.     cin>>s;
7.     for(int i=0;i<s.length();i++)
8.     {
9.         if(s[i]==' ')
10.            c++;
11.     }
12.     cout<<c;
13.     return 0;
14. }
```

4) Kunal hates waking up in the morning for classes and many a times misses them too! So, he then comes up with idea to devise a plan to check whether he will be going to class in morning or not. His idea was crazy and he came to Shubham for help to help him calculate. As Shubham was busy preparing content for all of you, he told to ask it you. So, your task is now to help Kunal to decide whether to go to class or not.

Kunal will take any number and if that number is a palindrome, then he only he will be going to classes. So, your task is to get whether Kunal will go to class or not.

Palindrome numbers – The numbers which are same from both ends. E.g.: 48384 which when read backwards reads the same.

Solution:

```
1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5.     int n, num, digit, rev = 0;
6.     cout << "Enter a positive number: ";
7.     cin >> num;
8.     n = num;
9.     do
10.    {
11.        digit = num % 10;
12.        rev = (rev * 10) + digit;
13.        num = num / 10;
14.    } while (num != 0);
15.    cout << " The reverse of the number is: " << rev << endl;
16.    if (n == rev)
```

```

17.         cout << " The number is a palindrome.";
18.     else
19.         cout << " The number is not a palindrome.";
20.     return 0;
21. }

```

5) Satyam loves to play PUBG but cannot play it much since he needs to study too! So, he devises a plan to play only that many numbers of matches which is the output after running a program. The program will take an array of integers as input and then will calculate the number of prime numbers in the array, **p** and give the output as **p%5** (the remainder obtained when p is divided by 5). So, Satyam would be able to play that many matches which is the output of program.

e.g.:

Input: {4, 5, 7, 1, 2, 9, 8, 11, 14, 13, 10, 17, 16, 19, 23}

Output: 3

Explanation: Since there are 8 prime numbers, $8\%5=3$ so 3 is output. So, Satyam will play 3 matches.

Solution:

```

1. #include <bits/stdc++.h>
2. bool isPrime(int num)
3. {
4.     int a=0;
5.     for(int t=1;t<=num;t++)
6.     {
7.         if(num%t==0)
8.             a++;
9.     }
10.    if(a==2)
11.        return true;
12.    else
13.        return false;
14. }
15. int main()
16. {
17.     int n,cp=0;
18.     cin>>n;
19.     int arr[n];
20.     for(int i=0;i<n;i++)
21.         cin>>arr[i];
22.     for(int j=0;j<n;j++)
23.     {
24.         if(isPrime(arr[j])==true)
25.             cp++;
26.     }
27.     cout<<(cp%5);
28. }

```

6) Pulkit has gone to buy cotton candies. After seeing lots of rates and varieties of cotton candies in the store, he is confused (yet again!!). He has **R** rupees to spend which he wants to spend as much as possible. He has to buy 3 cotton candies. The shopkeeper gives him a list of cotton candies with their prices. So, your task is to find out how much maximum Pulkit will be able to spend on buying 3 cotton candies. You will be provided with the number of candies available, maximum budget **R** and an array of integer with rates of different cotton candies. You have to print a single integer denoting the maximum he can spend.

e.g.:

Input: N=7, R=25, {8, 5, 12, 6, 7, 11}

Output: 25

Explanation: Since Pulkit had 25 rupees to buy 3 cotton candies and looking at cost of candies, we see that he is able to spend full 25 rupees for the three cotton candies ($8 + 5 + 12 = 25$ or $8 + 6 + 11 = 25$)

7) Nikhil wants to play free fire. But as he was getting low number of Booyahs he wanted to play Clash Squad to win but to improve his rank he must play Ranked game too! So, finally he thought of some plan and thought that the number of clash squad and ranked games will be decided by a program. So, as he needs programming, he went to Shubham and asked him to write the code for him. Your task is to write the code. You will get a string as input and your task is to find the length of longest palindromic substring, **ls**, then you will output **ls%5** as your answer which would denote the number of games out of 5, he would play ranked games, rest would be clash squads.

e.g.:

Input: Wronglevelgood

Output: Ranked Games – 2, Clash Squads – 3

Explanation: Since “glevelg” was longest palindrome substring, its length is 7 so number of ranked games would be $7\%5=2$. Thus 3 clash squads would be there as total number of games allowed is 5.

Solution:

```
1. // A C++ dynamic programming
2. #include <bits/stdc++.h>
3. using namespace std;
4. // Function to print a substring str[low..high]
5. void SubStr( string str, int low, int high )
6. {
7.     for( int i = low; i <= high; ++i )
8.         cout << str[i];
9. }
10. // It also returns the length of the longest palindrome
11. int longestPalSubstr(string str)
12. {
```

```

13. // get length of input string
14. int n = str.size();
15. // table[i][j] will be false if substring str[i..j]
16. // is not palindrome.
17. // Else table[i][j] will be true
18. bool table[n][n];
19. memset(table, 0, sizeof(table));
20. // All substrings of length 1 are palindromes
21. int maxLength = 1;
22. for (int i = 0; i < n; ++i)
23.     table[i][i] = true;
24. // check for sub-string of length 2.
25. int start = 0;
26. for (int i = 0; i < n-1; ++i)
27. {
28.     if (str[i] == str[i+1])
29.     {
30.         table[i][i+1] = true;
31.         start = i;
32.         maxLength = 2;
33.     }
34. }
35. // Check for lengths greater than 2. k is length
36. // of substring
37. for (int k = 3; k <= n; ++k)
38. {
39.     // Fix the starting index
40.     for (int i = 0; i < n-k+1; ++i)
41.     {
42.         // Get the ending index of substring from
43.         // starting index i and length k
44.         int j = i + k - 1;
45.         // checking for sub-string from ith index to
46.         // jth index iff str[i+1] to str[j-1] is a
47.         // palindrome
48.         if (table[i+1][j-1] && str[i] == str[j])
49.         {
50.             table[i][j] = true;
51.             if (k > maxLength)
52.             {
53.                 start = i;
54.                 maxLength = k;
55.             }
56.         }
57.     }
58. }
59. SubStr( str, start, start + maxLength - 1 );
60. // return length of LPS
61. return maxLength;
62. }
63. // main function
64. int main()
65. {
66.     string str;
67.     cin>>str;
68.     cout <<(longestPalSubstr(str)%5);
69.     return 0;
70. }

```

8) Arnab wants to meet more people of his community. So, he devises a test for those people to test them that they actually belong to their community. As usual, the test was

strange, to say the least. The test is that people give him two strings and he ask whether the strings are anagrams or not. If they are anagrams, Arnab accepts them as their friend and if they are not anagrams, then he does not consider them of same community.

Anagrams – words or sentence which contains same characters but not in same order.

e.g.: rose and sore, rtou and urto etc. are anagrams.

There are no spaces in between words in the string.

Solution:

```
1. #include<bits/stdc++.h>
2. bool areAnagram(char* str1, char* str2)
3. {
4.     // Create a count array and initialize all values as 0
5.     int count[NO_OF_CHARS] = { 0 };
6.     int i;
7.     // For each character in input strings, increment count in
8.     // the corresponding count array
9.     for (i = 0; str1[i] && str2[i]; i++) {
10.         count[str1[i]]++;
11.         count[str2[i]]--;
12.     }
13.     // If both strings are of different length. Removing this condition
14.     // will make the program fail for strings like "aaca" and "aca"
15.     if (str1[i] || str2[i])
16.         return false;
17.     // See if there is any non-zero value in count array
18.     for (i = 0; i < NO_OF_CHARS; i++)
19.         if (count[i])
20.             return false;
21.     return true;
22. }
23. main()
24. {
25.     string st1,st2;
26.     cin>>st1>>st2;
27.     if(areAnagram(st1, st2)==true)
28.         cout<<"Accepted in Community!";
29.     else
30.         cout<<"Not accepted in community";
31. }
```

9) Kartikaya and Divyanshu were both given a string each **A** and **B**. Then Divyanshu told Kartikaya to make his string equal to his by any means so that they have same strings (strange demands!!). So Kartikaya is asking for your help in this question. Your task is to find out minimum steps required to convert Kartikaya's string to Divyanshu's string.

Allowed operations are: - Insertion, replacement or removal of a character in each step

e.g.:

Input: CodeDaily and Coduily

Output: 3

Explanation: Three steps are required. First replace e with u and then remove D and a to make them same.

10) Code Daily is conducting an interview for hiring more members to their team. Shubham is leading the interview process. So, as you are there for interview, you have been asked a programming question. You have been given an array of integers and your task is to find a product array such that $\text{productarray}[i] = \text{product of all elements of array except element at } i\text{th position}$.

Note: Division approach is not preferred

e.g.:

Input: {4, 6, 2, 5}

Output: 60 40 120 48

Explanation: 1st term: $6 * 2 * 5 = 60$

2nd term: $4 * 2 * 5 = 40$ and so on.

11) Krrish has gone to New Vision Express for getting a new sunglass as he broke his last one. He has been given the whole list of glasses available for sale. All the model numbers are provided in an array to you. Now, Krrish was unable to make a choice as all were good. So, Krrish decided to use programming to decide. He took the array of model numbers and decided he would be choosing the model number which is a local maximum. If there are more than one local maximum, then print any one of those.

e.g.:

Input: {5, 7, 2, 8, 9}

Output: 7

Explanation: 7 is greater than both its neighbors which are 5 and 2 so it is a local maximum.

12) Sarvesh is a meme lover. Tushar is a meme creator. One day, they both met, and as usual Sarvesh demanded Tushar to create some great memes for him to enjoy. Tushar has his own strange plans. He decided to make memes on a condition that the maximum number of memes he can make is four and the number of memes will be decided by a program. He asks Sarvesh to give him an array of numbers and also a target integer. Then your task is to find out the maximum length of consequent subsequence in array whose sum

equals to the target number. If no such subarray is possible, print 0. If such an array is possible then print the remainder when length of array is divided by 5 ($\text{length} \% 5$).

e.g.:

Input: {5, 4, 1, 7, 8} and 12

Output: 3

Explanation: The subarray is 4, 1, 7 whose sum is 12 and its length is 3.

13) Saif, Piyush, Ayush (Zic Zelly!) and Shubham (me) are all discussing together how many songs should Ayush a.k.a. Zic Zelly release each month on YouTube. So, as always, strange ideas came to mind. So, Piyush suggested that we do it via a program. Saif then told let's do it by Fibonacci series. So, at last we all came at conclusion that the number of songs will be equal to the last digit of sum of N Fibonacci numbers where N will be provided by Zic Zelly himself. You are given the number of digits N in Fibonacci series and your task is to determine last digit of sum of Fibonacci series up to N terms.

e.g.:

Input: 6

Output: 2

Explanation: Series is 0, 1, 1, 2, 3, 5 so sum= 12 and last digit of sum is 2.

14) Its winter time and its chilling cold! Hence, Vidya Bhushan is fearing to bath. But bathing is important so he decides that he will bath after a specified number of days, which will be calculated by a program. The program would accept an array of integers and your task is to calculate the minimum number of swaps needed to sort the array, **mswaps**. Then you need to print the value of **mswaps%7**. This number would give the number of days after which Vidya Bhushan baths.

e.g.:

Input: 7 5 4 1

Output: 2

Explanation: We require two swaps. Swap 1 with 7 and then swap 5 with 4. This would sort the array.

15) Satyam, Mayank and Priyanshu plans a cricket game on Sunday. They have invited many friends to play with them on Sunday. Everyone has accepted the offer but all will actually not come. Satyam wants to know how many of their friends would actually be coming to play on Sunday. This cannot be guessed, right! But they devised a funny and weird way to calculate it. They gave all of their friends an integral ID and put all those integers in an array. Then they want you to write a code for them so that they can calculate how many would come to play. That would be decided by some conditions. The numbers which are divisible by the product of numbers which precede and succeed it will be the one who are **not** coming for play. For 1st and last digit of array only consider the number which is available and consider its product with 1. Return the array of integers who are coming to play.

e.g.:

Input: {3, 7, 4, 8, 6, 7}

Output: {3, 7, 6, 4}

Explanation: 4 and 8 are removed because they are divisible by product of preceding and succeeding numbers. 4 is divisible by $7*8$ and 8 is divisible by $4*6$ so they are removed.

For more practice, visit the Practice section. There you will find very good questions for your self-practice.

This app provides you with great coding questions with solutions daily.

Hope you enjoyed learning!

If you have any doubts regarding any topic, do contact us!

Rate our App on Play Store - <https://play.google.com/store/apps/details?id=com.skcoding.lrnprog>

