

C++ - a programming language

Compilers

- Computers understand only 0101010.... (machine or binary code)
- We write code in a programming language
- So, we need to use a **compiler** to translate our programming language code to the 010101.. (binary or machine code) form
- Compiler also checks for errors and tells you, if there is any error, along with the line number, in which the error is present

Which compiler can we use?

- Geany IDE or CodeBlocks
- or Online IDEs
 1. <https://csacademy.com/workspace/>
 2. <https://ideone.com/>
- or Mobile App (Search on Playstore for C++ compiler)
(Prefer mobile app, only when you can't access a laptop or PC)
- **For setting up compiler**, view instructions at this link:
<https://tinyurl.com/codeism-2024-cpp-setup>

Structure of code in C++

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

int main()
{
    // You write your code here
    return 0;
}
```

Meaning of these lines :

(It's completely okay, if you can't understand the meaning much, you will understand those, after learning some basics concepts of the language)

1. **#include <bits/stdc++.h>**

```
// This line imports all the standard
// necessary libraries
```

2. **using namespace std;**

```
// Just, an indication to the compiler, that we
// will use the standard functions
```

3. int main()

// main() function is the entry point function
of our program

// Our program will start running from the
main() function always

// main() is a compulsory function() and
required in every C++ program

4. return 0;

// We return 0 from main() function to indicate
that our program ended error-free

5. In C++, every statement should end with a semicolon (;)

[Similar to full-stop (.) in English language
or | in Hindi]

6. In C++, capital letters (A,B,C) are different from small letters (a,b,c). (So, C++ is a case sensitive language)

Output

- Output means to display something on the screen, in our program
- We use cout with << operator (Output operator or insertion operator)

Eg. cout<<88;

- We use the term **string** for english sentences or words
Suppose I want to print a string
`cout<<"Hello CodeISM";`

- You can also combine multiple outputs using `<<` (Output operator)

Eg. `cout<<"Hello CodeISM "<<2024;`

- You can also write something on the next line using '`\n`' (**Newline character**)

Eg.

```
cout<<'\n'<<"Welcome";
```

- We use `{}` (**curly brackets or braces**) for enclosing similar things in one group

- **Caution:** You can also use `cout<<endl;` to print on the next line but it is slow compared to '`\n`' (It is slow because it flushes the buffer everytime it is used)

Difference between 'a' and “a”

- Single inverted comma is always used for 1 single character. Eg. `cout<<'a';`
- Double inverted comma is used for strings
Eg. `cout<<"Hello world";`

Variables

- Consider variables as a box, where some value is stored.
- Value of variable can be changed

Syntax for declaring a variable:

<data-type> <name-of-variable> ;

Eg. int temp;

// Computer will give some memory to a box
// named “temp”



temp

- **For naming a variable**, follow these rules:

1. Variable name shouldn't start with a number
2. Variables should not contain space.

Instead use underscore.

int abc def; // error : Incorrect name of
//variable

3. Variables should not use any reserved name . Like int

- You can also give a value to a variable
using =

Eg.

1. int temp=8;
2. int temp;
temp=8;

// Computer will give some memory to a box
// named “temp” and give it a value 8.



8

temp

Program to add 2 numbers

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int a=4;
    int b=10;
    int c=a+b;
    cout<<c;
    return 0;
}
```

Input

- For input, use `cin` and `>>` (Input operator or extraction operator)

Eg. `int num;
cin >> num;`

Program using user-input to multiply 2 numbers

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

int main()
{
    int a,b,result;
    cout<<"Enter first number:";
    cin>>a;
    cout<<"Enter second number:";
    cin>>b;
    result=a*b;
    cout<<"Answer is: "<<result<<'\n';
    return 0;
}
```

Program to use multiple inputs in same line

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

int main()
{
    int a,b,result;
    cout<<"Enter 2 numbers:";
    cin>>a>>b;
    result=a*b;
    cout<<"Answer is: "<<'n'<<result;
    return 0;
}
```

Data Types

1. int // integer

Range of integer in C++ is -2³¹ to +2³¹
(-2,147,483,648 to 2,147,483,647)

2. long int // long integer

Range increases

3. long long int // long long integer

Range is -10¹⁸ to +10¹⁸

Exact range of long long int

(No need to remember this. Just remember approx range from above)

**-9,223,372,036,854,775,808 to
9,223,372,036,854,775,807**

4. **float // for fractional / decimal values**

Eg. **float pi=3.14159;**
// Float has less precision

5. **double // Also used for**

fractional/decimal values, has greater precision than ‘float’

6. **long double // Also used for fractional/decimal values, highest precision for decimal values**

7. **string // for english words or sentences**
string str = “Hello everyone”;
cout<<str;

8. **char // for single character**
Eg. **char ch='p';**
cout<<ch;

Program to print double of a decimal (fractional) number

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    float abc;
    cin>>abc;
    cout<<abc*2;
    return 0;
}
```

Program to print double of a decimal (fractional) number without user input

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

int main()
{
    long double abc=23.2929;
    cout<<abc*2;
    return 0;
}
```

Program to print a word taken from user

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    string str;
    cin>>str;
    cout<<"str="<
```

// Try giving “Hello world” as input in your screen with above program

- Input Operator (>>) will stop at spaces (or any invisible character like Enter key)

Program to print 2 words taken from user

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

int main()
{
```

```
string str,str2;
cin>>str>>str2;
cout<<"words are "<<str<<" "<<str2<<'\n';
return 0;
}
```

Program to take 1 full line as input

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

int main()
{
    string str;
    getline(cin,str);
    cout<<"Full sentence is "<<str;
    return 0;
}
```

Constants

- Just has a **constant value**. You will get an error if you try to change the value of a constant.
- Use const for declaring a constant

Eg.

```
const int abc=0;  
const float pi=3.14;
```

#Functions in C++ -

- A function is a group of statements that together perform a task.
- A function may take some inputs (or arguments) and may or may not return a value.
Like a $\sin(x)$ function may return a float value but a function which just prints something, may not return a value

Syntax of function definition

(Used when you need to create your own function)

```
[return type] [function name] (inputs)  
{
```

.

.

All the tasks you wanna perform;

.

.

Finally return the value(If any);

```
}
```

Examples:-

1. Function to return the sum of 2 integers

```
int Add(int firstNum, int secondNum) {  
  
    int sum = firstNum + secondNum;  
    return sum;  
}
```

2. Function to print the info of a student.

```
void PrintInfo(string name,int age,double height)  
{  
    cout<<"My name is:"<<name<<endl;  
    cout<<"My age is:"<<age<<endl;  
    cout<<"My height is:"<<height<<"\n";  
}  
  
// Why return type of this function is void ?  
// Because this function doesn't return anything. Simple
```

When to use a void function?

- When your function performs a task but it doesn't need to return a value (or when it doesn't need a value to be used somewhere else)
- Like, you went to a cinema hall, watched a movie, but after returning, you didn't bring anything from there. i.e. You performed a task but didn't

- need to remember a value for later use
- Similarly, in C++, you may want to print something but not return a value, so you will use a void function in that case.

Calling a function

- Once a function is defined (or created) as above, you can call the function inside any other function like the main() function, to execute it.
- While calling a function, you can simply pass the inputs (or arguments) to it using () (round brackets or parentheses).

Examples:-

1. Calling the above Add() function to add 3 and 8 to print the answer on the screen

```
int ans = Add(3,8);  
cout<<ans;
```

2. Calling the above PrintInfo() function to print some information

```
PrintInfo("Chirag", 18, 6.0);
```

Complete code to print the info. using function

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

void PrintInfo(string name,int age,double height)
{
    cout<<"My name is :"<<name<<endl;
    cout<<"My age is: "<<age<<endl;
    cout<<"My height is: "<<height<< "\n";
}

int main()
{
    string name;
    int age;
    double height;
    cin>>name;
    cin>>age;
    cin>>height;

    PrintInfo(name,age,height); //Function Call

    PrintInfo("Chirag", 18, 6.0);
}
```

```
//You may also call the function like this, It is hard-coded.  
  
    return 0;  
}
```

Homework Questions :

1. Write a program to print the cube of a number (take user input)
2. Write a program to take a temperature in degree Celsius and print the equivalent temperature in degree Fahrenheit.
3. Write a program to take the radius of a circle as input and print the area and perimeter of the circle in different lines
4. Write a program to take a First Name and Last Name of a user and print the full name. Also take college name (with spaces) as input and print it.
5. Implement your own subtraction, multiplication, and division functions which take two numbers as an input and return the answer after performing the given task.
6. Write a program to swap the value of 2 variables.
[Hint: Maybe, using a third variable helps?]
7. Write a program to swap the value of 2 variables without using any 3rd variable
8. Write a program using a function to swap the values of 2 variables using a function. The values swapped inside the function, should also be swapped outside that function.
9. Write a program to return the remainder when a number is

divided by another (Take both numbers as input)

Solution of Q.4

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

```
/*
```

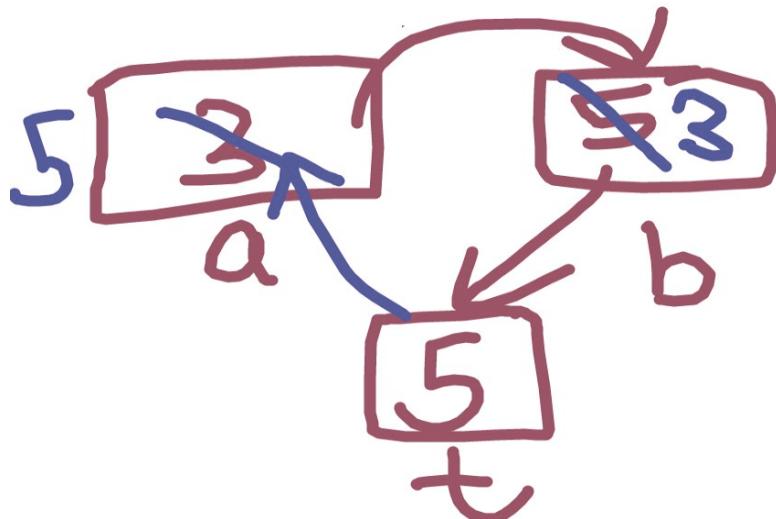
Write a program to take a First Name and Last Name of a user
and print the full name.

Also take college name (with spaces) as input and print it.

```
*/
```

```
int main()
{
    string fname;
    string lname;
    cout<<" Enter first name: ";
    cin>> fname;
    cout<<" Enter last name: ";
    cin>>lname;
    cout<<"Full name is : "<<fname<<" "<<lname<<'\n';
    string college;
    cout<<" Enter college name: ";
    getline(cin, college);
    getline(cin, college); // getline stops reading input when
    "Enter key" is pressed
    cout<<"College name is :"<<college;
    return 0;
}
```

6.



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

/*
Swap 2 variables.
*/

int main()
{
    // = operator moves RHS value to LHS
    int a=3;
    int b=5;
    int t;
    cout<<"Earlier values \n";
    cout<< a<< " "<< b;
    t = b;
    b = a;
```

```
a = t;
cout<<"Values now \n";
cout<<a<<" "<<b;
return 0;
}

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

/*
Swap 2 variables without a third new variable.
*/

int main()
{
    // = operator moves RHS value to LHS
    int a=3;
    int b=5;
    cout<<a<<" "<<b<<'\n';
    a = a + b;
    b = a - b;
    a = a - b;
    cout<<a<<" "<<b<<'\n';

    return 0;
}
```

C++

- For competitive programming

Function calling methods - Call by Value and Call by Reference

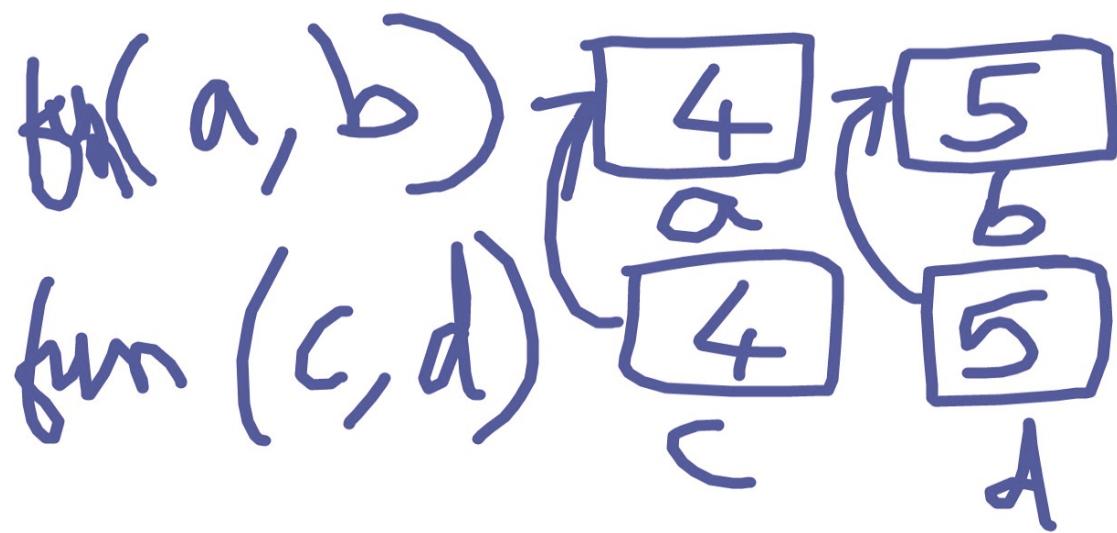
We have function `fun(int a, int b)` .

Let us say, in `main()`, we call it as:

```
int c=4,d=5;
```

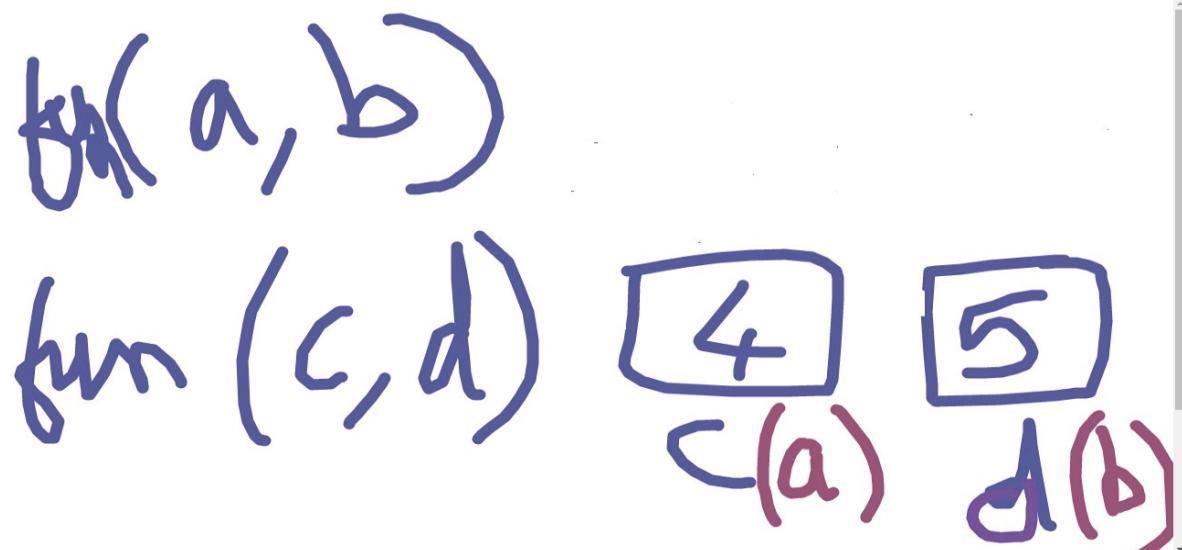
```
fun(c,d)
```

By default, the function call uses a **call by value** mechanism. In call by value, function creates different copies of all the input parameter. So, any change in value inside the function is not reflected outside that function.



We use call by reference mechanism to return the new (changed) values of input parameters of function in main()

Consider same function, but in call by reference , it just creates an alias (or pet name) of the same variables. It don't create a new variable.



To use call by reference, use & sign in front of the parameters in function definition only.

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

// Swap 2 variables without a third new variable.
```

```
void swapNums(int & a, int & b)
{
    cout<<"In function, before swapping : a=<<a<<"  
b=<<b<<'\\n';
    int t=b;
    b=a;
    a=t;
    cout<<"In function, after swapping : a=<<a<<"  
b=<<b<<'\\n';
}

int main()
{
    // = operator moves RHS value to LHS
    int a=3;
    int b=5;
    cout<<"In main(), before swapping : a=<<a<<"  
b=<<b<<'\\n';
    swapNums(a,b);
    cout<<"In main(), after swapping : a=<<a<<"  
b=<<b<<'\\n';
    return 0;
}
```

You use call by reference, when you want that input parameters of function should be changed outside that function (where it is called)

Which is faster ? Why ?

Call by reference, because it just gives a pet name to the original variables passed.

= operator (Assignment Operator)

= operator moves RHS value to LHS

For eg.

```
int a;  
a = 2;  
// After this line, value of a becomes 2
```

Recursive functions

Recursive functions are a function which calls itself in the function definition.

Eg. // Suppose, you are creating function for factorial calculation

// == operator in C++ is used for equality checking

```
int factorial(int n)  
{  
    if (n==0)  
    {
```

```
        return 1;
    }
    return n * factorial(n-1);
}
int main()
{
cout<<factorial(4);
return 0;
}
```

Always a function stops when it encounters a return statement.

Eg. call factorial(4);
return 4 * factorial(3)
return 4 * 3 * factorial(2)
return 4 * 3 * 2 * factorial(1)
return 4 * 3 * 2 * 1 * factorial(0)
return 4 * 3 * 2 * 1 * 1

...

This will stop at n = 0

So, for stopping (terminating) any recursive function, you always need a base condition. Like, in above function, if (n==1) return 5;
(If you remove that condition, it will give an infinitely running program)

Homework Questions on Recursion

1. Create a recursive function `rec_power(int base, int p)` to find the value of base^p (base raised to power p)
2. Create a recursive function to find the sum of digits of a number
3. Create a recursive function to print the multiplication table of any number.

Basics of arrays

Array are used when you need to store a large number of variables of same data type.

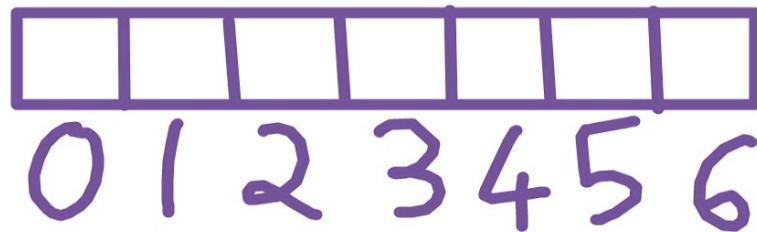
Syntax for declaring array :

`<data-type> <name-of-array> [<array-size>]`

`<array-size>` should always be a **constant positive integer**. For example,
`int marks [100];`

```
const int size=100;  
int marks[size];
```

Counting or indexing of elements of array starts with 0.



How to access a particular element of an array ?

<arry-name> [index or position of the element]

For eg. marks[0]; // gives first element

marks[1]; // gives second element

marks[2]; // gives third element

..... and so on

i.e. marks[i - 1] gives i^{th} element

Example 1 Take input of 2 students marks in an array of size 6

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

int main()
{
```

```
int marks[6];
cout<<"Enter marks of first student: ";
cin>>marks[0];
cout<<"Enter marks of second student: ";
cin>>marks[1];
cout<<"Marks are : "<<marks[0]<<
"<<marks[1]<<'\n';
return 0;
}
```

Example 2 Find size of an array

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

int main()
{
    int marks[6];
    cout<<" No. of element is array :
"<<sizeof(marks)/sizeof(int);
    return 0;
}
```

Relational Operators (a few call it Conditional Operators)

1. == Checks the equality of LHS and RHS

Eg.

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

int main()
{
    cout<<(2==2)<<'\n';
    cout<<(3==2)<<'\n';
    return 0;
}
```

Output

```
1
0
```

(In C++, **true** is equivalent to **1** and **false** is equivalent to **0**)

2. != Checks non-equality

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

int main()
```

```
    cout<<(2!=2)<<'\n';
    cout<<(3!=2)<<'\n';
    return 0;
}
```

Output:

```
0
1
```

- 3. < operator (Less than)**
- 4. > operator (Greater than)**
- 5. <= operator (Less than or equal)**
- 6. >= operator (Greater than or equal)**

For eg.

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

int main()
{
    cout<<(2<=3)<<'\n';
    cout<<(2>3)<<'\n';
    return 0;
}
```

Try it yourself: You can compare strings, char, etc. with == operator, != operator etc.

Caution : Don't compare floating point (or double or long double data types) with == or != . Because it may not give you an accurate value.

If statements

Syntax of if statement :

```
if ( condition )
{
    // statements
    // ...
    // ...
}
```

For eg.

```
int a=2;
if (a < 5)
{
    cout<<"Less than 5";
}
```

```
int a=2;
if (a > 5)
{
    cout<<"Greater than 5";
}
```

If else statement

Syntax:

```
if ( condition )
{
// statements
// ...
// ...
}
else
{
// statements
// ...
// ...
}
```

Example:

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

```
int main()
{
    int a=8;
    if (a < 5)
    {
        cout<<"Less than 5";
    }
    else {
        cout<<"Greater than 5";
    }
    return 0;
}
```

Logical Operators

&& - AND operator in C++

|| - OR operator in C++

! - NOT operator in C++

Example 1 -

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

int main()
```

```
int percentile;
int boardPer;

cout<<" Enter Percentile :";
cin>>percentile;

cout<<" Enter Board percentage :";
cin>>boardPer;

if ( (percentile > 85) || (boardPer > 75)
)
{
    cout<<"You are qualified";
}
else
{
    cout<<"Not qualified, sorry";
}

return 0;
}
```

Example 2 -

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

```
using namespace std;

int main()
{
    int percentile;
    int boardPer;

    cout<<" Enter Percentile :";
    cin>>percentile;

    cout<<" Enter Board percentage :";
    cin>>boardPer;

    if ( (percentile > 85) && (boardPer > 75) )
    {
        cout<<"You are qualified";
    }
    else
    {
        cout<<"Not qualified, sorry";
    }

    return 0;
}
```

Example 3

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

int main()
{
    int percentile;
    int boardPer;

    cout<<" Enter Percentile :";
    cin>>percentile;

    cout<<" Enter Board percentage :";
    cin>>boardPer;

    if ( !((percentile > 85) && (boardPer > 75)) )
    {
        cout<<"You are qualified";
    }
    else
    {
        cout<<"Not qualified, sorry";
    }

    return 0;
}
```

```
}
```

Many if ..else can be combined also.

Example 4

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

int main()
{
    int n;
    if (n==0)
    {
        cout<<"Zero";
    }
    else if(n==1)
    {
        cout<<"One";
    }
    else if(n==2)
    {
        cout<<"Two";
    }
    else if(n==3)
    {
        cout<<"Three";
    }
}
```

```
    return 0;  
}
```

Loops

- You want to perform same piece of code, many number of times. Then loops are your saviour!
- In C++, there are 3 types of loops:
 1. While loop
 2. For loop
 3. Do while

While Loop

Syntax:

```
while (condition for running loop)  
{  
    // statements  
}
```

Example: Program to print hello 10 times

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

```
using namespace std;

int main()
{
    int cnt=0;

    while(cnt<10)
    {
        cout<<"Hello ";
        cnt=cnt+1;
    }

    return 0;
}
```

2. Print hello Infinite times

Note that 1 in condition means true and 0 means false.

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

int main()
{
    int cnt=0;
```

```
while(1)
{
cout<<"Hello ";
cnt=cnt+1;
}

return 0;
}
```

Break statement (I wana jump out of loop)

Example

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

int main()
{
    int cnt=0;

    while(1)
    {
        cout<<"Hello ";
        cnt=cnt+1;

        if (cnt == 10)
```

```
{  
    break; // break will stop your loop  
}  
  
}  
  
return 0;  
}
```

Program for printing counting from 1 to 8

```
#include <bits/stdc++.h>  
using namespace std;  
  
int main()  
{  
    int i = 1;  
    while (i<=8)  
    {  
        cout<<i<< '\n';  
        i++;  
    }  
  
    return 0;  
}
```

For Loop

```
for ( <variable-initialisation> ; <condition> ;  
<update-variable> )  
{  
    // statements  
}
```

<variable-initialisation> means to give a starting value to any variable . It occurs only 1 time.

For eg.

```
for (int i=1 ; i<=8; i=i+1 )  
{  
    cout<<i<<'\n';  
}
```

++ operator increases value of variable by 1

For eg.

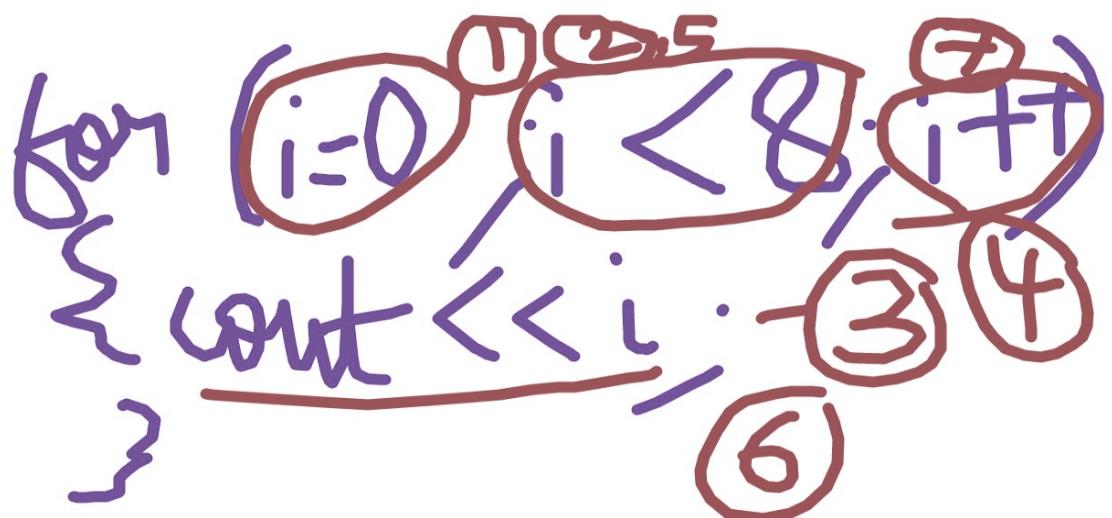
```
int a=4;  
a++;  
cout<<a<<'\n';
```

-- decrease value of variable by 1

For eg.

```
int a=4;  
a--;  
cout<<a<<'\n';
```

Order of execution of statement in a for loop



Example:

```
#include <bits/stdc++.h>  
using namespace std;  
  
int main()  
{  
    for (int i=1 ; i<=8; i++ )  
    {
```

```
        cout<<i<<'\n';
    }
    return 0;
}
```

Do while loop

Syntax-

```
do
{
    // statements
} while (condition) ;
```

It always executes at least one time

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

int main()
{
    int i=1;
    do
    {
        cout<<i<<'\n';
        i++;
    } while(i<=4);
```

```
    return 0;  
}
```

auto for loop for strings

Q. Print all characters of a string in different line

```
#include <bits/stdc++.h>  
using namespace std;  
  
int main()  
{  
    string str="Hello world";  
    for(auto c: str)  
    {  
        cout<<c<<'\\n';  
    }  
    return 0;  
}
```

Practice Questions (Do try these)

1. Take a number as input from user and print whether the number is odd or even.

[Hint : Use % operator to find remainder. In C++, a%b gives remainder when a is divided by b]

2. Take marks of a student as input. And print his grade as per the following distribution

Marks	Grade
0-40	F
41-60	C
61-80	B
81-100	A

3. Write a program to take input of a number n and print the multiplication table of n using a for loop.

4. Write a program to take input of a number and print sum of its digits using a while loop.

5. Write a program to find the sum of the series: (using while loop)

$1+11+111+\dots+111$ upto n.

(Take n as input from user)

For eg. if n=4, the series is : $1+11+111+1111$.

6. Take marks of 10 students as an input in an array of size 10 from user and print the average marks and highest marks using for loop.

7. Write a program to take a number n as input and print the following star triangle pattern with n lines: (using for loop)

Eg. For n=4, the pattern would be :

```
*  
**  
***  
****
```

8. Write a program to take a number n as input and print the following numerical triangle pattern with n lines: (using for loop)

Eg. For n=4, the pattern would be :

```
1  
1 2  
1 2 3  
1 2 3 4
```

9.

<https://www.hackerrank.com/challenges/staircase/problem>

10.

<https://www.hackerrank.com/challenges/kangaroo/problem>

Time and Space complexity and other Pre-Requisites for Competitive Programming

Generally, the time limit for running your code on platforms like Hackerrank, Atcoder, Codeforces, etc. is 1-2 seconds. So, we need to write an efficient code that passes this time limit constraint.

Big Oh (O)

- It represents the **upper bound of a function**
- Used to approximate time complexity of a code

If you have function $f(x)$, then consider a function $g(x)$ such that

$$f(x) \leq c.g(x)$$

For all values of $x >= x_0$ for some value of x_0

Then we say, $f(x) = O(g(x))$

Eg. 1:

If $f(n) = n - 2$

$f(n) \leq 8.n$

$f(n) \leq 8.g(n)$

Where $g(n) = n$

By definition, $f(n) = O(n)$

Eg. 2: If $f(n) = 3n^2 + 5n + 8$

By definition of Big Oh, $f(n) = O(n^2)$

[In polynomial functions, only see the highest degree term to find Big Oh]

Eg. 1 Consider an array A of size n

```
for(int i=0; i<n; i++)
{
    cout<<A[i];
}
```

Find time complexity of this code.

$i=0$; will run only 1 time

$i < n$; will be run $n+1$ times

$i++$ will be run n times

$cout \ll A[i]$; will be run n times

$$\text{Time complexity} = 1 + n+1 + n + n$$

$$= 3n + 2$$

$$= \mathbf{O(n)}$$

Eg. 2 Find time complexity of this code.

```
int val;  
bool found=false;  
for(int i=0; i<n; i++)  
{  
    if ( A[i] == val )  
    {  
        found=true;  
        break;  
    }  
}
```

We always consider the worst case scenario in finding time complexity of a code.

Here, the worst case is when the array A doesn't contain the value val. So, loop will run n times.

Thus, Time complexity: **O(n)**

Eg. 3 Find time complexity of this code.

```
int b;  
int a = 2*b;  
cout<<b;
```

Time complexity: **O(1) [constant]**

Eg. 4 Find time complexity of this code.

```
for(int i=0; i<n; i++)
{
    for(int j=0; j<i; j++)
    {
        cout<<2;
        ...
    }
}
```

i=0; inner loop will run 0 times

i=1; inner loop will run 1 times

i=2; inner loop will run 2 times

....

i=n-1 ; inner loop will run n-1 times

$$\text{Total steps} = 0 + 1 + 2 + \dots + n-1$$

$$= (n * (n-1)) / 2$$

$$= O(n^2)$$

Eg. 4: Find time complexity of this code.

```
for(int i=1; i<=n; i=i*2)
{
    sum=sum+i;
}
```

For $i=1, 2, 4, 8, 16, \dots, 2^k$

Let us assume that it breaks out of loop in k steps

$$1.(2^k) > n$$

$$2^k > n$$

$$k \text{ approximately } \log_2(n)$$

Time complexity: $O(\log_2(n))$

Eg. 5 Find time complexity of this code.

```
for(int i=n; i>0; i--)
{
    for(int j=0 ; j<i; j=j+2)
    {
        .....
        cout<<1;
    }
}
```

For $i=n$, inner loop will execute $n/2$ times

$i=n-1$, inner loop will execute $(n-1)/2$ times

$i=n-2$, inner loop will execute $(n-2)/2$ times

.....

$i=1$, inner loop will execute 1 time

Time complexity: $n/2 + (n-1)/2 + (n-2)/2 + \dots$

$= O(n^2)$

HW-1: Find time complexity of this code.
(Find answer at the end of this doc)

```
for(int i=1; i*i<=n; i++)  
{  
cout<<2;  
}
```

Important: In 1 second, only 10^7 - 10^8 operations can be performed.

Eg. 6 : Suppose, $n \leq 10^5$

(i) You have written a code with time complexity $O(n^2)$. Find whether your code will pass time limit of 1 second.

Sol: In worst case, code takes 10^{10} operations to perform. But, this is greater than 10^7 - 10^8 . So, it is slow. It will not pass the time limit of 1 second.

(ii) If you write code with time complexity $O(n \log n)$ Find whether your code will be able to pass time limit of 1 second.

Sol: In worst case, $N \log N = 10^5 \cdot \log(10^5) \leq 10^8$

So, It will pass the time limit of 1 second.

Important: In general, $O(1) < O(\log N) < O(\sqrt{N}) < O(N) < O(N \log N) < O(N^2) < O(N^3) < \dots < O(N^{100}) < O(2^N)$

Common errors in online platforms

1. Compiler error (CE)

- Indicated by compiler itself with the line number in which there is error

2. Wrong Answer (WA)

eg. Yes not = YES

- Read the input and output format in the question very carefully

3. Time Limit Exceeded (TLE)

Time limit is generally 1 second. And if your code is slow to pass this time limit, you will get this error.

- You can also use Fast input / output with cin, cout:

```
int main()
{
    ios_base::sync_with_stdio(false);
```

```
cin.tie(NULL);
cout.tie(NULL);

.....
// All your code after this
}
```

4. Runtime error

- (a) If you are accessing an invalid element of an array.

```
int arr[100];
cout<<arr[1000]; // runtime-error
cout<<arr[-1]; // runtime-error
```

- (b) When you divide by 0

```
cout<<a/0;
```

(c) Overflow [Important to prevent such errors]

Try running this code:

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

int32_t main()
{
```

```
int a=1000000000;
int b=1000000000;
int ans=a*b;
cout<<ans;
return 0;
}
```

// Output: -1486618624 (something like this)

// Surprising, right ?

Why this happens?

int can store integers only upto 10^9 approximately.

Numbers greater than this, can't be stored in an int variable.

For bigger integers , upto 10^{18} , use long long variable.

```
int a=1000000000;
int b=1000000000;
long long ans=a*b;
cout<<ans;
```

// Output: -1486618624 (something like this)

Still, it will give the same wrong answer

**Because you need to convert the integer to long long,
during the multiplication also.**

Now, try running this code:

```
int a=1000000000;
int b=1000000000;

long long ans = (long long)a * b;
cout<<ans;
```

Now, you will get correct answer

One more method, is always use long long variables.

```
long long a=1000000000;
long long b=1000000000;

long long ans = a * b;
cout<<ans;
```

// This is also correct

**Checking equality of floating point numbers
(decimal numbers)**

Never compare floating point numbers with == sign

```
float a=1.00000001;
float b=1.00000000;
```

```
if (a==b)
{
cout<<"equal";
}
else
{
cout<<"Not equal";
}
```

// The above code may give wrong answer in some places
due to lack of precision in float operations

We use this method for comparison:

```
const float eps = 0.000001; // 1e-6
float a=1.00000001;
float b=1.00000000;
if ( abs(a-b) < eps )
{
    cout<<"Equal";
}
else
{
cout<<"Not equal";
}
```

Space Complexity:

It denotes the Big Oh of space taken by variables in a program, etc.

Eg. 1

```
int arr[n];
```

This takes $O(n)$ space

Eg. 2

```
int arr[n][m];
```

This takes $O(n*m)$ space

You can't declare an integer global array of size $> 10^7$ or 10^6 . So, while declaring an array, take care of the size.

Otherwise, you get a **MLE (Memory limit exceeded) error** on platforms like hackerrank, atcoder, etc.

Ans of HW-1: **$O(\sqrt{N})$**

Where \sqrt{N} = square root of N

STL Vectors in C++

Arrays: A collection of variables of same data-type. It has a **fixed (or static) size**

```
int arr[100];  
// an array of size 100
```

Vectors are arrays with **dynamic size**. Vectors are provided by STL library of C++

When you use vectors ?

When you don't know the size of the array required beforehand, you use vectors.

In `#include<bits/stdc++.h>` , vectors are already included. So, you don't need to include any extra header file for using vectors.

Syntax for declaring vector:

`vector <data-type> name; // for an empty`

`vector <data-type> name(size);`

Eg.

```
vector<int> vec; // vec is empty
```

```
vector<int> vec(1000); // vec is a vector of size 1000
```

.size()

Time complexity of O(1)

```
cout<<vec.size();
```

Accessing a particular element of vector

0-based indexing

vec[0] => 1st element of vector

vec[1] => 2nd element of vector

vec[2] => 3rd element of vector

.....

vec[n-1] => nth element of vector

Declaring vector with some default value

```
vector<data-type> name(size, defaultValue);
```

Eg. `vector<int> vec(5, 2);`

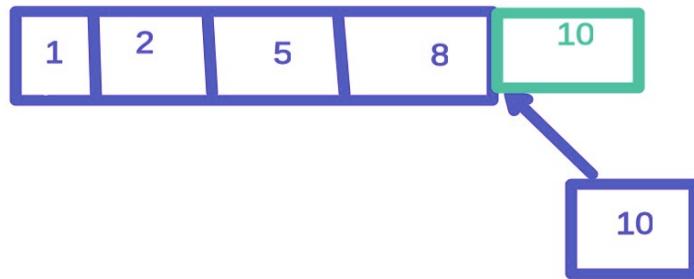
`{ 2 , 2, 2, 2, 2 }`

Printing all the values of a vector

```
int n=vec.size();
for(int i=0; i<n; i++)
{
    cout<<vec[i]<<" ";
}
```

.push_back()

Pushes a value at the back of vector and increases size of vector by 1



Syntax:

```
vec.push_back(value);
```

Time complexity:

O(1)

Eg. 1 A code using push_back():

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

int32_t main()
{
    vector<int> vec(3, 8);
    vec.push_back(22);
    int n=vec.size();
    for(int i=0; i<n; i++)
    {
        cout<<vec[i]<<" ";
    }

    return 0;
}
```

Eg. 2 Difference between changing value using vec[i] and push_back():

```
vector<int> vec={2, 3, 4, 6, 9};
```

Case I: (Changing value using vec[i])

```
vec[2]=99;
```

```
// New value of vec: { 2, 3, 99, 6, 9}
```

Case II: (Inserting element using push_back())

```
vec.push_back(100);
```

```
// New value of vec: { 2, 3, 4, 6, 9, 100};
```

.resize()

Resize the vector to a different size

Eg.

```
vec.resize(100);
```

```
// Make the size of vector 100
```

.sort()

```
// Used to arrange all elements of vector in increasing or  
ascending order
```

// O(N log N)

```
sort(vec.begin(), vec.end());
```

.reverse()

// Used to reverse a vector

// O(N)

```
reverse(vec.begin(), vec.end());
```

Q. Suppose, the input format is given like this:
All elements are given as space separated integers and last element is -1. Write program to take input of these integers.

```
int num=100;
vector<int> vec;
while(num != -1)
{
    cin>>num;
    vec.push_back(num);
}
```

For auto loop to print all elements:

```
for(auto num: vec)
{
    cout<<num<<" ";
```

Note: But using this syntax, you can't change any value of the original array. (Similar to call by value)

If you want to change all the elements of the array, use reference variable by putting & sign before variable name in loop (Similar to call by reference)

Example

```
for(auto &num: vec)
{
    num=100;
}
```

STL Pairs, sort() and structure in C++

pop_back() in vector

- Removes the last element of a vector

```
vector<int> vec={1,2,3,4,5}; // (size=5)  
vec.pop_back();
```

After pop_back(),

vec becomes {1,2,3,4}; (size=4)

Time Complexity = O(1)

sort()

Time Complexity = O(n log n)

vec:={5,1,7,3,4,9}

syntax-> sort(starting_iterator, ending_iterator)

starting_iterator-> vec.begin();

ending_iterator-> vec.end();

Example

```
sort(vec.begin(),vec.end());
```

swap()

Syntax: swap(a, b);

swap() is used to interchange the values of any 2 variables

Example:

```
int a=5;  
int b=6;  
swap(a,b);  
cout<<a<<endl;  
cout<<b<<endl;
```

Example for sort() :

Let n = size of vector

```
vec:= {1,2,3,4,5,6};  
Starting_itr -> vec.begin();  
Second_itr -> vec.begin()+1;  
Third_itr -> vec.begin()+2;  
..  
Last_itr -> vec.begin()+n-1;  
vec.begin()+n == vec.end();  
sort(vec.begin(), vec.begin()+n);  
sort(vec.begin(), vec.end());  
L---r sort(l,r+1);
```

reverse()

Time complexity: O(n)

vec-> {4,1,2,8,3};

How will you sort in decreasing order ?

```
sort(vec.begin(), vec.end()); // {1,2,3,4,8}
reverse(vec.begin(), vec.end()); // {8,4,3,2,1}
```

How do we sort in case of array?

```
int arr[5] = {4,1,2,8,3};
```

n = size of array (Here, it is 5)

Starting_itr -> arr;

Second_itr -> arr+1;

Third_itr -> arr+2;

..

Last_itr -> arr+n-1;

Ending_itr -> last_itr+1 == arr+n-1+1 == arr+n;

```
sort(arr, arr + n);
```

Code-1 (To reverse an array without using reverse())

```
#include <bits/stdc++.h>
using namespace std;
int main(){
```

```
int arr[5]={4,1,2,8,3};  
int n=sizeof(arr)/sizeof(int);  
int l=0,r=n-1;  
while(l<=r){  
    swap(arr[l],arr[r]);  
    l++;  
    r--;  
}  
for(int i=0;i<n;i++) cout<<arr[i]<<" ";  
return 0;  
}
```

Struct

Code-1

```
#include <bits/stdc++.h>  
using namespace std;  
  
struct Freshers{  
    string name;  
    string AdmNo;  
    int age;  
    double height;  
};
```

```
// structure definition ends with a semicolon (;)
int main(){

    Freshers fresher;
    fresher.name = "Manyank";
    fresher.AdmNo = "20JE0655";
    fresher.age = 18;
    fresher.height = 6.1;

    cout<<"Info of freshers is :"<<endl;
    cout<<fresher.name<<endl;
    cout<<fresher.AdmNo<<endl;
    cout<<fresher.age<<endl;
    cout<<fresher.height<<endl;
    return 0;
}
```

Code-2

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

struct Point{
    int x;
    int y;
```

```
};

int main(){

    Point point[n];
    for(int i=0;i<n;i++)
cin>>point[i].x>>point[i].y;

    (x1,y1);
    (x2,y2);
    (x3,y3);
    ...
    (xn,yn);

    int X[n];
    int Y[n];
    for(int i=0;i<n;i++) cin>>X[i]>>Y[i];

    (xi,yi);

    cout<<X[i]<<" "<<Y[i]<<endl;

    return 0;
}
```

Pair in C++ STL

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

int main(){
    pair<int,int> point;
    cin>>point.first>>point.second;
    cout<<point.first<<" "<<point.second<<endl;

    pair<string,double> Fresher;
    cin>>Fresher.first;
    cin>>Fresher.second;

    cout<<"The name of the student is: " <<
Fresher.first << endl;
    cout<<"The height of the student is: " <<
Fresher.second << endl;

    return 0;
}
```

Point in 3D->

```
#include <bits/stdc++.h>
using namespace std;
int main(){

    pair<int,pair<int,int>> point3D;
    // x-> point3D.first;
    // y-> point3D.second.first;
    // z-> point3D.second.second;

    return 0;
}
```

Hackerrank Problem: Equalize the array

Link:

<https://www.hackerrank.com/challenges/equality-in-a-array/problem>

(First, try it by yourself, then only look at the solution below)

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n;
```

```

cin>>n;
int a[n]; // 3 1 1 2 2 2 3 3 3
for(int i=0;i<n;i++) {
    cin>>a[i];
}
sort(a,a+n); // 1 1 2 2 2 3 3 3
int ans = 10000;
for(int i=0;i<n;i++) {
    int freq=0;
    int com = a[i];
    while(i<n&&a[i]==com) {
        freq++;
        i++;
    }
    i--;
    ans = min(ans,n-freq);
}
cout<<ans;
}

```

Try the below problem, it would be discussed in next class:

https://atcoder.jp/contests/abc187/tasks/abc187_d

STL set and map in C++

Some important C++ STL(Standard Template Library)

Containers

Vector

Try to solve this problem:

https://atcoder.jp/contests/abc187/tasks/abc187_d

Sol:

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
int32_t main()
{
    int n;
    cin>>n;
    pair<int,int> a[n];
    for(int i=0;i<n;i++){
        cin>>a[i].first>>a[i].second;
    }
    int x=0,y=0; // x-> aoki's votes and y->
    takahashi's votes
    for(int i=0;i<n;i++){
        x+=a[i].first;
    }
```

```

for(int i=0;i<n;i++){
    int temp = a[i].first;
    a[i].first = 2*a[i].first+a[i].second;
    a[i].second = temp;//{2a+b,a}
}
sort(a,a+n);
int ans=0;
int i=n-1;
while(x>=y){
    ans++;
    y+=a[i].first-a[i].second;
    x-=a[i].second;
    i--;
}
cout<<ans;
/*
    a[i].first=a,a[i].second=b;
    2*a+b
*/
}

```

Note: By default, the sort() function sorts the vector elements on basis of first element of pairs and compare the second element, only when first element of two pairs are same.

Why we used #define int long long ?

- To prevent overflows

- Since, it will replace int by long long everywhere in the program

Why we used `int32_t main()` instead of `int main()` ?

- Because int is replaced by long long in our program, it will become `long long main()`, which gives error on compiling. So, we need to write `int32_t main()`. `int32_t` is same as `int` .

Set

set is a special kind of STL container which stores **unique** elements in **sorted order**.

Syntax of declaring a set

```
set<data_type> set_name;
```

Examples:

```
set<int> st;
set<float> st2;
set<double> st3;
```

```
set<int> st;
```

insert() function in set

Time complexity: $O(\log n)$

```
st.insert(4); // {4}
st.insert(3); // {3,4}
```

```
st.insert(1); // {1,3,4}  
st.insert(3); // {1,3,4}
```

size() function in set

Time complexity: O(1)

Example:

```
int n=st.size();
```

erase() function in set

Time complexity: O(log n)

Example:

```
st.erase(3);
```

1. If the number is present then it will remove it from the set.
2. If not then nothing happens.

empty() function in set

To know whether the set is empty or not

Time complexity: O(1)

```
bool isEmpty = st.empty(); // true/false
```

Q. You will be given n numbers and after inserting each number you have print all the inserted numbers till now in sorted form;

1 -> 1
3 -> 1,3
2 -> 1,2,3

Method 1 - Use set

```
set<int> st;
for(int i=0;i<n;i++){
    int x;
    cin>>x;
    st.insert(x); // O(logn);
    print(st); // O(n);
}
```

Total Time complexity: $O(n^2)$

Method 2 - Use array

```
vector<int> vec;
for(int i=0;i<n;i++){
    int x;
    cin>>x;
    vec.push_back(x);
    sort(vec.begin(),vec.end()); // O(nlogn)
    int sz = vec.size();
```

```
for(int j=0;j<sz;j++) cout<<vec[j]<<" "; // O(n);
cout<<endl;
}
```

Time complexity: $O(n^2 \log(n))$
[Slower than method 1]

Print elements in set

```
for(auto it=st.begin();it!=st.end();it++){
    cout<<*it<<" ";
}
```

find() function in set

set -> {1,3,5,6,7,8};

```
auto it = st.find(6);
cout<< *it << endl; // 6
```

If 6 is not present then it = st.end(); NULL;
// *it -> run time error.

Q. Check whether a number x is present or not in the set.

Method - 1: (Using .find())

```
auto it = st.find(x);
if(it==st.end()) cout<<"NOT PRESENT";
else cout<<"PRESENT";
```

Method - 2: (Using .count())

count() returns the number of times an element occurs in the set

```
int cnt = st.count(x);
if(cnt) {
    cout<<"PRESENT";
} else {
    cout<<"NOT PRESENT";
}
```

Map

Map is a special kind of STL container which stores elements as **key-value pair**. No two mapped values can have same key. **All the keys are sorted in ascending order.**

All the keys are unique.

Example: All the Freshers will be having a unique admission number.

“20JE0666” -> Sakshi ;
“20JE0648” -> Saksham ;

“20JE0654” -> Shivali ;

Key -> value;

Syntax of declaration ->

```
map<firstDatatype, secondDatatype> mp;
```

Example:

```
map<string, string> mp;
mp[ "20JE0666"]="Sakshi";
mp[ "20JE0648"]="Saksham";
mp[ "20JE0654"]="Shivali";
```

Note-> Two different keys may have same values.
Like admission number “20JE0888” and “20JE0898” both
can have name as “Yash”

map->[{Sakhi: 20JE0666}, {Saksham: 20JE0648},.....];

Another example:

```
map<int,char> mp1;
mp1[1]='A';
mp1[2]='B';
mp1[3]='C';
```

```
map<char,int> mp2;
mp2[ 'A' ]=1;
mp2[ 'B' ]=2;
mp2[ 'C' ]=3;
```

1. size() function in map:

```
int n = mp.size();
```

Time complexity: O(1)

2. erase() function in map:

```
mp.erase(key);
```

Time complexity: O(logn);

3. count() function in map:

```
mp.count(key);
```

Time complexity: O(logn);

-> mp[key] = value;

If not present that key -> random value.

If case of integer -> default 0;

4. Printing all elements of a map:

The elements of map are a **pair** of key and value. So, you can use .first to access key and .second to access value and iterate through all values, similar to that in a set.
Since, iterators are pointers, use arrow operator to access their members.

The code to print everything inside a map named mp would be like:

```
for (it = mp.begin(); it != mp.end(); it++) {  
cout<<( it->first )<< ' ' <<( it->second )<< '\n';  
}
```

For more functions of map, refer:

<https://www.cplusplus.com/reference/map/map/>

Question1->

Print this pattern using for loop:

```
1  
1,2,  
1,2,3,  
1,2,3,4  
1,2,3,4,5
```

Approach: (Use a variable i for row)

i=1 1
i=2 1,2,
i=3 1,2,3,
i=4 1,2,3,4
i=5 1,2,3,4,5

```
for(int i=1;i<=n;i++){
    for(int j=1;j<=i;j++) cout<<j<<" ";
    cout<<endl;
}
```

Question2

<https://www.hackerrank.com/challenges/marcs-cakewalk/problem>

Approach:-

1, 2, 4, 8, 16..... [constant value]
A₁ A₂ A₃ A₄ A₅..... [any particular arrangement]

minimise this expression(A₁+2*A₂+4*A₃+8*A₄....)

[1,2,3] ascending order ----> 17 (maximum)
[2,1,3] any random order ----> 16 (in between)
[3,2,1] descending order ----> 11 (minimum)

```
int n=calorie.size();
long ans=0;
sort(calorie.begin(),calorie.end());
reverse(calorie.begin(),calorie.end());
for(int i=0;i<n;i++) {
    long currValue =
(long)pow(2,i)*calorie[i];
    ans+=currValue;
}
return ans;
```

Note:

When I was using “**int currValue**” then there was a problem of **overflow** so always look at the **worst case** value and here when I changed it to “**long currValue**” then **all the test cases got passed**.

ASCII Value using “typecasting”

When you convert a character of string into int, it will get converted to its ASCII code.

Each character has a unique ASCII code.

Like '0' has ASCII code 48

'1' => 49

'2' => 50

...

...
'9' => 57

'a' => 97
'b' => 98

...

...

'z' => 122

'A' => 65
'B' => 66

...

...

'Z' => 90

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

int main(){
    char c='a';
    cout<<c<<endl;
    int value = (int)c;
    cout<<value<<endl;
    return 0;
}
```

Prefix arrays and STL custom comparators

Prefix Array

Q.) You are given an array of n numbers, find the sum between index “l” to “r”.

Arr : {-1, 3, 0, 2, -1, 2, 1, -2}
idx : 0 1 2 3 4 5 6 7
 |.....r

```
int sum=0;
for(int i=l;i<=r;i++) sum+=Arr[i];
cout<<"Sum between l to r is :" <<sum<<endl;
```

Time complexity -> O(n);

Q.) You have to solve the same problem but now there will be “q” number of queries.

$Q_i \rightarrow l, r \text{ sum between } l \text{ to } r.$

$q \rightarrow \text{queries}$

$n \rightarrow \text{number of elements}$

```
for(int query=1;query<=q;query++){
    int l,r;
    cin>>l>>r;
    int sum=0;
    for(int i=l;i<=r;i++) sum+=Arr[i];
    cout<<sum<<endl;
}
```

Time Complexity -> $O(q*n)$

When $q \leq 10^5$ and $n \leq 10^5$ [TLE]

Prefix sum->

Arr : {-1, 3, 0, 2, -1, 2, 1, -2}

idx : 0 1 2 3 4 5 6 7

int Prefix[n];

Prefix[i]-> sum of all elements from start till index i.

Prefix[3] = 4

Prefix[6] = 6

Prefix[2] = 2

Prefix[7] = 4

Sum[l....r] -> Prefix[r]-Prefix[l-1];
 Arr : {-1, 3, 0, 2, -1, 2, 1, -2}
 idx : 0 1 2 3 4 5 6 7
 l.....r
 Prefix[r] -> {0,1,2,3...l-1,l,l+1....r}
 Prefix[l-1]-> {0,1,2,3...l-1}
 Prefix[r]-Prefix[l-1] -> {l,l+1....r}

Prefix[r] -> sum[0....r];
 Prefix[l] -> sum[0.....l];
 Prefix[l-1] -> sum[0.....l-1];

Sum[l....r] = Prefix[r]-Prefix[l-1]

-> How to calculate prefix array?

```

int Prefix[n],suffix[n];
Prefix[0]=arr[0];
suffix[n-1]=arr[n-1];
for(int i=1;i<n;i++)
Prefix[i]=Prefix[i-1]+Arr[i];
for(int i=n-2;i>=0;i--)
suffix[i] = suffix[i+1]+arr[i];
  
```

Prefix[i-1] -> sum from [0.....i-1];

Prefix[i] -> sum from [0.....i]?

= sum[0.....i-1]+Arr[i];

= Prefix[i-1]+Arr[i];

Time Complexity-> O(n);

Solution->

Prefix Array has already been calculated. O(n);

```
for(int query=1;query<=q;query++){ // O(q);
    int l,r;
    cin>>l>>r;
    int sum = 0;
    if(l!=0)
        sum=Prefix[r]-Prefix[l-1]; // O(1);
    else sum=Prefix[r]; // O(1);
    cout<<sum<<endl;
}
```

Time Complexity-> O(max(q,n))=O(n)

Q.) You are given an array of n numbers and q number of queries, in each query you will be given 3 integers “l”, “r”, “x” and do an operation of adding the number x to each element between index “l” to “r”.
After all the q operations what would be the final array?

A[10] = {0,0,0,0,0,0,0,0,0,0}

Q queries -> l r x

l=3,r=7,x=5

1<=n<=10^5 , 1<=q<=10^5

l=2 r= 4 ,x=3

A[10] = {0,0,0,5,0,0,0,-5,0}
= {0,0,3,5,0,-3,0,0,-5,0}
= {0,0,3,8,8,5,5,5,0,0}

```
int n;  
cin>>n;  
int a[n+1]={0};
```

```
int q;
cin>>q;
while(q--){ // -> for(int i=1;i<=q;i++)
    int l,r,x;
    cin>>l>>r>>x;
    a[l]+=x; // l=0, r=n-1;
    a[r+1]-=x;
}
for(int i=1;i<n;i++){
a[i]=a[i]+a[i-1];
}
```

<https://codeforces.com/contest/1473/problem/D>

Comparator function in sort:-

```
bool compare(int p1,int p2)
{
    if(p1>=p2){
        return 1;
    }
    return 0;
}
```

```
int main(){
int n;
cin>>n;
int a[n];
for(int i=0;i<n;i++){
    cin>>a[i];
}
sort(a,a+n,compare);
}
```

Home-Work Problem

Q.) When we sort a vector of pairs then by default it sort according to the first element of the pair but using comparator function could you sort according to the second element?

(1,2),(2,5),(4,3),(2,2)

By Default(By first element of pair) ->

(1,2) (2,2) (2,5) (4,3) // **According to first element**

Sorting by second (By second element of pair) ->

(1,2) (2,2) (4,3) (2,5) // **According to second element**

Problem -> **Define your own comparator function for doing this process. (Home Work)?**

Note regarding last discussed problem-

```
bool com(pair<int,int> p1,pair<int,int> p2)
{
    if(2*p1.first+p1.second>=2*p2.first+p2.second){
        return 1;
    }
    return 0;
}
```

pair<int,int> p[n]; //{{x,y}}

pair<int,pair<int,int> > p;
p.first,p.second.first,p.second.second

vector<pair<int,int> > v

```
int x;
string s=to_string(x);
int y=stoi(s);
```

Number Theory - 1

Properties of Modulus

```
int x = -8%5; // x = -3
```

```
-> -3,-8,-13,2
```

```
x+=m
```

Modulus:

Non-negative remainder when x is divided by m.

It is also said as “x mod m”.

Range of “x mod m” : [0,m-1]

$x \% m$ -> modulus

If x is negative -> modulus = $(x \% m + m) \% m$

Q. N is a given number

Find the factorial N % m

Eg. n=5 m = 20

$5! = 120$

$120 \% 20 = 0;$

```
long long a=1e18,b=2e18;
int m = 1e9+7; // -> 1000000007
int x = (a+b)%m;
cout<<x;
```

a>0,b>0

Properties of Modulus:

$$1. (a+b)\%m = ((a\%m)+(b\%m))\%m$$

2. $(a-b)\%m = ((a\%m)-(b\%m)+m)\%m$
3. $(a*b)\%m = ((a\%m)*(b\%m))\%m$
4. $(a/b)\%m = ((a\%m)*(b^{-1}\%m))\%m$

$m=5, a = 9, b=14;$
 $a\%m = 4, b\%m = 4$
 $m=5, a=8, b=4;$
 $(a-b)\%m = 4$
 $a\%m = 3$
 $b\%m = 4$
 $-1\%m = -1$

Q. Find $(N!) \% m$. Where $(N! = 1*2*3*4*5*.....*n)$

$N \rightarrow$ input

$M = 1e9+7;$

$(n!\%m)?$

$N!\%m = (n*(n-1)*(n-2)*(n-3)....1)\%m = ((n\%m)*((n-1)*(n-2)...1)\%m)\%m$
 $1 \leq n \leq 1000$

```
int ans=1;
for(int i=1;i<=n;i++){
    ans = (ans%m)*(i%m);
    ans%=m;
}
return ans;
```

Q. Find (x^n) . (x raised to the power n or x^n)

Method 1: (Using a for loop)

```
int ans=1;
for(int i=1;i<=n;i++){
```

```
    ans*=x;
}
```

Method 2: (Using a recursive function)

```
int fun(int x,int n) //fun(x,n) -> x^n;
{
    if(n==0){
        return 1;
    }
    return x*fun(x,n-1);
}
```

fun(2,5) -> fun(2,4)-> fun(2,3)-> fun(2,2)->fun(2,1)-> fun(2,0)

Time complexity of both methods is O(n)

Binary Exponentiation

$2^8 = (2^2)^4 = 4^4 = (4^2)^2 = 16^2 = (16^2)^1 = 256 \rightarrow \log n$
 $x^n \rightarrow \log n$
 $2^5 = 2 \cdot 2^4$
 x^n

Q. Find $(x^n) \bmod m$

Method 1: Using a recursive function

```
int binaryExponentiation(int x,int n,int m) // x^n      O(logn)
{
    if(n==0){
        return 1;
    }
    if(n%2==0){
        return binaryExponentiation(((x%m)*(x%m))%m,n/2,m);
    }
}
```

```

    return
((x%m)*binaryExponentiation(((x%m)*(x%m))%m,(n-1)/2,m)%m)%m;
}

```

Iterative (Using a loop) :-

```

int binaryExponentiation(int x,int n,int m)    // O(logn)
{
    int res=1;
    while(n!=0){
        if(n%2==1){
            res = ((res%m)*(x%m))%m;
        }
        x = ((x%m)*(x%m))%m;
        n/=2;
    }
    return res;
}

```

Time complexity of both methods is O(log N) here, which works very fast.

Even, for large numbers like $N=10^{20}$, $\log N$ has a very small value.

Prime Number

Definition

Prime numbers are those numbers that are divisible by only 1 and the number itself. i.e the number of divisors should be 2.

Q. 2,3,4,6,7,8,9

P P N N P N N

Q. Write a C++ code to check whether the given number is prime or not.

```
int n;
cin>>n;
int divisors=0;
for(int i=1;i<=n;i++){
    if(n%i==0) {
        // n is divisible by i
        divisors=divisors+1;
    }
}
if(divisors==2) cout<<"The given number is a prime
number"<<endl;
else cout<<"The given number is not a prime number"<<endl;
```

Time Complexity - O(n)
[This was a slow method]

Important Key Point

Consider a natural number N

If i is a divisor of N.

Then, (N/i) is also a divisor of N.

e.g

N=6;

2 is divisor of N bcz ($6\%2==0$);

$6/2 = 3$ is also divisor of N bcz ($6\%3==0$);

$N \leq 10^{10}$ [Worst case of (N)]

i is divisor of N then (N/i) is also a divisor of N .

$N=12 \rightarrow 1\ 2\ 3\ 4\ 6\ 12$

1 to $\leq \sqrt{N}$

**Property: If you have got a divisor $> \sqrt{N}$,
Then there must be a divisor that is less than
 \sqrt{N}**

$N=a*b;$
 $a \leq \sqrt{N};$
 $b \geq \sqrt{N};$

Fast method to check if n is prime

```
int n;
cin>>n;
int divisors=0;
for(int i=1;i<=sqrt(n);i++){
    if(n%i==0) {
        // n is divisible by i
        int first_Divisor=i;
        int second_Divisor=(N/i);
        if(first_Divisor!=second_Divisor) divisors+=2;
        else divisors++;
    }
}
if(divisors==2) cout<<"The given number is a prime
number"<<endl;
else cout<<"The given number is not a prime number"<<endl;
```

Time complexity of this method: O (sqrt(N))

[Faster than previous method]

sqrt(16)=4

16-> 1 2 4 8 16

i=4 i*i=16

$10^6 \rightarrow n/2 == 5*10^5$

$\text{sqrt}(N) == 10^3$

Some important in-built functions

- 1. pow(n, x) => Finds n^x in O(logn)**
- 2. sqrt(n) => Finds square root on n.**

Caution: If the numbers are small then only use pow() function, otherwise use the Binary Exponentiation method to calculate power.

Number Theory - 2

Important Properties

(a-b)%k = (x-y) Then (a-x)%k = (b-y)%k

Proof (Just for your understanding)

LHS =

$$(a-b) = N*k + (x-y);$$

$$(a-x) = (a-b)*k + (b-y); // After reorder.$$

//Take modulo with k on LHS and RHS

$$(a-x)%k = 0 + (b-y)%k;$$

$$(a-x)%k = (b-y)%k$$

= RHS

GCD(a,b)

Q. Write a C++ code to calculate GCD of two numbers.

GCD-> Greatest common divisors.

12, 16.

12-> 1 2 3 4 6 12 O(Sqrt(m))

16-> 1 2 4 8 16 O(sqrt(n));

GCD(12,16) = 4

First Solution->

1. Calculate all divisors of first number
2. Calculate all divisors of second number.
3. And just find the divisor which is common to both and have max value.

Time Complexity -> $O(\text{Sqrt}(m))^* O(\text{sqrt}(n))=O(\text{sqrt}(m*n))$

Euclidean Algorithm for GCD

$\text{gcd}(a,b) = a$, if ($b==0$)

$\text{gcd}(a,b) = \text{gcd}(b,a\%b)$, if($b!=0$)

```
int GCD(int a,int b){  
    if(b==0) return a;  
    return GCD(b,a%b);  
}
```

Time Complexity -> $O(\log n)$

[Fast method]

LCM (Loweset Common Multiple)

$$\text{LCM}(3,4) = 12$$

$$\text{LCM}(12,16) = 48$$

$$\text{LCM}(3,9) = 9$$

Def. The lowest number which is divisible by both a and b.

$$\text{LCM}(a,b) = a*b/\text{gcd}(a,b); = (a/\text{gcd}(a,b))*b$$

$$5,6 \rightarrow 30 = 5*6$$

$$\max(a,b) \text{ to } a*b$$

$$a*b = \text{gcd}*\text{lcm}$$

$$\text{Lcm} = (a*b)/\text{gcd}$$

$$A,b \rightarrow \text{order of } 10^{10}$$

$$\text{Lcm} = (a/\text{gcd})*b$$

There is also an in-built function for GCD in C++, `__gcd()`.

```
int a,b;
cin>>a>>b;
int gcd = __gcd(a,b);
int lcm = (a/gcd)*b;
int x = __gcd(a,__gcd(b,c));
```

$N \rightarrow \text{sqrt}(n); 1 \leq n \leq 10^{16}$

Q queries are given.

In each query, you are given 1 number x , you have to find whether x is prime or not.

$1 \leq q \leq 1000, 1 \leq x \leq 10^6$

Naive solution $\rightarrow q * \text{sqrt}(x)$

$1 \leq q \leq 10^6, 1 \leq x \leq 10^6$

[Naive solution is very slow, it will give TLE]

So, we use this method called **sieve of Erasthones**:

```
bool isPrime[1000001];
// isPrime[i] = 1 if i is prime
// isPrime[i] = 0 if i is not prime

// numbers=1 2 3 4 5 6 7 8 9 10 11 12 13 14
// isPrime   = 0 1 1 0 1 0 1 0 0 0   1   0   1   0
for(int i=0;i<=1000000;i++){
    isPrime[i]=1;
}
isPrime[1]=0;
isPrime[0]=0;
```

```

for(int i=2;i*i<=1000000;i++){
    if(isPrime[i]==1){
        for(int j=i*i;j<=1000000;j+=i){
            isPrime[j]=0;
        }
    }
}

```

Time complexity: $n/2 + n/3 + n/5 + n/7 \dots = n \log(\log n)$

Multiple of 2 -> 4,6,8,10,12...

Multiple of 4 -> 8,12,16....

Multiple of 3 -> 6,9,12,15....

Multiple of 6 -> 12,18,24...

$j_{\min} = i^2, i^3, i^4, \dots, i^i$

$j_{\max} \leq 1000000$

$j_{\min} \leq j_{\max}$

$i^i \leq 1000000$

$i \leq 1000 = \sqrt{10^6}$

Sieve of Eratosthenes

```

isPrime[1]=0;
isPrime[0]=0;
for(int i=2;i*i<=1000000;i++){
    if(isPrime[i]==1){
        for(int j=i*i;j<=1000000;j+=i){
            isPrime[j]=0;
        }
    }
}

```

Time Complexity -> $n(\log(\log(\sqrt{n})))$

Space complexity -> $O(n)$

Time complexity - $O(q + x \log(\log(\sqrt{x})))$

Smallest Prime Factor(SPF)

spf[i] -> smallest prime number that divides i.

(3,6,8,10)

If z is a prime number, $\text{spf}[z] = z$

1<=q<=10^6, 1<=x<=10^6

Find the $\text{spf}[x]$ for each query?

```
for(int i=0;i<=1e6;i++){
    spf[i] = i;
}
for(int i=2;i*i<=1e6;i++){
    if(spf[i]==i){
        for(int j=i*i;j<=1e6;j+=i){
            if(spf[j]==j){
                spf[j]=i;
            }
        }
    }
}
int n;
cin>>n;
int a[n];
```

```
for(int i=0;i<n;i++){
    cin>>a[i];
}
```

Comparator function in Set

Q. Sort a vector of pair in reverse order using a set.

Number Theory - 3

Q.1)- <https://www.codechef.com/COOK126B/problems/PTUPLES>

-> Naive Approach

(a,b,c)

a,b,c->primes number and $(a+b)=c$;

1. Except 2 all the prime numbers are odd.
2. If all a,b,c are odd prime.

Then for $(a+b=c)$ (odd+odd=(even)=odd) not Possible.

That's Why we must have to put one even-prime-number on LHS.

And fortunately we have only one even-prime-number and which is 2.

$(2+b)=c$; $b=(c-2)$;

i.e $a=2$;

```
#include <bits/stdc++.h>
using namespace std;
int main(){

    const int MAX=1000000;
    bool is_Prime[MAX+1];
    //memset(is_prime,true,sizeof(is_prime));
    for(int i=0;i<=MAX;i++) is_Prime[i]=true;
    is_Prime[0]=false;
    is_Prime[1]=false;
```

```

for(int i=2;i*i<=MAX;i++){
    if(is_Prime[i]==true){
        for(int j=i*i;j<=MAX;j+=i) is_Prime[j]=false;
    }
}
//O(Nlog(logN))
int t;
cin>>t;
for(int i=1;i<=t;i++){ //O(t*n)
    int n;
    cin>>n;
    if(n<=4){
        cout<<"0"<<endl;
    }
    else{
        int ans=0;
        for(int c=5;c<=n;c++){ //O(N)
            if(is_Prime[c]==true){
                int b=c-2;
                if(is_Prime[b]==true) ans++;
            }
        }
        cout<<ans<<endl;
    }
}
return 0;
}

```

// Overall Time complexity - O(T*N);

This is a slow Solution

-> Points to Optimize the code.

Tuples[i]-> all possible tuples using numbers (1,2,3,4,5....i);

Points-> 1,2,3,4,5,6,7,8....n-1,n;

Tuple[n]=Tuples[n-1]+(the tuple form by using n);

5=>(2,3,5);

6=> (2,3,5)+(is it possible to get one tuple using 6?);

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

int main(){

    const int MAX=1000000;
    bool is_Prime[MAX+1];
    //memset(is_prime,true,sizeof(is_prime));
    for(int i=0;i<=MAX;i++) is_Prime[i]=true;
    is_Prime[0]=false;
    is_Prime[1]=false;
    for(int i=2;i*i<=MAX;i++){
        if(is_Prime[i]==true){
            for(int j=i*i;j<=MAX;j+=i)
                is_Prime[j]=false;
        }
    }
}
```

```

        }
    }
//O(Nlog(logN))

vector<int> Tuples(MAX+1);
Tuples[0]=Tuples[1]=Tuples[2]=Tuples[3]=0;
for(int c=4;c<=MAX;c++){ //O(N);
    int b=c-2;
    Tuples[c]=Tuples[c-1];
    if(is_Prime[b]==true && is_Prime[c]==true){
Tuples[c]++;
}
}

int t;
cin>>t;
for(int i=1;i<=t;i++){ //O(t)
    int n;
    cin>>n;
    cout<<Tuples[n]<<endl; //O(1);
}
return 0;
}
//Overall time complexity = O(Nlog(logN))

```

Fast Solution (Optimized Code)

Q.2)-

<https://www.hackerrank.com/challenges/minimum-distances/problem>

```
#include<bits/stdc++.h> // O(n*n)
#define int long long
using namespace std;
int32_t main()
{
    int n;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    int ans = n+1;
    for(int i=0;i<n;i++){
        for(int j=i+1;j<n;j++){
            if(a[i]==a[j]){
                ans = min(ans,j-i);
            }
        }
    }
}
```

```

if(ans==n+1){
    cout<<-1;
}else{
    cout<<ans;
}

```

```

#include<bits/stdc++.h> // O(n)
#define int long long
using namespace std;
int32_t main()
{
    int n; // 1<=n<=10^6, 1<=a[i]<=10^5
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    int ans = n+1;
    int m[100001]={-1}; // m[i] is the index of value i
encountered so far
    for(int i=0;i<n;i++){
        if(m[a[i]]==-1){ // this a[i] is the first value
encountered
            m[a[i]]=i;
        }else{
            ans = min(ans,i-m[a[i]]);
            m[a[i]]=i;
        }
    }
    cout<<ans;
}

```

```

        }
    }
    if(ans==n+1){
        cout<<-1;
    }else{
        cout<<ans;
    }
}
/*
    current element -> ith index
    a[i] from 0th index to (i-1)th index or not
    [1,2,3,4,3,3]
*/

```

Q.3)-

<https://codeforces.com/problemset/problem/230/B>

We know that prime numbers are positive integers that have exactly two distinct positive divisors.

Similarly, we'll call a positive integer t T-prime, if t has exactly three distinct positive divisors.

You are given an array of n positive integers. For each of them determine whether it is T-prime or not.

The first line contains a single positive integer, n ($1 \leq n \leq 10^5$), showing how many numbers are in the array. The next line contains n space-separated integers x_i ($1 \leq x_i \leq 10^{12}$).

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is advised to use the `cin, cout` streams or the `%I64d` specifier.

Eg 4-> 1,2,4

9 -> 1,3,9

16-> 1,2,4,8,16

25-> 1,5,25

Sol:-

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
bool prime[1000001];
int32_t main()
{
    for(int i=2;i<=1e6;i++){
        prime[i]=1;
    }
    for(int i=2;i*i<=1e6;i++){
        if(prime[i]){
            for(int j=i*i;j<=1e6;j+=i){
                prime[j]=0;
            }
        }
    }
    map<int,int> m;
    for(int i=2;i<=1e6;i++){
        if(prime[i]){
            int x = i*i;
            m[x]=1; // (x,1) such that x is t-prime
        }
    }
    int n;
    cin>>n;
```

```
int a[n];
for(int i=0;i<n;i++){
    cin>>a[i];
}
for(int i=0;i<n;i++){
    if(m[a[i]]==1){
        cout<<"YES\n";
    }else{
        cout<<"NO\n";
    }
}
```

Number Theory - 4

Q.1)- You will be given a number N, find the sum of its all prime divisors. ($N \leq 10^6$)

12-> 1,2,3,4,6,12

Sum of all prime divisors = $(2+3)=5$;

Find its all of its divisors and then check how many of them are prime number.

$O(N\log(\log(N))) + O(N) \Rightarrow O(N\log(\log(N)))$

Q.2)- Now you have to solve the same above problem but for q number of queries. ($q \leq 10^6$)

Time Complexity of Native solution-> $O(q * N\log(\log(N)))$

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

int main(){

    const int MAX=1000000;
    bool is_prime[MAX+1];
    int Sum[MAX+1];
    // Sum[i]-> sum of its all prime divisors.
    memset(is_prime,true,sizeof(is_prime));
```

```

is_prime[0]=is_prime[1]=false;

for(int i=2;i<=MAX;i++){
    if(is_prime[i]==true){
        for(int j=i;j<=MAX;j+=i){
            if(j>i) is_prime[j]=false;
            sum[j]+=i;
        }
    }
}

//O(NlogN)
i=2(Prime);
2,4,6,8,10.....
Maked all these numbers as Non prime number
and Sum[j]+=i;
int q;
cin>>q;
while(q--){ //O(q)
    int n;
    cin>>n;
    cout<<Sum[n]<<endl;
}

//OverAll timeComplexity = O(N(logN))
return 0;
}

//log(sqrt(N)) = ½ log(N)

```

Q.3)- You will be given a number N, find the number of its divisors. ($N \leq 10^6$)

```
Int count=0;
for(int i=1;i*i<=N;i++){
    if(N%i==0){
        Int first_divisor = i;
        Int second_divisor = (N/i);
        if(first_divisor!=second_divisor) count+=2;
        Else count++;
    }
}
cout<<count<<endl;
Time Complexity -> O(sqrt(N))
```

Q.4)- Now you have to solve the same above problem but for q number of queries. ($q \leq 10^6$)

Brute Force Time Complexity -> $O(q * \sqrt{N}) == 10^9$

Points->

$N \rightarrow z_1^{k_1} * z_2^{k_2} * z_3^{k_3} * \dots$

$z_i \rightarrow$ prime number.

Number of divisors = $(k_1+1)*(k_2+1)*(k_3+1)\dots$

Bool is_prime[MAX+1];

Int SPF[MAX+1];

```

Bool is_prime[MAX+1];
Int SPF[MAX+1];
// NLog(logN))
Int q;
cin>>q;
while(q--){
    Int n;
    cin>>n;
    Int ans=1; (12)
    while(n>1){
        Int k=0;
        Int spf = SPF[n];
        while(n%spf==0){
            n/=spf;
            k++;
        }
        ans=ans*(k+1);
    }
    cout<<ans<<endl;
}

```

Time complexity -> O(qlogn);

$N = 2^{20}$
 $Spf = SPF[N] = 2;$
 $while(N \% spf == 0) n /= 2;$
 It will run 20;
 $\text{Log}_2(N);$

$N = 2^{10} * 3^{10}$

Q. <https://codeforces.com/problemset/problem/1360/D>

Sol:-

Number of packages*Number of shovels in that package = n
Number of packages = $n / (\text{number of shovels in that package})$
->Number of shovels in that package must divide n.
-> number of shovels is a divisor of n.
-> number of shovels $\leq k$
→ the max divisor of n $\leq k$. -> ans

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
int32_t main()
{
    int t;
    cin>>t;
    while(t--){ //for(int i=0;i<t;i++)
        int n,k;
        cin>>n>>k;
        vector<int> divisors;
        for(int i=1;i*i<=n;i++){
            if(n%i==0){
                divisors.push_back(i);
                divisors.push_back(n/i);
            }
        }
        int ans=n;
        for(int i=0;i<divisors.size();i++){
            if(divisors[i]<=k){
```

```

        ans = min(ans,n/divisors[i]);
    }
}
cout<<ans<<endl;
}
}

```

Q: <https://codeforces.com/problemset/problem/776/B>

n=3 -> jewelery -> 1,2,3

 Price -> 2,3,4

 2 colors -> 1,1,2

 1,2,2

n=7

Prices -> 2,3,4,5,6,7,8

 -> 1,1,2,1,2,1,2

n=2 -> 2,3

n=1,2 -> number of color = 1

Else number of color =2

n=12 -> 2,3,4,5,6,7,8,9,10,11,12,13

 -> 1,1,2,1,2,1,2,2,2,1,2,1

Q. <https://codeforces.com/problemset/problem/1108/B>

Eg 20 8

Divisors of 20 -> 1,2,4,5,10,20

Divisors of 8 -> 1,2,4,8

10 2 8 1 2 4 1 20 4 5

20 -> 1,2,4,5,10,20

Euler Totient Function:-

$\phi(n)$ = count of numbers from 1 to n that are coprime with n.

Coprime:- 2 numbers x and y are coprime if $\gcd(x,y)=1$.

$\phi(2)=1$ (1)

$\phi(4)=2$ (1,3)

$\phi(8)=4$ (1,3,5,7)

Number p -> prime number

$\phi(p) \rightarrow 1 \dots p-1 = p-1$

$\phi(p^2) \rightarrow p^2-p$

9 -> $3^2 \rightarrow 3,6,9 = 9/3 = p^2/p$

$\phi(p^k) \rightarrow p^k \rightarrow p, 2*p, 3*p, 4*p \dots p^k \rightarrow p^{(k-1)}$

$p^k = p + (n-1)*p$

$p^{(k-1)} = 1 + n - 1$

$N = p^{(k-1)}$

$\phi(p^k) = p^k - p^{(k-1)}$

$N = (p_1^{k_1} \cdot p_2^{k_2} \dots)$

Where p_1, p_2, \dots are prime numbers.

$\phi(a*b) = \phi(a) \cdot \phi(b)$ if a and b are coprime.

$\phi(n) = \phi(p_1^{k_1} \cdot p_2^{k_2} \dots)$

$\Phi(n) = \phi(p_1^{k_1}) \cdot \phi(p_2^{k_2}) \dots$

$$\begin{aligned}
 \phi(n) &= (p_1^{k_1} - p_1^{k_1-1})^*(p_2^{k_2} - p_2^{k_2-1}) \dots \\
 &= \{p_1^{k_1} \cdot p_2^{k_2} \dots\} (1 - 1/p_1)(1 - 1/p_2) \dots \\
 &= n^* (1 - 1/p_1)^* (1 - 1/p_2) \dots \\
 &= n * \prod \{1 - 1/p_i\}
 \end{aligned}$$

```

Int phi[1000001];
for(int i=0;i<=1e6;i++){
    phi[i]=i;
}
for(int i=2;i<=1e6;i++){
    if(phi[i]==i){
        for(int j=i;j<=1e6;j+=i){
            Phi[j] = phi[j]-phi[j]/i;
        }
    }
}

```


Number Theory - 5

$$(a+b)\%m = ((a\%m)+(b\%m))\%m;$$
$$(a/b)\%m = ((a\%m)*((b^{-1})\%m))\%m;$$

Today's goal is to calculate $(b^{-1})\%m$.

-> **Modulo inverse of b.**

Modulo Multiplicative Inverse.

$$(a*x)\%n = 1$$

Then it is known as the modulo inverse of a w.r.t n = $x = (a^{-1})$.

And $1 \leq x \leq (n-1)$

And it will exist only when $\gcd(a,n)=1$;

When ever you found $\gcd(a,n)=1$ then you may write it like
 $a*x+n*y=1$;

Here

x -> modulo inverse of a w.r.t n = $(a^{-1})\%n$;

Take modulo w.r.t n on LHS and RHS then

$$(a*x)\%n = 1$$

y -> modulo inverse of n w.r.t a;

Take modulo w.r.t a on LHS and RHS then

$$(n*y)\%a = 1$$

Important Property

$$(a^{\text{ETF}(p)}) \% p = 1 \text{ (always)}$$

Where a and p are coprime i.e $\text{gcd}(a,p)=1$;

If p is a prime number then $\text{ETF}(p) = p-1$;

$$(a^{(p-1)}) \% p = 1 \text{ (Fermat's Little Theorem)}$$

$$(a * (a^{(p-2)})^*) \% p = 1;$$

$$\rightarrow (a^x \% p = 1 \text{ where } x = a^{p-1})$$

$$\text{So } x = a^{(p-2)}$$

$$[(a^{-1}) \% p = (a^{(p-2)}) \% p][$$

-> Final outcome->

modulo inverse of a w.r.t m($(a^{-1}) \% m$) is equal to $((a^{(m-2)}) \% m$; [m is prime number].

$$x = (a^{-1}) \% m; 1 \leq x \leq (m-1)$$

Last property-> $(a^z \% m = (a^{(z \% \text{ETF}(m)) \% m})$;

Note-> In 99% of the cases you would find m as a prime number.

Find modulo with $m = 10^{10} + 7$; (prime number);

**Q.1) Find $(nCr \% m) = \text{fac}[n]/(\text{fac}[r]*\text{fac}[n-r])$;
Where $m = 1e9+7$ (prime number);**

```
const int max=1e6;
vector<long> fac(max+1);
fac[0]=1;
for(int i=1;i<=max;i++) fac[i]=fac[i-1]*i%m;
```

Using above knowledge find $(nCr \% m)$;

```
nCr = fac[n]/(fac[r]*fac[n-r]);
      fac[n]*(fac[r]^(m-2))*(fac[n-r]^(m-2))%m;
```

$m \Rightarrow$ prime number

-> Function to calculate the modulo inverse

```
long ModuloInverse(long a,long m){
    //((a^(m-2))%m;
    //using binary exponentiation calculate value of
    (a^(m-2))%m and return ans;
    Long ans = (a^(m-2))%m;
```

Return ans;

}

Q.2) <https://codeforces.com/problemset/problem/300/C>

a=2,b=3,n=10;

2222222333 -> 23

Total n digits -> i of them are a

-> n-i digits would be b

Sum = $a^i \cdot (n-i)^{n-i}$

Total numbers that can be formed using i a's and (n-i) b's

= nC_i

Sol:-

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
int fac[1000001];
int rem = 1e9+7;
void pre()
{
    fac[0]=1;
    for(int i=1;i<=1e6;i++){
        fac[i] = fac[i-1]*i; // (a*b)%m = ((a%m)*(b%m))%m
        fac[i]%=rem;
    }
}
int binExp(int x,int n)
{
    int res=1;
```

```

while(n){
    if(n%2==1){
        res*=x;
        res%=rem;
    }
    n/=2;
    x*=x;
    x%=rem;
}
return res;
}

int ncr(int n,int r)
{
    int temp1 = fac[n];
    int temp2 = fac[n-r]*fac[r];
    temp2%=rem;
    int temp3 = binExp(temp2,rem-2); // temp3 is the
inverse
    temp1*=temp3;
    temp1%=rem;
    return temp1;
}

bool check(int sum,int a,int b) // return 1 if sum is a good
number else it returns 0;
{
    for(int i=sum;i>0;i/=10){ //645 -> 64 -> 6-> 0 and 6 and
4 and 5

```

```
int r = i%10;
if(r!=a&&r!=b){
    return 0; // number is not good
}
return 1; // number is good
}
int32_t main()
{
    int a,b,n;
    cin>>a>>b>>n;
    pre();
    int ans=0;
    for(int i=0;i<=n;i++){
        int sum = a*i+(n-i)*b;
        if(check(sum,a,b)==1){
            //add nci to ans
            ans+= ncr(n,i);
            ans%=rem;
        }
    }
    cout<<ans;
}
```

$$N \rightarrow (n-3) + (n-4) + (n-5) + \dots + 3$$

$$N-1 \rightarrow (n-4) + (n-5) + \dots + 3$$

$$n \rightarrow (n-1) + (n-3)$$

Bit Manipulation

Bitwise operators and their application

Note: Time complexity of all the bitwise operator are approx O(1)

& => bitwise AND
| => bitwise OR
^ => bitwise XOR
>> => bitwise Right shift
<< => bitwise Left shift
~ => bitwise NOT

Note that & (bitwise AND) is different from && (logical AND) , | (bitwise OR) is different from || (logical OR) and ~ (bitwise NOT) is different from ! (logical NOT)

Rules->

AND

0&0=0
0&1=0
1&0=0
1&1=1

OR

0&0=0
0&1=1
1&0=1
1&1=1

XOR

0&0=0
0&1=1

$$\begin{aligned}1 \& 0 = 1 \\1 \& 1 = 0\end{aligned}$$

Properties of XOR:

$$\begin{aligned}a \wedge a &= 0 \\0 \wedge a &= a\end{aligned}$$

NOT

$$\begin{aligned}\sim 0 &= 1 \\\sim 1 &= 0\end{aligned}$$

```
int a=5 (101), b=7 (111);
cout << a&b ; // (101) & (111)=(101)=5
cout << a|b ; // (101) | (111)=(111)=7
cout << a^b ; // (101) ^ (111)=(010)=2
cout << ~a ; // ~(101)=(010)=2
```

Left shift operator(<<)

$$\begin{aligned}a &\rightarrow 0000001110 \\a<<1 &\rightarrow 000011100\end{aligned}$$

Right shift operator(>>)

$$\begin{aligned}a &\rightarrow 0000001110 \\a>>1 &\rightarrow 0000000111\end{aligned}$$

E.g int a=5;
 a=a<<1; // then a? = 10 (2*a);
 a=5;
 a=a>>1; // then a? = 2 (a/2) e.g (5/2)=2;

Sol

$$\begin{aligned}a &= 5(00101); \\a << 1 &\rightarrow (01010) = 10;\end{aligned}$$

Q.) You are given a number and find the value of (a<<b)?

Ans: $a * (2^b)$

Q.) You are given a number and find the value of (a>>b)?

Ans: $a/(2^b)$

Why? $2^3 \quad 2^2 \quad 2^1 \quad 2^0 \quad (2^2+2^0=5)$

$a=5 \Rightarrow \begin{matrix} 0 \\ 1 \\ 0 \end{matrix}$

$a<<1 \Rightarrow \begin{matrix} 1 \\ 0 \\ 1 \\ 0 \end{matrix}$

$$(2^3+2^1)=2(2^2+2^0)=2*a;$$

Thus, $(1<<n)$ is equivalent to 2^n

Q.) You are given an array of N numbers in which all the numbers are repeated twice except one number which is present exactly once then find out that number?

Link: <https://www.hackerrank.com/challenges/lonely-integer/problem>

e.g-> {2,3,4,4,2} so Answer=3;

Hint-> use XOR property

$$a \wedge a = 0;$$

$$a=5 \Rightarrow (101) \text{ then } a \wedge a \quad (101) \wedge (101) = (000) = 0;$$

If we use xor of all the numbers present in the array i.e

$$2^3 \wedge 4 \wedge 4 \wedge 2 = 3$$

If And-> $(1&2&4&4&2) \Rightarrow (1&2&3&4)$ No Need.

```
int main()
{
int n;
cin>>n;
vector<int> arr(n);
for(int i=0;i<n;i++) cin>>arr[i];
int ans = 0;
for(int i=0;i<n;i++){
ans=ans^arr[i];
```

```
 }  
cout<<ans;  
}
```

Q.) You have to check whether the given number is odd or even but you are not allowed to use % operator then how do you do that?

a=5-> (0101)
b=10-> (1010)
a&1=>(0101)&(0001)=(0001)=1
b&1=>(1010)&(0001)=(0000)=0

```
if((a&1)==1){  
    cout<<"ODD"<<endl;  
}  
  
else {  
    cout<<"EVEN"<<endl;  
}
```

Q.) How to check whether i^{th} bit (from right) is 1 or 0 for the given input number n.

001010101010
....43210 (index)
0->0
1->1
 $(1 << i) \rightarrow 0000000100000000$
j

Answer:

```
if(n&(1<<i))  
{  
    cout<<"i-th bit is set"\  
}  
else
```

```
{  
    cout<<"i-th bit is not set" << endl;  
}
```

Q.) How to set the ith bit to 1 for the given input number n;

Ans:

$n = n | (1 \ll i);$

Q.) How to set the ith bit to 0 for the given input number n;

Ans:

$n = n \& (\sim(1 \ll i))$

a-> 0000010101010100000
 i
(1<<i) 0000000100000000000
~(1<<i) 11111111011111111111

Q.) How to calculate the number of setbits(1) in the given number.

$0 \leq n \leq 2^{63}-1$

Sol:-

```
long long n; // 0 <= n <= 2^63-1  
cin>>n;  
long long ans=0;  
// ans stores the number of set bits  
for(int i=0;i<64;i++){  
// 1<<i -> int  
// 1LL<<i -> long long int  
if(n&(1LL<<i)){  
    ans++;  
}  
}  
cout<<ans;
```

Q.) How to swap two numbers using the XOR operator.

Sol: $(x \wedge y) \wedge y \rightarrow x$

$(x \wedge y) \wedge x \rightarrow y$

$x = x \wedge y; // x = x \wedge y, y = y$

$y = x \wedge y; // y = (x \wedge y) \wedge y \rightarrow x // x = x \wedge y, y = x$

$x = x \wedge y; // x \wedge y \wedge x = y // x = y, y = x$

Q.) How to generate all possible non empty subsequences of the given string.

e.g-> (abc) has following subsequences:

a,b,c,ab,bc,ac,abc

Subsequence -> delete some elements from anywhere in the string and concatenate the remaining.

Eg abcdefgh -> del c,f,h -> abdeg

Substring -> delete some elements from the beginning and some from the end.

Eg abcdefgh -> del a,b from begin and g,h from end

-> cdef

Sol :-

Assume that length of string is n.

Represent any subsequence of this string as a binary number of length n.

Eg abc -> binary number of length 3

If binary digit is 1 -> then that char is present in this sequence.

Else it is deleted.

abc

110 -> ab 1 to 7 -> 001,010,011,100,101,110,111

-> c, b , bc, a ,ac ,ab , abc

101-> ac

N -> 1 to $2^n - 1$

```
vector<string> seq;
// stores all non empty subsequences
string s;
cin>>s;
```

```

int n = s.length();
//n<=15
for(int i=1;i<(1<<n);i++){ // 1 to 2^n-1
    string temp="";
    for(int j=0;j<n;j++){
        if(i&(1<<j)){ // if jth bit in i is set
            temp+=s[j];
        }
    }
    seq.push_back(temp);
}
for(int i=0;i<seq.size();i++){
    cout<<seq[i]<<" ";
}

```

Q. <https://www.hackerrank.com/challenges/and-product/problem>

N queries -> a and b

Output a&(a+1)&(a+2)...b

12 and 15 -> 1100 and 1111 -> 1100 -> 12

14 15 -> 1110 and 1111 -> 1110 -> 14

1001 and 1101 -> 1000 -> 8

1001&1010&1011&1100&1101 -> 1000 -> 8

01111 and 10000 -> 0

```

string x;
for(int i=31;i>=0;i--){
    if(a&(1<<i)){
        x+='1';
    }else{
        x+='0';
    }
}

```

```

while(a){
    int x = a%2;
}

```

```
    x+=( '0'+x);
    x/=2;
}
reverse(x.begin(),x.end());
```

Q. <https://www.hackerrank.com/challenges/sansa-and-xor/problem>

3 4 5

3

4

5

3,4

4,5

3,4,5

Answer=> $(3 \wedge 4 \wedge 5) \wedge (3^4 \wedge 4^5 \wedge 3^{4^5})$

-> If occurrence is even then don't take it.

$$4 \rightarrow a \wedge a \wedge a \wedge a = 0$$

-> If occurrence is odd then include it in your answer.

$$5 \rightarrow a \wedge a \wedge a \wedge a \wedge a = a$$

```
int sansaXor(vector<int> arr) {
    int ans=0;
    int n=arr.size();
    for(int i=0;i<n;i++){
        long occ=(i+1)*(n-i);
        if(occ&1) ans^=arr[i];
    }
    return ans;
}
```

Binary Search

Q. Given an array arr of n integers and an integer x.

Write C++ program to find any position i such that arr[i] = x. If x is not present in the array, i = -1. Print i.

For eg:

Input:

arr=[2, 5, 1, 4, 7], x=4

Output:

3

```
int pos=-1;
for(int i=0; i<n; i++)
{
    if(arr[i]==x)
    {
        pos=i;
        break;
    }
}
cout<<pos;
```

This is called **linear search**.

Time complexity: O(n) [Worst case time complexity]

Q. Given an array arr of n integers and an integer x. Array arr is sorted in non-decreasing i.e $\text{arr}[i] \leq \text{arr}[i+1]$.

Write C++ program to find any position i such that $\text{arr}[i] = x$. If x is not present in the array, i = -1. Print i.

$$\text{mid} = (0+9)/2 = 4$$

$$\text{mid} = (5+9)/2 = 7$$

Find a position i such that $\text{arr}[i] = 59$

$$\text{mid} = (\text{lo} + \text{hi})/2$$

1	3	4	7	12	31	44	59	71
0	1	2	3	4	5	6	7	8

lo ↑
 hi ↑

$$(0+3)/2 = 1$$

$$(2+3)/2 = 2$$

Find a position i such that $\text{arr}[i] = 8$

$$\text{mid} = (\text{lo} + \text{hi})/2$$

1	3	4	7	12	31	44	59	59	71
0	1	2	3	4	5	6	7	8	9

hi ↑
 lo ↑

Code:

```
int lo=0, hi=n-1;
int mid;
int pos=-1;
while(lo <= hi)
{
    mid = (lo+hi)/2;
    if(arr[mid]==x)
    {
        pos=mid;
        break;
    }
    else if(arr[mid]<x)
    {
        lo = mid +1;
    }
    else {
        hi = mid - 1;
    }
}
cout<<pos;
```

This is called **binary search**.

Time complexity: $O(\log N)$ [At every step, active region of search reduces by half]

$n \Rightarrow n/2$

$n/2 \Rightarrow n/4$

$n/4 \Rightarrow n/8$

.....

1

$n/(2^i) = 1$

It will be $O(\log N)$

To calculate the middle element, the above formula $\text{mid} = (\text{lo}+\text{hi})/2$, then overflow may occur.

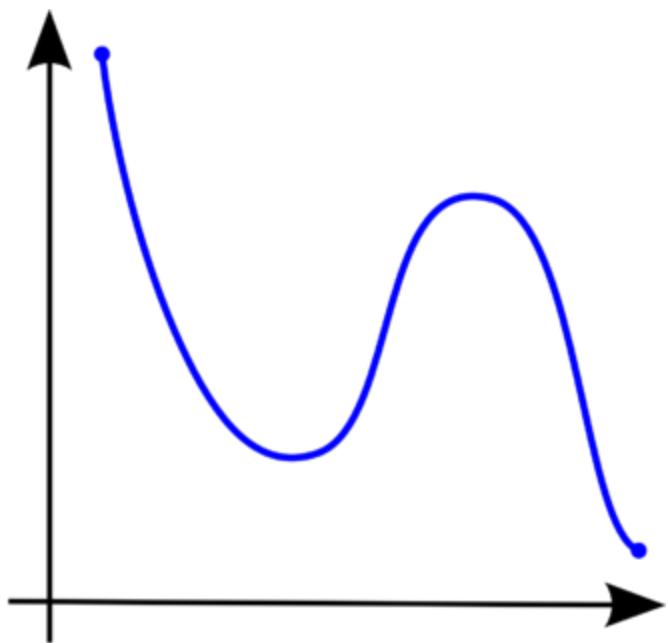
So, you should always use this formula to calculate middle element:

$\text{mid} = \text{lo} + (\text{hi}-\text{lo})/2;$

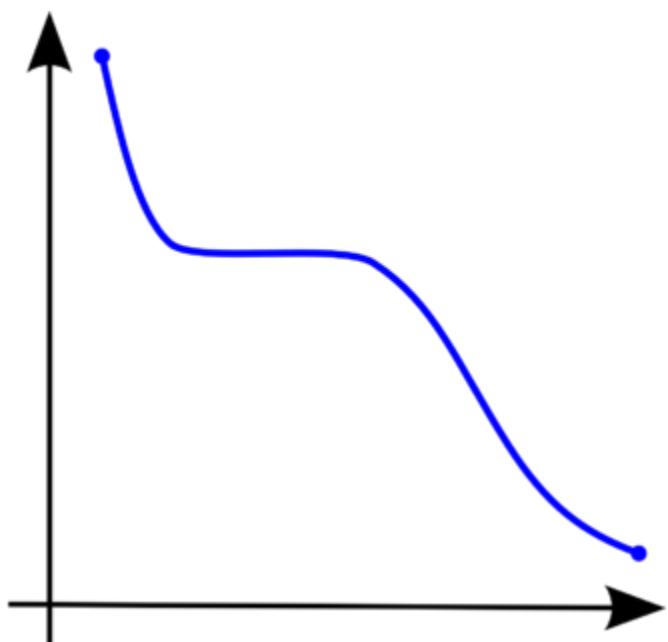
or

$\text{mid} = \text{lo} + (\text{hi}-\text{lo}+1)/2;$

Q. Which of the following is a monotonic function?



Function a



Function b

Function b is a monotonic function because in the whole interval, it is non-increasing

Function a is not monotonic because it first decreases, then increases and

then again decreases.

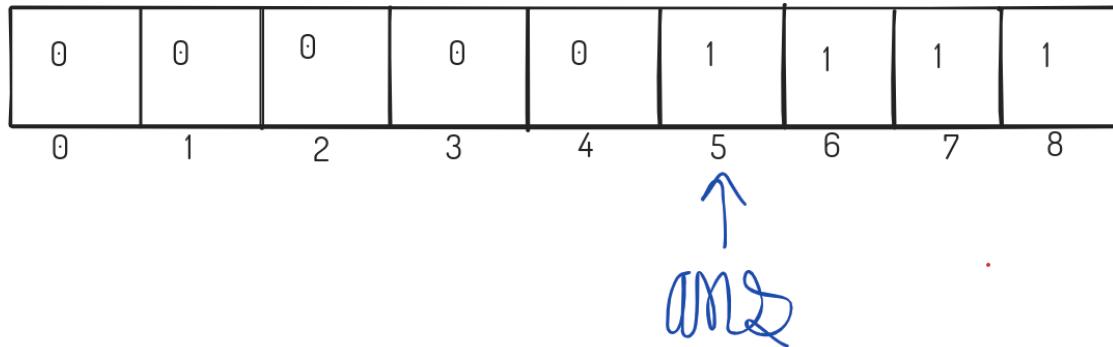
Note: Binary search can always be applied only on monotonic functions. [IMPORTANT]

Q. Given an array arr of n integers consisting of 0's and 1's and all the 1's are after all the 0's, find the position of the first 1.

The array elements are monotonically non-decreasing, so we can apply binary search here.

Eg:

Q. Given an array arr of n integers consisting of 0's and 1's and all the 1's are after all the 0's.
Find the position of the first 1.



Code:

```
int lo=0,hi=n-1;
int mid;
int ans=-1;
while(lo<=hi)
{
    mid = lo + (hi-lo)/2;
    if (arr[mid] == 0)
```

```
{  
    lo = mid+1;  
}  
else  
{  
    ans = mid;  
    hi = mid-1;  
}  
}  
cout<<ans;
```

Q.

<https://codeforces.com/edu/course/2/lesson/6/1/practice/contest/283911/problem/C>

Function : $\text{arr}[i] \geq x$ (Monotonic Function, since array is sorted)

0 0 0 0 1 1
3 3 5 8 9 12

x = 6

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

using namespace std;

int main() {
    int n, k;
    cin >> n >> k;

    vector<int> vec(n);

    for (int i = 0; i < n; i++) {
        cin >> vec[i];
    }

    int x;
```

```
while (k--) {
    cin >> x;
    int lo = 0, hi = n - 1;
    int mid;
    int ans = n;
    while (lo <= hi) {
        mid = lo + (hi - lo) / 2;
        if (vec[mid] < x) {
            lo = mid + 1;
        } else {
            ans = mid;
            hi = mid - 1;
        }
    }
    cout << ans + 1 << '\n';
}

return 0;
}
```

HW: (Solve the 4 problems given here)

<https://codeforces.com/edu/course/2/lesson/6/1/practice>

Q: <https://www.spoj.com/problems/EKO/>

N trees in a row: h₁, h₂, ..., h_n

A woodcutter needs m length of wood

A[1...n] -> heights of trees

M

h -> height of the sawblade.

Approach:

Length of wood that mirka gets res = sum (max(0, a[i]-h)) , 1 <= i <= n

If (res < m) -> h+1, h+2..... Are not possible

Else -> h can be your ans 1...h-1 are not required

l=0; -> cut all the trees completely

r=1e9; // log(10^9) = 9log10

Sol: n <= 10⁹

log(n) < log(1e9)

n log(n) < n log(1e9)

```
int l=0, r=1e9, ans=0; // n log(1e9)  n<=10^6  logn<log(1e9)

while(l<=r){
    int mid = l+(r-l)/2;
    int res=0;
    for(int i=0; i<n; i++){
        if(a[i]>mid){
```

```

        res+=a[i]-mid;
    }
}
if(res>=m){ //mid is a possible ans
    ans=mid;
    l=mid+1;
}else{
    r=mid-1;
}
cout<<ans;

```

You can also think of the problem as :

// Let us say height of sawblade = x
// Target : m metres of wood
// $f(x) = 1$, when you can get m metres of wood using a sawblade
// of height x and 0, otherwise

x: 0 1 2 ... 7 8 ... 1e9

$f(x)$: 1 1 1... 0 0 0 [Monotonically non-increasing]

We need to find the last value of x for which $f(x)=1$.

In case of binary search over floating point (decimal) values:

```

double lo, hi;
const double eps = 0.000001; // or 1e-6;
while ( (hi - lo) > eps)
{
    .....

```

}

More practice problems:

1. Solve all the problems here:

<https://codeforces.com/edu/course/2/lesson/6/2/practice>

2. <https://www.spoj.com/problems/AGGRCOW/>

3. <https://codeforces.com/problemset/problem/1195/B>

4. <https://codeforces.com/problemset/problem/1119/B>

5. <https://www.spoj.com/problems/NOTATRI/>

6. https://atcoder.jp/contests/abc174/tasks/abc174_e

7. <https://www.spoj.com/problems/PIE/>

8. <https://www.spoj.com/problems/HACKRNDM/>

9. <https://codeforces.com/problemset/problem/760/B>

10. <https://www.spoj.com/problems/BOOKS1/>

Binary Search Problems

Q: <https://codeforces.com/contest/1486/problem/C1>

There is an array a of n **different** numbers. In one query you can ask the position of the second maximum element in a subsegment $a[l..r]$. Find the position of the maximum element in the array in no more than **40** queries.

A subsegment $a[l..r]$ is all the elements a_l, a_{l+1}, \dots, a_r . After asking this subsegment you will be given the position of the second maximum from this subsegment **in the whole array**.

$a = [5, 1, 4, 2, 3]$

$l=1, r=5 \rightarrow 3$

$l=3, r=5 \rightarrow 5$

In interactive problems:

Judge

(Knows the array)
int x;
cin>>x;

You

(can ask queries and get answer)
cout<<"?" <<l<<" " <<r<<endl;

$a = [5, 1, 4, 2, 3]$

int n;
cin>>n;

```

int l=1,r=5;
cout<<"?" "<<l<<" "<<r<<endl;
int x;
cin>>x; // x=3
cout<<"?" "<<4<<" "<<5<<endl;
cin>>x; // x=4
int ans;
cout<<"!" "<<ans<<endl;

```

1.....n
 $(1,n) \rightarrow z$
 $\text{int mid} = (n+1)/2;$
 $(1,mid) \text{ and } (mid+1,n)$
 $a_1, a_2, a_3, a_4 \rightarrow (a_1, a_2) \ (a_3, a_4) \rightarrow =z$
 $\qquad (a_2, a_4) \ (a_1, a_3) \rightarrow !=z$
 $\qquad (a_2, a_3) \ (a_1, a_4) \rightarrow !=z$
 $a_2 \text{ is in first half}(1...mid)$
 $(1,mid) \rightarrow x$

$\log(n) \rightarrow (15-17)*2 \leq 40$

Sol:

```

map<pair<int,int>,int> m;
int query(int l,int r)
{
    if(m.find(mp(l,r))!=m.end()){
        return m[mp(l,r)];
    }
    cout<<"?" "<<l<<" "<<r<<endl;
}

```

```

int z;
cin>>z;
m[mp(l,r)]=z;
return z;
}
void solve()
{
    int n;
    cin>>n; //5,1,4,2,3
    int l=1,r=n; //l=1,r=3
    while(r-l>1){
        int mid = (l+r)/2;
        int z = query(l,r);
        if(z<=mid){ //a2 is present in the first half
            int x = query(l,mid);
            if(x==z){ //a1 and a2 present in l...mid -> go
left
                r=mid;
            }else{ // a1 and a2 present in different
halfs....go right
                l=mid;
            }
        }else{ //a2 is present in 2nd half(mid+1...r)
            int x = query(mid,r);
            if(x==z){ // a1 and a2 present in same
half...mid to r
                l=mid;
            }else{ // a1 and a2 present in diff
half...go l...mid
        }
    }
}

```

```

        r=mid;
    }
}
int z = query(l,r);
if(z==l){
    cout<<"! "<<r<<endl;
}else{
    cout<<"! "<<l<<endl;
}
}

```

Q: https://atcoder.jp/contests/abc174/tasks/abc174_e

```

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

int32_t main(){

    int n,k;
    scanf("%d%d",&n,&k);
    vector<int> a(n);
    for(int i=0;i<n;i++) cin>>a[i];
    int low=1,high=1e9;
    // search space-> length of max stick;
    int ans=-1;
    while(low<=high){
        int mid=low+(high-low)/2;
        int req_number_of_cuts=0;

```

```

        for(int i=0;i<n;i++){
            if(a[i]%mid!=0)
req_number_of_cuts+=(a[i]/mid);
            else
req_number_of_cuts+=((a[i]/mid)-1);
        }
        if(req_number_of_cuts<=k){
            ans=mid;
            high=mid-1;
        }
        else low=mid+1;
    }
    cout<<ans<<endl;
}

```

Some useful built-in functions for Binary Search in C++:

1. If you are given a sorted vector ($a[i] \leq a[i+1]$) and a number x , and you need to find the index of the first element $\geq x$, this is called **lower_bound of x**.

In C++, there is a builtin function to find this in $O(\log N)$ using binary search.

Eg:

```
int ind = lower_bound(a.begin(), a.end(), x) - a.begin();
```

2. If you are given a sorted vector ($a[i] \leq a[i+1]$) and a number x ,

and you need to find the index of the first element $>x$, this is called **upper_bound of x**.

In C++, there is a builtin function to find this in $O(\log N)$ using binary search.

```
int ind = upper_bound(a.begin(), a.end(), x) - a.begin();
```

Suppose, there is no such element, then $\text{ind} = \text{size of the array}$.

Q.

<https://codeforces.com/edu/course/2/lesson/6/1/practice/contest/283911/problem/D>

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

using namespace std;

int32_t main() {
    int n;
    cin >> n;

    vector<int> vec(n);

    for (int i = 0; i < n; i++) {
        cin >> vec[i];
    }

    sort(vec.begin(), vec.end());

    int k;
    cin >> k;
```

```

int l, r;
while (k--) {
    cin >> l >> r;
    int y = upper_bound(vec.begin(), vec.end(), r) -
vec.begin();
    int x = lower_bound(vec.begin(), vec.end(), l) -
vec.begin();

    cout << y - x << ' ';
}

return 0;
}

```

Q.

<https://codeforces.com/edu/course/2/lesson/6/2/practice/contest/283932/problem/E>

```

#include <bits/stdc++.h>

using namespace std;

typedef long double ld;

const ld eps = 1e-6; // 10^(-6)

int32_t main() {
// fastio;

```

```
ld c;
cin >> c;

long double lo = 1, hi = 1e10, mid;

while (hi - lo >= eps) {
    mid = lo + (hi - lo) / 2;

    ld val = mid * mid + sqrt(mid);

    if (val <= c) {
        lo = mid;
    } else {
        hi = mid;
    }
}

cout << fixed << setprecision(7) << mid << endl;

return 0;
}
```

Q.

<https://codeforces.com/edu/course/2/lesson/6/2/practice/contest/283932/problem/D>

D. Children Holiday

time limit per test: 2 seconds 

memory limit per test: 512 megabytes

input: standard input

output: standard output

The organizers of the children's holiday are planning to inflate m balloons for it. They invited n assistants, the i -th assistant inflates a balloon in t_i minutes, but every time after z_i balloons are inflated he gets tired and rests for y_i minutes. Now the organizers of the holiday want to know after what time all the balloons will be inflated with the most optimal work of the assistants, and how many balloons each of them will inflate. (If the assistant has inflated the balloon and needs to rest, but he will not have to inflate more balloons, then it is considered that he finished the work immediately after the end of the last balloon inflation, and not after the rest).

Input

The first line of the input contains integers m and n ($0 \leq m \leq 15000, 1 \leq n \leq 1000$). The next n lines contain three integers each, t_i , z_i , and y_i , respectively ($1 \leq t_i, y_i \leq 100, 1 \leq z_i \leq 1000$).

Output

In the first line print the number T , the time it takes for all the balloons to be inflated. On the second line print n numbers, the number of balloons inflated by each of the invited assistants. If there are several optimal answers, output any of them.

Let $f(x) =$ No. of balloons that can be inflated in x minutes

It is a monotonic non-decreasing function.

Let $g(x) =$ Time taken by one person to inflate x balloons.

It is a monotonic non-decreasing function.

$x = \text{no. of balloons}$

Time taken by i^{th} assistant

$$= x \times t + \left(\frac{(x-1)}{2} \right) \times y$$

<https://codeforces.com/edu/course/2/lesson/6/2/practice/contest/283932/problem/D>

```
#include <bits/stdc++.h>
#define int long long

using namespace std;

int get_balloons(int p, int ti, int zi, int yi) {
    // p: time provided
    // return the maximum number of balloons that
    // this person can fill in t=x minutes

    int lo = 0;
    int hi = 1e8;
    int mid;
    int ans = 0;
```

```
while (lo <= hi) {
    mid = lo + (hi - lo) / 2;
    int gx = mid * ti + ((mid - 1) / zi) * yi;
    // time taken to fill x=mid balloons
    if (gx <= p) {
        ans = mid;
        lo = mid + 1;
    } else {
        hi = mid - 1;
    }
}
return ans;

}

int32_t main() {
// fastio;

int m, n;
cin >> m >> n;

vector<int> t(n), z(n), y(n);

for (int i = 0; i < n; i++) {
    cin >> t[i] >> z[i] >> y[i];
}

int lo = 0;
int hi = 1e8;
int mid;
```

```
int ans;

vector<int> balloons;

while (lo <= hi) {
    mid = lo + (hi - lo) / 2;

    // Find the number of balloons that can be
    // inflated in x=mid minutes

    int num = 0;    // no. of balloons filled
    vector<int> temp;
    for (int i = 0; i < n; i++) {
        if (num == m) {
            temp.push_back(0);
            continue;
        }
        int cnt = get_balloons(mid, t[i], z[i], y[i]);
        num += cnt;
        if (num >= m) {
            int diff = num - m;
            cnt -= diff;
            num = m;
        }
        temp.push_back(cnt);
    }

    if (num >= m) {
```

```
    balloons = temp;
    ans = mid;
    hi = mid - 1;
} else {
    lo = mid + 1;
}
}

cout << ans << '\n';

for (int i = 0; i < balloons.size(); i++) {
    cout << balloons[i] << ' ';
}
return 0;
}
```

Time complexity: $O(\log(10^8) * n * \log(10^8))$

2-pointers

Pointers mean an index in an array.

Q. You are given 2 sorted arrays: A of size n and B of size m.
Merge them into one sorted array.

Eg.

Input :

A = [1, 6, 9, 13, 18, 18]

B = [2, 3, 8, 13, 25]

Output:

[1, 2, 3, 6, 8, 9, 13, 13, 18, 18, 25]

Link:

<https://codeforces.com/edu/course/2/lesson/9/1/practice/contest/307092/problem/A>

Sol:

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

using namespace std;

typedef long long ll;

int32_t main() {
    int n, m;
    cin >> n >> m;
```

```
vector<int> a(n), b(m);

for (int i = 0; i < n; i++) {
    cin >> a[i];
}

for (int i = 0; i < m; i++) {
    cin >> b[i];
}

int i = 0, j = 0;
vector<int> c;

while (i < n && j < m) {
    if (a[i] <= b[j]) {
        c.push_back(a[i]);
        i++;
    } else {
        c.push_back(b[j]);
        j++;
    }
}

while (j < m) {
    c.push_back(b[j]);
    j++;
}
```

```

while (i < n) {
    c.push_back(a[i]);
    i++;
}

for (int i = 0; i < m + n; i++) {
    cout << c[i] << ' ';
}

return 0;
}

```

Time Complexity: $O(n + m)$

Q: <https://cses.fi/problemset/task/1641>

$A[] = \{2, 7, 5, 1\}$

$X = 8$

$2 \rightarrow \text{sumLeft} = 8 - 2 = 6$

$A[] = \{1, 2, 3, 4, 5, 6, 7, 8\}$

$1 \rightarrow \text{sumLeft} = x - 1 = 6 // x = 7$

$\text{Sum} = 10$

* $\text{Sum} == \text{sumLeft} \rightarrow$ we have found a triplet $\rightarrow 1, 2, 8$

* $\text{sum} < \text{sumLeft} \rightarrow$ we can ignore the minimum number i.e. the leftmost number

* $\text{sum} > \text{sumLeft} \rightarrow$ we can ignore the maximum number i.e. the rightmost number .

2,7,5,1 -> 1,3,4

1,2,3,4

1,2,5,7 -> 1,2,3

Sol:

```
int n,x;
cin>>n>>x;
vector<pii> v(n);
for(int i=0;i<n;i++){
    cin>>v[i].fi;
    v[i].se = i;
}
sort(v.begin(),v.end());
vector<int> ans;
for(int i=0;i<n;i++){ //v[i].fi is the 1st number of the
triplet
    int sumLeft = x-v[i].fi;
    int l=i+1,r=n-1;
    while(l<r){
        int sum = v[l].fi+v[r].fi;
        if(sum==sumLeft){ //a pair has been found
            // triplet = {v[i].se,v[l].se,v[r].se}
            ans.pb(v[i].se);
            ans.pb(v[l].se);
            ans.pb(v[r].se);
            break;
        }else if (sum<sumLeft){
            l++;
        }else{
```

```

        r--;
    }
}
if(ans.size()!=0){
    break;
}
sort(ans.begin(),ans.end());
if(ans.size()!=0){
    cout<<ans[0]+1<<" "<<ans[1]+1<<" "<<ans[2]+1;
}else{
    cout<<"IMPOSSIBLE";
}

```

Q: <https://codeforces.com/contest/279/problem/B>

N books numbered from 1 to n

Ith book takes $a[i]$ minutes

Free Time available: T minutes

Basically, we need to find largest continuous segment of the array such that the sum of elements in this segment $\leq t$

We can take 2 pointers - l and r

If sum within the range $[l, r] \leq t$, then we can increment r.

Otherwise if $sum > t$, we can increase l until sum becomes $\leq t$

Sol:

```
#include <bits/stdc++.h>
#define int long long

using namespace std;

typedef long long ll;

int32_t main() {
    int n, t;
    cin >> n >> t;

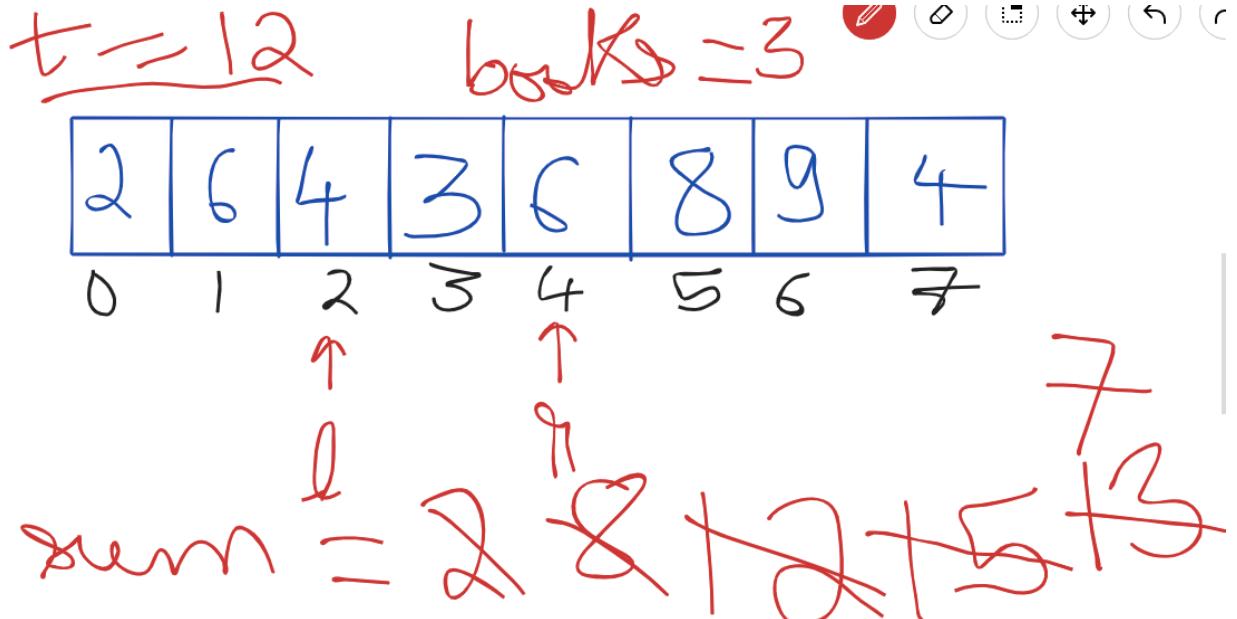
    vector<int> a(n);

    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }

    int sum = 0, ans = 0;
    int l = 0;
    for (int r = 0; r < n; r++) {
        sum = sum + a[r];
        while (sum > t) {
            sum -= a[l];
            l++;
        }
        ans = max(ans, r - l + 1);
    }
    cout << ans;
```

```
    return 0;  
}
```

Time complexity: $O(n)$



Q.

<https://codeforces.com/edu/course/2/lesson/9/2/practice/contest/307093/problem/B>

$s = 20$, and $= \cancel{5421} \cancel{27} \cancel{19}$

 $\text{Sum} = \cancel{28} + \cancel{25}$

```

#include <bits/stdc++.h>
#define int long long

using namespace std;

typedef long long ll;

int32_t main() {
    int n, s;
    cin >> n >> s;

    vector<int> a(n);

    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }

    int sum = 0, ans = 1e5 + 1;

```

```
int l = 0;

for (int r = 0; r < n; r++) {
    sum = sum + a[r];
    while (sum - a[l] >= s) {
        sum = sum - a[l];
        l++;
    }
    if (sum >= s) ans = min(ans, r - l + 1);
}

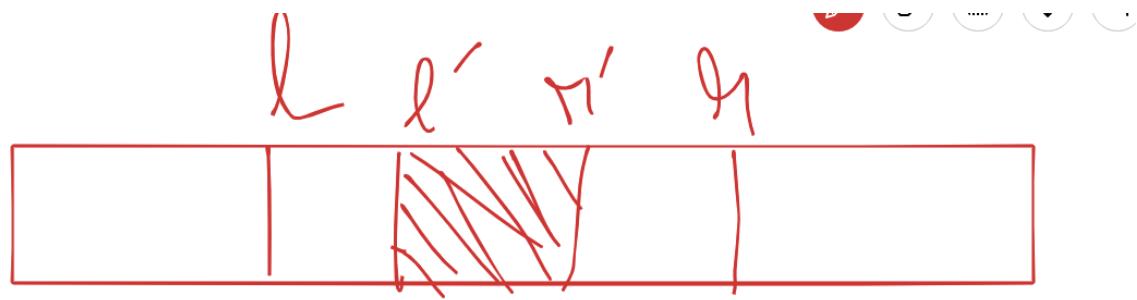
if(ans == 1e5 + 1)
    cout<<-1;
else
    cout << ans;

return 0;
}
```

Time Complexity: O(n)

When we can use 2-pointers?

Case 1:



If $[l, r]$ is good
 $\Rightarrow [l', r']$ is good

Example:

Sum of elements $\leq t$ (as solved in previous question - B. Books)

Case 2:



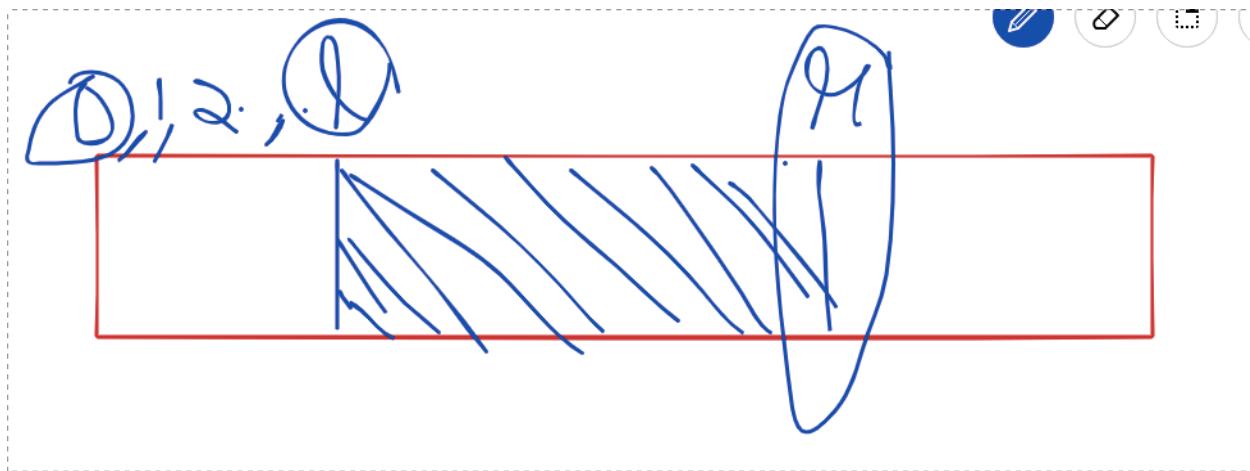
If $[l, r]$ is good
 $\Rightarrow [l', r']$ is good

Example:

Sum of elements $\geq s$ (as solved in the previous)

Q.

<https://codeforces.com/edu/course/2/lesson/9/2/practice/contest/307093/problem/D>



```
#include <bits/stdc++.h>
#define int long long

using namespace std;

typedef long long ll;

int32_t main() {
    int n, s;
    cin >> n >> s;
```

```

vector<int> a(n);

for (int i = 0; i < n; i++) {
    cin >> a[i];
}

int sum = 0, ans = 0;
int l = 0;

for (int r = 0; r < n; r++) {
    sum = sum + a[r];
    while (sum - a[l] >= s) {
        sum = sum - a[l];
        l++;
    }
    if (sum >= s) ans = ans + l + 1;
}
cout << ans;

return 0;
}

```

Time Complexity: O(n)

Practice Questions:

1.

<https://codeforces.com/edu/course/2/lesson/9/2/practice/contest/307093/problem/C>

2.

<https://codeforces.com/edu/course/2/lesson/9/2/practice/contest/307093/problem/E>

3.

<https://codeforces.com/edu/course/2/lesson/9/1/practice/contest/307092/problem/B>

4. <https://cses.fi/problemset/task/1640>

5. <https://codeforces.com/problemset/problem/702/C>

6. Try to solve the problems at:

<https://codeforces.com/edu/course/2/lesson/9/3/practice>

Warmup

Q.) You will be given a number x and you have to compute the value of $f(x)$

$$F[0]=1, F[1]=1$$

$$F(x)=F(x-1)+F(x-2) \quad x>=2$$

Fibonacci Series-> 1 1 2 3 5 8 13 21....

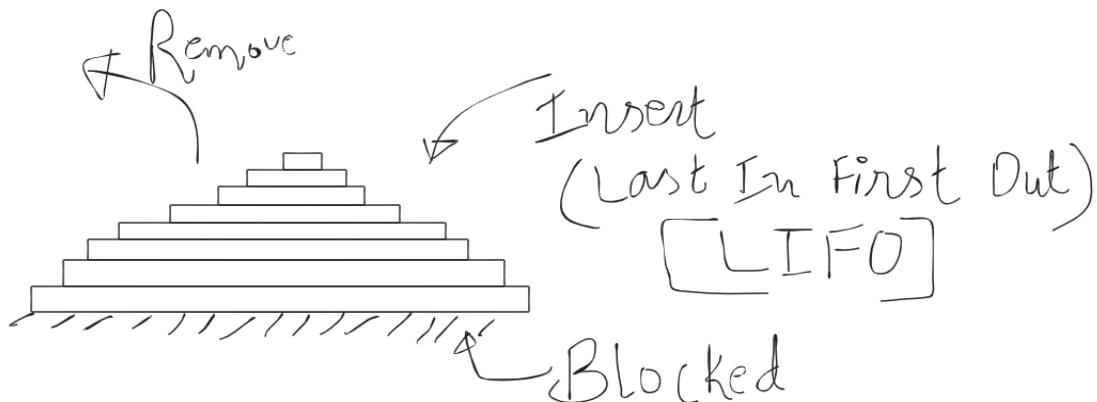
Recursive Solution->

```
int f(int x){  
    if(x==0 || x==1) return 1;  
    return f(x-1)+f(x-2);  
}
```

//f(5)-> f(4),f(3)-> f(3), f(2);

Stack

Stack is a linear data structure. The insertion and deletion happens only at one end. It follows the property of “**Last In First Out**”(LIFO)



Syntax for creating a stack

```
stack<data_type> stack_Name;
```

Example:

```
stack<char> st;  
stack<double> st;
```

Functions related to stack: Time Complexity=O(1)

1. push()-> Insert the last element at the back of your stack.

```
st.push(4);
```

2. pop() -> Remove the last element from the back.

```
st.pop();
```

3. empty()-> return true if the stack is empty()

```
st.empty(); //True/False
```

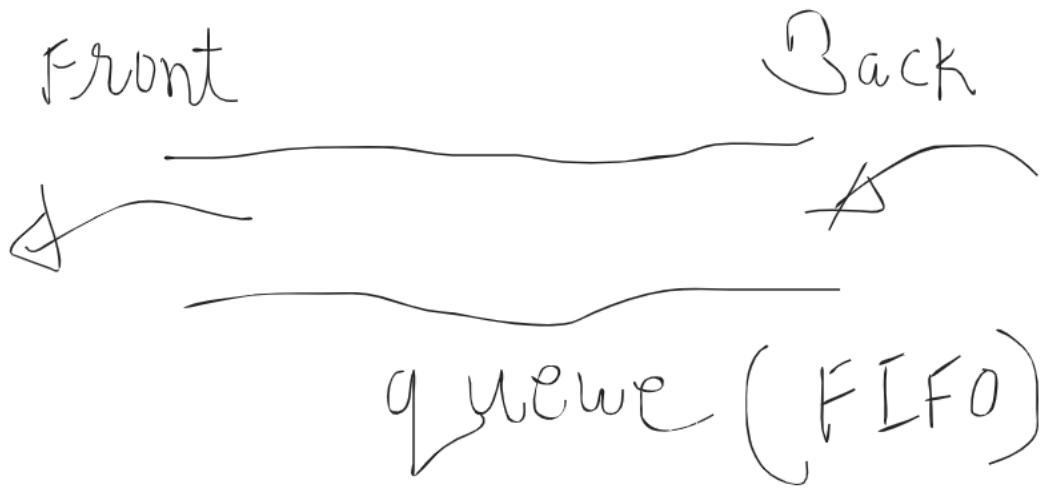
4. size() -> return the number of elements present in the stack.

5. top() -> show the top element

```
cout<<st.top();
```

Queue

Queue is also a linear data structure in which insertion happens from one end and deletion happens from another end.
It follows the property of “First In First Out”(FIFO)



Syntax for creating a queue

```
queue<data_type> queue_Name;
```

Example:

```
queue<char> q;  
queue<double> q;
```

Functions related to queue: Time Complexity=O(1)

1. push() -> insert the element at the back.

```
q.push(4);
```

2. pop() -> remove the element at the front.

```
q.pop();
```

3. front() -> show the element present at the front

```
cout<<q.front();
```

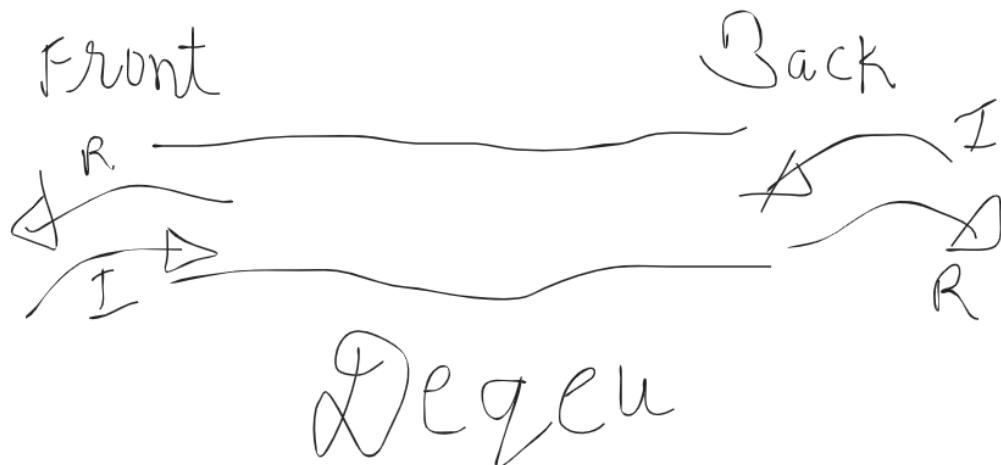
4. empty() -> return true if the queue is empty()

```
q.empty(); // returns True or False
```

5. size() -> return the number of elements present in the queue.

Deque (Double ended queue)

It is also a linear data structure but the main difference with queue is that insertion and deletion happens from both ends.



I-> Insertion

R-> Removal

Syntax of creating a deque

```
deque<data_type> deque_Name;
```

Example

```
deque<char> dq;  
deque<double> dq;
```

Functions related to deque: Time Complexity=O(1)

1. `push_back()` -> Insert an element at the back
2. `pop_back()` -> Remove the last element present at the back
3. `push_front()` -> Insert an element at the front
4. `pop_front()` -> Remove the first element present at the front
5. `empty()` -> return true if the deque is empty()

```
dq.empty(); // returns True or False
```

6. `size()`->return the number of elements present in the deque.

Note: All these functions for stack, queue and deque have O(1) constant time complexity

Some Problems

Q.) Balanced Parentheses.

A bracket is considered to be any one of the following characters: (,), {, }, [, or].

Two brackets are considered to be a matched pair if the opening bracket (i.e., (, [, or {) occurs to the left of a closing bracket (i.e.,),], or }) of the exact same type. There are three types of matched pairs of brackets: [], {}, and ()

Return whether the given string is a balanced parentheses or not.

Link of the problem:

<https://www.hackerrank.com/challenges/balanced-brackets/problem>

Examples:

()-> Balanced

[()]-> Balanced
[((])-> not balanced
[()]-> not balanced
{()}-> not balanced

Definition of Balanced Parentheses

Empty string is a balanced parentheses.

{, [], () -> balanced parentheses

Lets say a string s -> s is balanced.

{s}, (s), [s] -> balanced parentheses.

S and t -> s and t are balanced.

Then s+t -> balanced.

(D)

Solution Approach:

ith element -> if it is (, [, { -> push in the stack

-> if it is closing bracket] -> [,)->(, }-> {

(it should match with the top element in stack)

[()[]]

i=0 -> s[i]= [-> push in stack

Sol:

```
string s; // [[
cin>>s;
stack<char> st;
bool balanced=true;
for(int i=0;i<s.length();i++){
    if(s[i]=='('||s[i]=='{'||s[i]=='['){
        st.push(s[i]);
        continue;
    }
    // a closing bracket encountered....pop one element
from the stack
    if(!st.empty()){
        char c = st.top();
        st.pop();

if((s[i]==')'&&c=='(')|| (s[i]=='}'&&c=='{')||(s[i]==']'&&c==
'[')){
            // match of s[i] is found
    }else{
        // c and s[i] are not matching
        balanced=false;
        break;
    }
}else{
    balanced=false;
    break;
}
```

```
        }
    }
if(!st.empty()){
    balanced=false;
}
if(balanced){
    cout<<"Balanced";
}else{
    cout<<"Not balanced";
}
```

Q: Next Greater Element

<https://www.codechef.com/problems/DC206>

4,6,5

4,5,6

4,5,7,6

Maintain stack of possible answers

```
int a[n];
stack<int> s;
```

```
for(int i=n-1;i>=0;i--){
    while(!s.empty()&&a[i]>=s.top()){
        s.pop();
    }
    if(!s.empty()){
        ans[i]=s.top();
    } else
        ans[i]=-1;
    s.push(a[i]);
}
```

HW:

Try to write code for finding "previous smaller element" for each element.

Practice Problems on Stack, Queue, Deque

1. <https://www.hackerrank.com/challenges/deque-stl/problem>
2. <https://www.spoj.com/problems/STPAR/>
3. <https://www.hackerrank.com/challenges/largest-rectangle/problem>
4. <https://www.hackerrank.com/challenges/queries-with-fixed-length/problem>
5. <https://codeforces.com/contest/1373/problem/B>
6. <https://www.spoj.com/problems/JNEXT/>
7. <https://codeforces.com/contest/1374/problem/C>

8. <https://codeforces.com/contest/1313/problem/C2>

https://atcoder.jp/contests/abc190/tasks/abc190_d

Q.) How many arithmetic progressions consisting of integers with a common difference of 1 have a sum of n?

Approach:-

Series :- $a, a+1, a+2, \dots, a+(x-1)$ [a-> first term, x-> length]

$$\text{Sum} = x*(2*a + (x-1))/2 = n$$

$$x*(2*a + x - 1) = 2*n$$

$$x*(2*a + x - 1) = m$$

Solution:-

$$\underbrace{x}_{\text{X}} * \underbrace{[2a + (x-1)]}_{K_1} = m$$
$$K_1 = \frac{m}{x} \rightarrow \begin{cases} \text{if } x \neq 0 \\ \text{if } x = 0 \end{cases}$$
$$\frac{m}{x} = K_2 = x \rightarrow \begin{cases} \text{if } x \neq 0 \\ \text{if } x = 0 \end{cases}$$

```
#define ll long long int
ll n;
cin>>n;
ll m=2*n, ans=0;
for(ll x=1;x*x<=m;x++){
    if(m%x==0){
        ll k1=(m/x), k2=x;
        if((k1+1-x)%2==0) ans++;
        if(k2!=k1 && (k2+1-(m/x))%2==0) ans++;
        //If true then you got a series whose first term(a) and length=x
    }
}
cout<<ans<<endl;
```

Q:

Universe is defined by 2 elements \rightarrow s and t.

N universe are given .

Degree of connectivity is defined as

$$D(U_i, U_j) = S_i + S_j + |t_i - t_j|, \text{ if } |t_i - t_j| \leq m \\ 0 \quad \text{Else}$$

```
3 3 -> n and m
0 0 -> 1st universe
3 0 -> 2nd universe
9 2 -> 3rd universe
```

$$D(1,2) = 3$$

$$2 \leq n \leq 10^5$$

$$0 \leq m \leq 10^{16}$$

$$0 \leq s_i, t_i \leq 10^{16}$$

$$S_i + S_j + |t_i - t_j| \rightarrow S_i + S_j + t_i - t_j \text{ . if } t_i > t_j \\ S_i + S_j + t_j - t_i, \text{ if } t_j > t_i$$

$$S_i + S_j + t_i - t_j = (s_i + t_i) + (s_j - t_j)$$

Deque \rightarrow possible universes for that i that satisfy $t_i - t \leq m$

F.....b

B \rightarrow push i s-t =9

4,5,6

<https://codeforces.com/problemset/problem/582/A>

```
int main()
{
    int n;
    cin>>n;
    map<int,int,greater<int> > freq;
    // store frequency of all elements
    for(int i=0 ; i<n*n; i++)
    {
        int num;
        cin>>num;
        freq[num]++;
    }
    vector<int> ans;
    for(auto it=freq.begin(); it!=freq.end(); )
    {
        if(it->second>0)
        {
            freq[(it->first)]--;
            for(int i=0; i<ans.size(); i++)
            {
                freq[__gcd(ans[i],(it->first))]-=2;
            }
            ans.push_back(it->first);
        }
        else
        {
            it++;
        }
    }
}
```

```
    }
}

for(int i=0; i<n; i++)
{
    cout<<ans[i]<<" ";
}
}
```

Time complexity: $O(n^2 \log(n))$

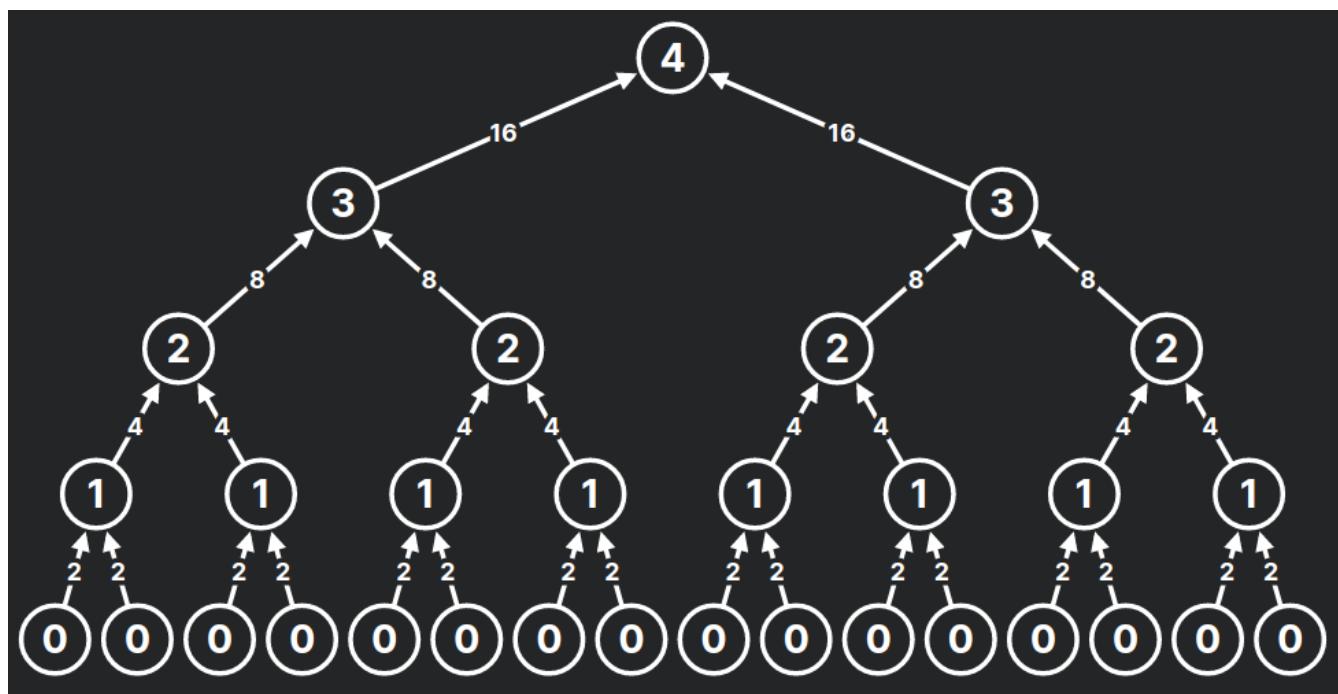
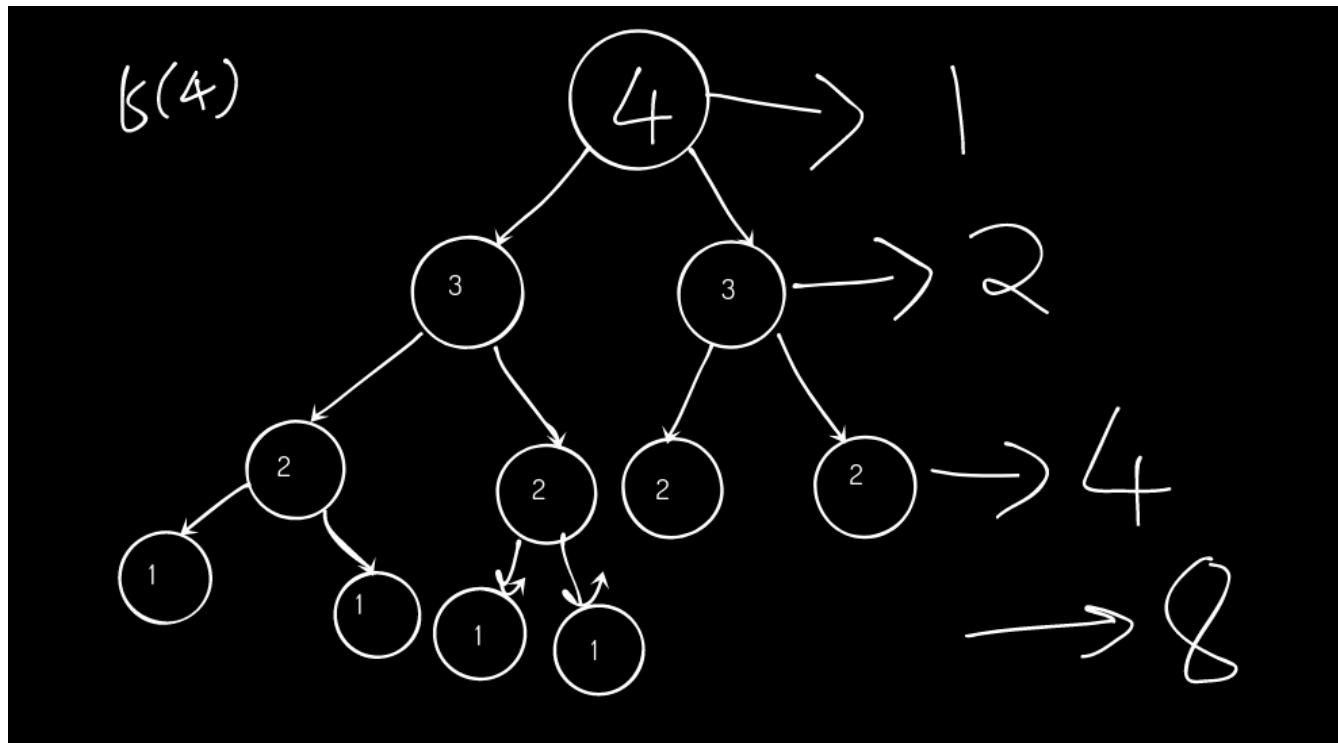
Introduction to Dynamic Programming

Finding time complexity of a recursive code

What will be the time complexity of this code ?

```
int f(int x)
{
    if(x==0)
    {
        return 2;
    }
    else
    {
        return f(x-1) + f(x-1);
    }
}
```

Let us take x = 4



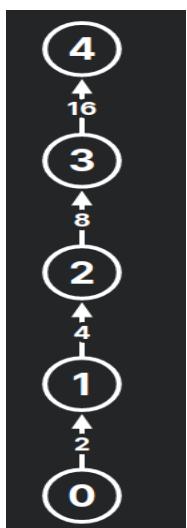
$$\begin{aligned}
 \text{Number of function calls} &= 1 + 2 + 4 + 8 + 16 \\
 &= 31 \\
 &= 2^5 - 1
 \end{aligned}$$

For if call $f(n)$, you will get $2^{(n+1)} - 1$ function calls
So, the time complexity = $O(2*2^n - 1) = O(2^n)$

Now, let's change 1 line in the code

```
int f(int x)
{
    if(x==0)
    {
        return 2;
    }
    else
    {
        return f(x-1) * 2;
    }
}
```

What is the time complexity of this code?



Time complexity = $O(n + 1) = O(n)$

You can try this website for visualising the recursion tree:
<https://recursion.now.sh/>

Intuition of Dynamic Programming

$$1 + 2 + 6 + 7 + 5 = ?$$

21

$$1 + 2 + 6 + 7 + 5 + 2 = ?$$

$$21 + 2 = 23$$

This is DP (Dynamic Programming). **Just remember the past answers and use it to compute your answer.**

Fibonacci Numbers

N : 1, 2, 3, 4, 5, 6.....

F(N) : 0, 1, 1, 2, 3, 5....

Recurrence relation:

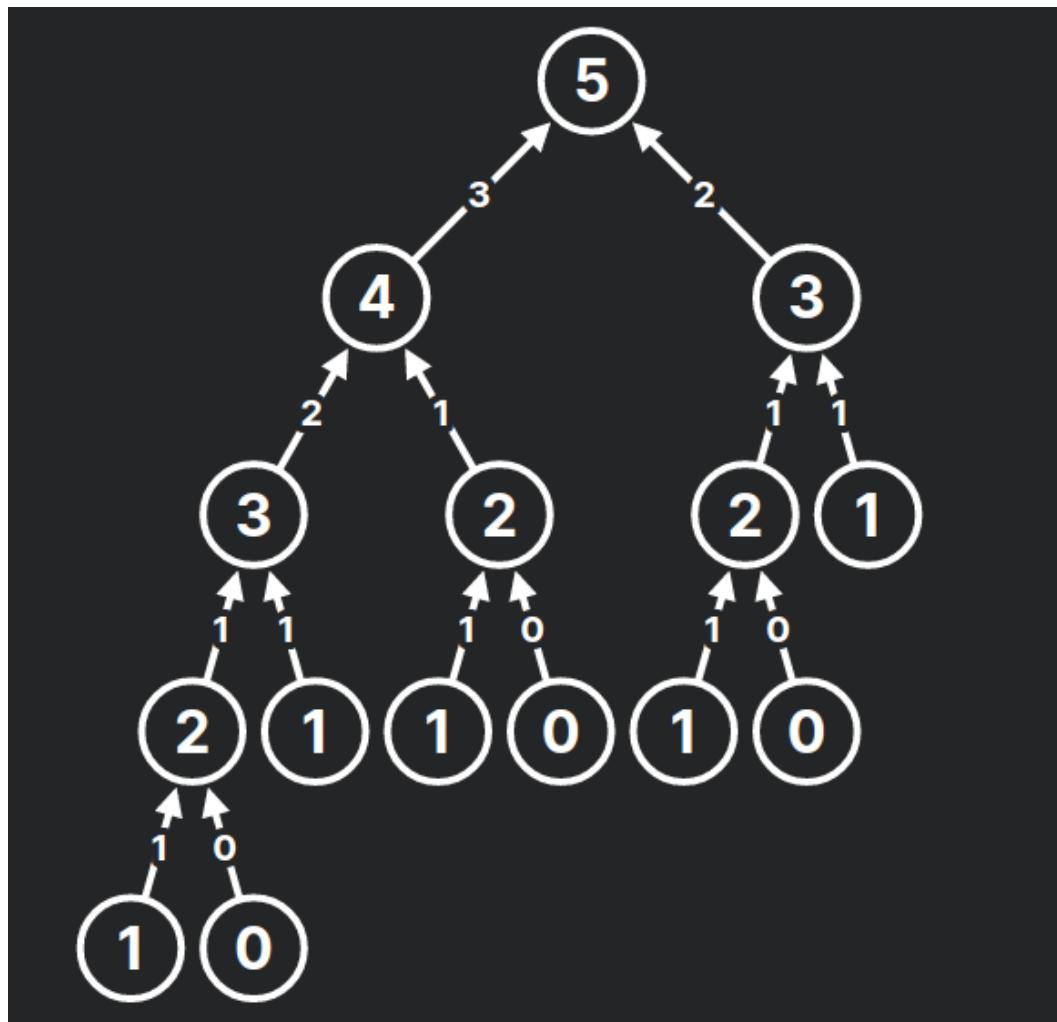
$$F(N) = F(N-1) + F(N-2)$$

Recursive code for find Nth Fibonacci number (Without DP)

```
int fib( int n)
{
    if(n==1)
        return 0;
    if(n==2)
        return 1;
    return fib(n-1) + fib(n-2);
```

}

Time complexity: $O(2^n)$



DP = Recursion + Memoization

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

using namespace std;

const int MAX = 100000+1;

int dp[MAX]; // dp[i] = i-th fibonacci number

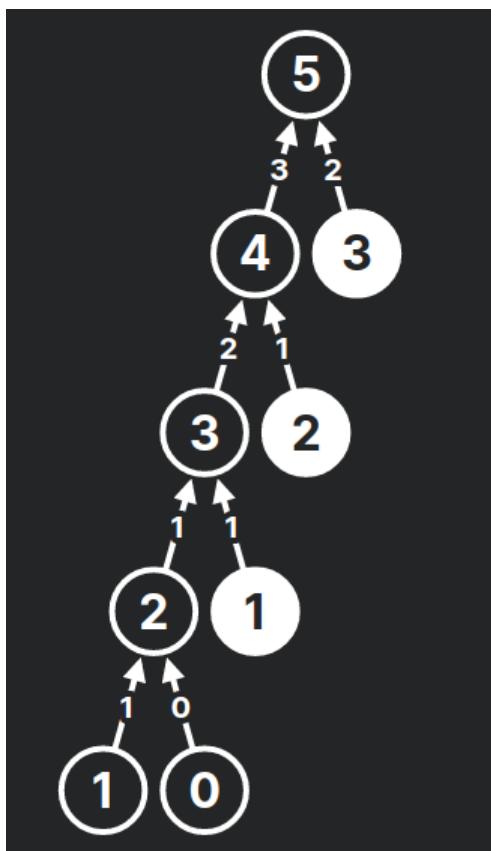
int fib(int n)
{
    if(n==1)
        return 0;
    if(n==2)
        return 1;
    if(dp[n] != -1)
    {
        return dp[n];
    }
    return dp[n]=fib(n-1) + fib(n-2); // Memoization
}

int main() {

    for(int i=0; i<MAX; i++)
    {   dp[i]=-1; // no values are computed at the beginning
    }
}
```

```
int n;  
cin>>n;  
cout<<fib(n);  
return 0;  
}
```

Time complexity: $O(n)$



DP without recursion (iterative DP)

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

```

using namespace std;

const int MAX = 100000+1;

int dp[MAX]; // dp[i] = i-th fibonacci number

int main() {

    int n;
    cin>>n;

    dp[1]=0;
    dp[2]=1;

    for(int i=3; i<=n; i++)
    {
        dp[i] = dp[i-1] + dp[i-2];
    }

    cout<<fib(n);
    return 0;
}

```

Time complexity: O(N)

Wherever we see a recursive solution that has repeated calls for the same inputs, we can optimize it using Dynamic Programming. The idea is to **simply store the results of subproblems**, so that we do not have to re-compute them when needed later. This simple

optimization reduces time complexities from exponential to polynomial.

Bottom Up vs Top Down

Bottom Up Approach:

Analogy to understand:

I am going to learn to program. Then, I will start practising. Then, I will participate in coding contests. I will improve by solving those questions which I couldn't solve during every contest. I will be able to crack an internship at a good company.

In Bottom-up you start with the small solutions (base case) and build up.

Example: The without-recursion approach (iterative) for finding n-th fibonacci number shown above

Advantages :

1. Fast and uses less memory than top down.
2. Shorter Code

Top Down Approach:

Analogy to understand:

I will be able to crack an internship at a good company. How? I will improve by solving those questions which I couldn't solve during every

contest. How? I will participate in coding contests. How? I will start practicing? I am going to learn to program.

In Top-down you start building the big solution right away by explaining how you build it from smaller solutions.

Example: The recursion + memoization approach shown above

Advantages :

1. Easy to apply
2. Order doesn't matter.

Q: https://atcoder.jp/contests/dp/tasks/dp_a

Recursion Solution:-

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

int minCost(int i){
// It will give me the minimum cost to reach at ith stone
    if(i==0) return 0;
    if(i==1) return abs(h[1]-h[0]);
    if(Memo[i]!=-1) return Memo[i];
    int lastCost = minCost(i-1) + abs(h[i]-h[i-1]);
    int lastlastCost = minCost(i-2) + abs(h[i]-h[i-2]);
    Memo[i]=min(lastCost,lastlastCost);
    return Memo[i];
```

```

}

int main(){
    int n;
    cin>>n;
    h.resize(n);
    Memo.resize(n);
    for(int i=0;i<n;i++) Memo[i]=-1;
    for(int i=0;i<n;i++) cin>>h[i];
    cout<<minCost(n-1);
    return 0;
}

```

Time Complexity without Memoization = $O(2^n)$

Time Complexity with Memoization = $O(n)$

- H.W- Solve the last problem using iterative DP.

Q.) You are climbing a staircase. It takes **n steps** to reach the top.
 Each time you can either **climb 1 or 2 steps**. In **how many distinct ways** can you climb to the top?

Iterative Solution:-

```

int climbStairs(int n) {
    vector<int> dp(n+1);
    //dp[i]-> number of ways to reach at i-th floor
    dp[0]=1;
    dp[1]=1;
    for(int i=2;i<=n;i++) dp[i]=dp[i-1]+dp[i-2];
    return dp[n];
}

```

- H.W- Solve the last problem using Recursive DP

Practice Problems:

1. https://atcoder.jp/contests/dp/tasks/dp_b
2. <https://www.hackerearth.com/practice/algorithms/dynamic-programming/introduction-to-dynamic-programming-1/practice-problems/algorithm/jump-k-forward-250d464b/>
3. https://atcoder.jp/contests/dp/tasks/dp_c
4. https://atcoder.jp/contests/abc129/tasks/abc129_c
5. <https://codeforces.com/problemset/problem/1245/C>
6. <https://codeforces.com/problemset/problem/455/A%7C>
7. <https://codeforces.com/problemset/problem/1195/C>
8. <https://www.spoj.com/problems/ACODE/>
9. <https://codeforces.com/problemset/problem/189/A>

Just Follow only these websites for practising in first year:

Codeforces, Atcoder, Codechef, Hackerrank, Hackearth, Spoj

On hackerrank, solve all the implementation problems:

(Very important for building the basics)

https://www.hackerrank.com/domains/algorithms?filters%5Bsub_domains%5D%5B%5D=implementation