L2 : Getting Started Practice Questions and Concepts

1-Tut : What is the output

Send Feedback

What will be the output of the following code ?

```
#include <iostream>
using namespace std;
int main(){
    cout << code ;

}
```

Options

This problem has only one correct answer

code
Error

Correct Answer

Solution Description

If we want to print something on screen exactly, we need to put that text in double quotes, otherwise error will come. For eg. cout << "code";

2-Tut : What is the output

Send Feedback

What will be the output of the following code ?

```
#include <iostream>
using namespace std;
int main(){
    cout << "career" << "labs";
}
```

Options

This problem has only one correct answer

careerlabs
career labs

Correct Answer

Solution Description

The text will be printed exactly like we placed inside double quotes. So first we are printing "career". After printing, our cursor will be right next to the 'r' of "carrer". So "labs" will be printed right next to the "career" without any spaces between them.

3-Tut : Datatype

Send Feedback

Which of the following data types stores the longest decimal number ?

Options

This problem has only one correct answer

long
float
double
short

Correct Answer

Solution Description

Out of all given options, only float and double can hold decimal numbers. Size of the float is 4 bytes and double is 8 bytes (in most of the compilers, as the size of data types is compiler specific). So double can store bigger decimal numbers.

4-Tut : Garbage value

Send Feedback

Compiler assigns a garbage value to an uninitialised local variable in C++ Programming ?

Options

This problem has only one correct answer

true
false

Correct Answer : true

Solution Description

In C++, all variables contain garbage values before their initialisation.

5-Tut : What is the output

Send Feedback

What will be the output of following statements -

```
bool a = true;
cout << a;
```

Answer

Type here : 1

Correct Answer : 1

Solution Description

In c++, when we print the value of a bool variable, it prints either 0 (for false) or 1 (for true). It doesn't print "true" or "false".

6-Tut : What is the output

Send Feedback

What is the output of the following code if the input is : 5 15?

```
#include <iostream>
 using namespace std;
 int main() {
       int a, b;
   cin >> a;
   cin >> b;
   cout << (a+b);
          return 0;
 }
```

Options

This problem has only one correct answer

5
15
20
Error

Correct Answer

Solution Description

We are asking two integer values from the user and input is "5 15" (without quotes). So 5 will be assigned to a and 15 will be assigned to b. We can enter multiple input values either separated by space or in new lines.

Hence, the result will be 20.

7-Tut : What is the output

Send Feedback

What is the output of the following code if the input is : 2 10.1 D ?

```
#include <iostream>
using namespace std;
int main() {
   int a;
   double d;
   char c;
   cin >> a >> d >> c;
   cout << a << d << c << endl;

}
```

Options

This problem has only one correct answer

2 10.1 D
2
210.1D
Error

Correct Answer : C

Solution Description

We are taking 3 inputs - one integer, one double and one character value. And our input is : "2 10.1 D" (without quotes). So, 2 will be assigned to a, 10.1 will be assigned to d and D will be assigned to c.
We can enter multiple input values either separated by space or in new lines.

Then we are printing all the three values one by one, without any space between them. So ans is : 210.1D

8-Tut : What is the output

Send Feedback

What will be the output of the following statement ?

```
cout << (‘a’ + 1);
```

Answer

Type here

Correct Answer

Solution Description

When you add a character and an integer, answer is also an integer. It will add the ASCII value of char ‘a’ i.e 97 and int 1. So answer will be 98.

9-Tut : What is the output

Send Feedback

What will be the output of following statement ?

```
int i = 'c';

cout << i;
```

Answer

Type here

Correct Answer

Solution Description

When we put char 'c' into an integer, its ASCII value will be put in the integer i.e. 99.

10-Tut : What is the output

Send Feedback

What is the output ?

```
#include <iostream>
using namespace std;
int main()
{
    char c = 74;
    cout << c;
    return 0;
```

```
}
```

Answer

Type here

Correct Answer

Solution Description

When we assign an integer to a character variable, that integer will be treated as ASCII value of corresponding character. 74 is ASCII value of letter 'J'. Hence, 'J' will be printed.

11-Tut : What is the output

Send Feedback

What is the output ?

```
#include <iostream>
using namespace std;
int main()
{
   int a = 10;
   char ch = 'a';
   ch = ch + a;
   cout << ch << endl;
}
```

Answer

Type here

Correct Answer

Solution Description

We know that the addition of a character and an integer leads to an integer. So the statement - "ch + a" will give the answer 107 (ASCII value of 'a' is 97). But we are assigning this integer now to the character variable. Hence, 107 will be saved in the variable "ch" and on printing it will print the character which is having ASCII value 107 i.e. 'k'.

12-Tut : What is the output

Send Feedback

What will be the output ?

```
#include <iostream>
using namespace std;
int main()
{
   double a = 6 / 4;
   int b  = 6 / 4;
   double c = a + b;
   cout << c;

}
```

Options

This problem has only one correct answer

3
1.5
2
2.5

Correct Answer : 2

Solution Description

When 6 / 4 is performed, both the operands of / are int hence the answer will be an int i.e. 1. Hence the value of both a and b is 1. Thus a + b will be 2. In C++, if a double value doesn't contain any decimal value, only an integer will be printed.

13-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main()
{
    double a = 55.5;
    int b = 55;
    a = a % 10;
    b = b % 10;
    cout << a << " "  << b;

}
```

Options

This problem has only one correct answer

5 5
5.5 5
6 5
Syntax error

Correct Answer : D

Solution Description

% can’t be used with double or float.

14-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main()
{
    int var1 = 5;
    int var2 = 6;
    cout << (var1 > var2);
```

```
}
```

Options

This problem has only one correct answer

true
false
0
1

Correct Answer : 0

Solution Description

'>' operator gives a boolean answer. And the condition is false. So the answer will be false and 0 will be printed (boolean variables print only 0 and 1).

L3: Conditionals and Loops Practice Questions

1-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main()
{
    int x = 5;
    if (x < 6)
        cout <<  “Hello ”;
    if(x == 5){
        cout << “Hi ”;
    }
    else {
        cout << “Hey ”;
    }

}
```

Options

This problem has only one correct answer

Hello
Hi
Hello Hi
Hello Hey

Correct Answer : c

Solution Description

First if condition is true, so "Hello " will be printed. After that, next if condition will be evaluated (as both if's are independent if's) which is again true. Hence, next "Hi " will be printed (obviously in same line). We won't go inside else, as if is already executed.

2-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main() {
    int x = 15;
    if(x <= 15) {
        cout << "Inside if ";
    } else if(x == 15) {
        cout << "Inside else if ";
    }
    cout << x;

}
```

Options

This problem has only one correct answer

Inside if
15
Inside if Inside else if 15
Inside if 15

Correct Answer

Solution Description

The condition inside if is true, hence the statement "Inside if" will be printed. After that, we'll directly jump to the statement : cout << x. So, after that 15 will be printed.

3-Tut: What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main()
{
      int var1 = 5;
      int var2 = 6;
      if ((var2 = 1) == var1)
        cout << var2;
      else
         cout  << (var2 + 1);

}
```

Options

This problem has only one correct answer

7
1
2
6

Correct Answer

Solution Description

Inside if condition, we are doing if((var2 = 1) == var1), So first 1 will be assigned to var2 and then it'll be compared with var1. Before comparison, var2 = 1 and var1 = 5, which are not equal. So, we'll move to else part and there the value of (var2 + 1) i.e. 2 will be printed.

4-Tut : If statement

Send Feedback

For what values of the x is an if-statement block not executed ?

```
    if((1+x) != (1-x)) {
        cout<<"In if block";
```

```
    }
```

Options

This problem has only one correct answer

0 and all negative values
0 and -1
0, all negative values, all positive values except 1
0

Correct Answer : D

Solution Description

The if-statement block is only not executed when the value of x is 0. For all other values, it will be treated as true.

5-Tut: What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main() {
   int a = 10, b = 20, c = 100;
   if(a <= b || c <= b) {
      cout << "hello" << endl;
   }
   else if(a <= b || a <= c) {
      cout << "hi" << endl;
   }
   else {
      cout << "hey" << endl;
   }
}
```

Answer

Type here : hello

Correct Answer : hello

Solution Description

In if condition, we are using logical OR operator ("||"). So if even one condition is true, final answer will be true.

6-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main() {
   int a = 10, b = 20, c = 30;
   if(a <= b && !b) {
      cout << "hello";
   }
   else if(c >= a && c >= b) {
      cout << "hi";
```

```
    }
    else {
        cout << "hey";
    }

}
```

Answer

Type here : hi

Correct Answer

Solution Description

a <= b evaluates to true, but !b evaluates to false (! operator negates the value, b is non-zero and hence !b becomes 0 which is false). For AND operator ("&&"), all the condition must be true. So we'll move to else if condition which evaluates to be true as both conditions (c >= a and c >= b) are true. Hence, "hi" will be printed.

7-Tut : Check Case

Send Feedback

Write a program that takes a character as input and prints either 1, 0 or -1 according to the following rules.

1, if the character is an uppercase alphabet (A - Z)

0, if the character is a lowercase alphabet (a - z)

-1, if the character is not an alphabet

Input format :
Single Character
Output format : 1 or 0 or -1
Constraints : Input can be any character
Sample Input 1 : v
Sample Output 1 : 0
Sample Input 2 : V
Sample Output 2 : 1
Sample Input 3 : #
Sample Output 3 :-1

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

int main() {
        // Write your code here
    char c;
    cin >> c;
    if(c >= 'A' && c <= 'Z'){
        cout << 1 ;
    }else if(c >= 'a' && c <= 'z'){
        cout << 0;
    }else {
```

```
        cout << -1;
    }


}
```

8-Tut : Sum of Even Numbers till N

Send Feedback

Given a number N, print sum of all even numbers from 1 to N.

Input Format : Integer N
Output Format : Required Sum
Sample Input 1 : 6
Sample Output 1 : 12

```
#include<iostream>
using namespace std;
int main(){
        /*
                Read input as specified in the question.
                Print output as specified in the question.
        */
    int N,sum = 0,ceven = 0;
    cin >> N;
    while(ceven <= N){
        sum = sum + ceven;
        ceven = ceven + 2;
    }
   cout << sum ;
}
```

9-Tut: Fahrenheit to Celsius Table

Send Feedback

Given three values - Start Fahrenheit Value (S), End Fahrenheit value (E) and Step Size (W), you need to convert all Fahrenheit values from Start to End at the gap of W, into their corresponding Celsius values and print the table.

Input Format :
3 integers - S, E and W respectively

Output Format :
Fahrenheit to Celsius conversion table. One line for every Fahrenheit and corresponding Celsius value. The Fahrenheit value and its corresponding Celsius value should be separate by single space.

Constraints :

0 <= S <= 80
S <= E <= 900
0 <= W <= 40

Sample Input 1:

0
100
20

Sample Output 1:

0 -17
20 -6
40 4
60 15
80 26
100 37

Sample Input 2:

20
119
13

Sample Output 2:

20 -6
33 0
46 7
59 15
72 22
85 29
98 36
111 43

Explanation For Input 2:

Start calculating the Celsius values for each Fahrenheit Value which starts from 20. So starting from 20, we need to compute its corresponding Celsius value which computes to -6. We print this information as

<Fahrenheit Value> <a single space> <Celsius Value> on each line. Now add 13 to Fahrenheit Value at each step until you reach 119 in this case. You may or may not exactly land on the end value depending on the steps you are taking.

```
#include<iostream>
using namespace std;
int f2c(int f){
    return (5.0/9)*(f-32);
};

int main(){

      /*  Read input as specified in the question.
          * Print output as specified in the question.
          */
   int S, E , W;
   cin >> S >> E >> W;
  int f = S;
   while(f <= E){
      cout << f << " "<< f2c(f) << endl;
      f = f+ W;

   }
}

```

10-Tut: Number Pattern 1

Send Feedback

Print the following pattern

Pattern for N = 4

```
1
23
345
4567
```

Input Format :

N (Total no. of rows)

Output Format :

Pattern in N lines

Sample Input 1 : 3

Sample Output 1 :

```
1
23

345
```

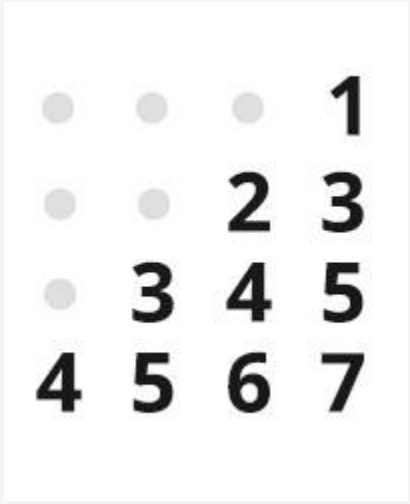
```cpp
#include<iostream>
using namespace std;
int main(){

        /*  Read input as specified in the question.
            * Print output as specified in the question.
            */
    int N ;
     cin >> N;
    int i =1;
    while(i <= N){
       int j =1, k=i;

       while( j <= i){
          cout << k;
          j = j+1;
          k = k+1;
       }
       cout << endl;

       i = i +1;
    }
}
```

11-Tut : Number Pattern 2

Send Feedback

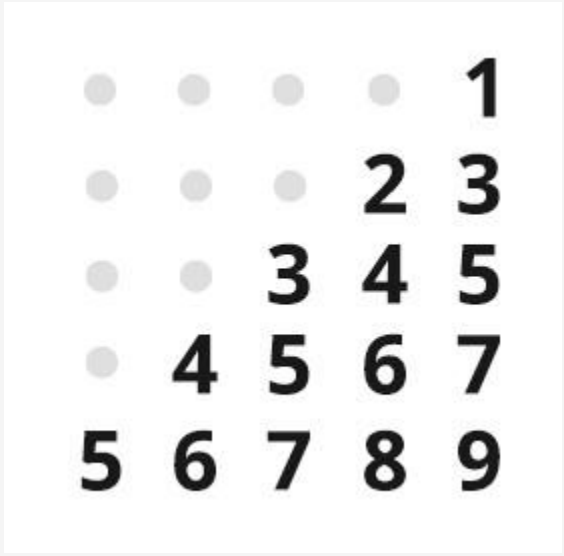
Print the following pattern

Pattern for N = 4



The dots represent spaces.

Input Format :
N (Total no. of rows)
Output Format :
Pattern in N lines
Sample Input : 5
Sample Output :



The dots represent spaces.

```
#include<iostream>
using namespace std;
int main(){

        /*  Read input as specified in the question.
            * Print output as specified in the question.
            */
    int N ;
    cin >> N;
    int i = 1;
    while(i <= N){
        int s = 1;
        while(s <= N-i){
            cout << " ";
            s++;
        }
        s = i;
        int j = 1;
        while(j <= i){
            cout << s;
            j++;
            s++;
        }

```

```
        cout << endl;
        i++;
    }
}
```

12-Tut : Start Pattern

Send Feedback

Print the following pattern

Pattern for N = 4

```
···*
··***
·*****
*******
```

The dots represent spaces.

Input Format : N (Total no. of rows)
Output Format : Pattern in N lines
Constraints : 0 <= N <= 50
Sample Input 1 : 3
Sample Output 1 :

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

Sample Input 2 : 4
Sample Output 2 :

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

*******
```

```
#include<iostream>
using namespace std;
int main(){

    /*  Read input as specified in the question.
        * Print output as specified in the question.
        */
    int N ;
     cin >> N;
    int i =1;
    while(i <= N){

        int s = 1;
        while(s <= N-i){
            cout << " ";
```

```
            s++;
        }

        s =1;
        while(s <= 2*i -1){
            cout << "*";
            s++;
        }
        cout << endl;

        i++;
    }
}
```

13-Ass : Total Salary

Send Feedback

Write a program to calculate the total salary of a person. The user has to enter the basic salary (an integer) and the grade (an uppercase character), and depending upon which the total salary is calculated as -

totalSalary = basic + hra + da + allow – pf

where :

hra = 20% of basic
da = 50% of basic
allow = 1700 if grade = ‘A’
allow = 1500 if grade = ‘B’
allow = 1300 if grade = ‘C' or any other character
pf = 11% of basic.

Round off the total salary and then print the integral part only.

Note: Try finding out a function on the internet to do so

Input format :

Basic salary & Grade (separated by space)

Output Format :

Total Salary

Constraints :

0 <= Basic Salary <= 7,500,000

Sample Input 1 :

10000 A

Sample Output 1 :

17600

Sample Input 2 :

4567 B

Sample Output 2 :

8762

Explanation of Input 2:

We have been given the basic salary as Rs. 4567. We need to calculate the hra, da and pf.
Now when we calculate each of the, it turns out to be:
hra = 20% of Rs. 4567 = Rs. 913.4
da = 50% od Rs. 4567 = Rs. 2283.5
pf = 11% of Rs. 4567 = Rs. 502.37
Since, the grade is 'B', we take allowance as Rs. 1500.

On substituting these values to the formula of totalSalary, we get Rs. 8761.53 and now rounding it off will result in Rs. 8762 and hence the Answer.

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

int main() {
        // Write your code here
        unsigned long int basic;
    char grade;
    cin >> basic >> grade ;

    double hra,da,pf;
    hra = 0.2 * basic;
    da = 0.5 * basic;
    pf = 0.11 *basic;

    float allow;
    if(grade == 'A'){
        allow = 1700;
    }
    else if(grade == 'B'){
        allow = 1500;
    }else {
        allow = 1300;
    }
    double totalsalary = basic + hra + da + allow - pf;
    unsigned long int totalsalary1 = totalsalary;
    // double totalsalary2 = totasalary1 + 0.5;
    if( totalsalary >= totalsalary1 +0.5){
        totalsalary1 = totalsalary1+1;
    }
    cout << totalsalary1;
}
```

14-Ass : Sum of even & odd

Send Feedback

Write a program to input an integer N and print the sum of all its even digits and sum of all its odd digits separately.

Digits mean numbers, not the places! That is, if the given integer is "13245", even digits are 2 & 4 and odd digits are 1, 3 & 5.

Input format :

Integer N

Output format :

Sum_of_Even_Digits Sum_of_Odd_Digits

(Print first even sum and then odd sum separated by space)

Constraints 0 <= N <= 10^8

Sample Input 1: 1234

Sample Output 1: 6 4

Sample Input 2: 552245

Sample Output 2: 8 15

Explanation for Input 2:

For the given input, the even digits are 2, 2 and 4 and if we take the sum of these digits it will come out to be 8(2 + 2 + 4) and similarly, if we look at the odd digits, they are, 5, 5 and 5 which makes a sum of 15(5 + 5 + 5). Hence the answer would be, 8(evenSum) <single space> 15(oddSum)

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

int main() {
        // Write your code here
        int N;
    cin >> N;
    short int even_sum = 0, odd_sum = 0;

    int n =N;

    while(n != 0)
    {
       int r = n %10;
      if(r % 2 == 0)
      {
          even_sum = even_sum + r;
      }else
      {
          odd_sum = odd_sum + r;
      }
       n = n/10;
    }

    cout << even_sum << " " << odd_sum;
}

```

15-Ass : Find power of a number

Send Feedback

Write a program to find x to the power n (i.e. x^n). Take x and n from the user. You need to print the answer.

Note : For this question, you can assume that 0 raised to the power of 0 is 1

Input format :

Two integers x and n (separated by space)

Output Format :

x^n (i.e. x raise to the power n)

Constraints:

0 <= x <= 8

0 <= n <= 9

Sample Input 1 :

3 4

Sample Output 1 :

81

Sample Input 2 :

2 5

Sample Output 2 :

32

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

int main() {
        // Write your code here
        int x,n;
    cin >> x >> n;
    if(x == 0 && n == 0){
        cout << 1;
    }
    else{
        int i = 1,val =1;
        while(i <= n){
            val = val *x;
            i ++;
        }

        cout << val;
    }
}
```

L4 : Pattern 1 Practice Questions

1-Tut : Code : Square Pattern

Send Feedback

Print the following pattern for the given N number of rows.

Pattern for N = 4

4444
4444
4444
4444

Input format :

Integer N (Total no. of rows)

Output format :

Pattern in N lines

Constraints

0 <= N <= 50

Sample Input 1:

7

Sample Output 1:

7777777
7777777
7777777
7777777
7777777
7777777
7777777

Sample Input 1:

6

Sample Output 1:

666666
666666
666666
666666
666666
666666

```
#include<iostream>
using namespace std;
int main(){

        /*  Read input as specified in the question.
            * Print output as specified in the question.
            */
    int N;
```

```
    cin >> N;
    int i =1;
    while(i <= N){
            int j=1;
          while(j <= N){
              cout << N;
              j++;
          }
       cout << endl;
       i++;
    }
}
```

2-Tut : Code : Triangular Star Pattern

Send Feedback

Print the following pattern for the given N number of rows.

Pattern for N = 4

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

Note : There are no spaces between the stars (*).

Input format :

Integer N (Total no. of rows)

Output format :

Pattern in N lines

Constraints

0 <= N <= 50

Sample Input 1: 5

Sample Output 1:

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

Sample Input 2: 6

Sample Output 2:

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

******
```

```
#include<iostream>
using namespace std;
int main(){

        /*  Read input as specified in the question.
            * Print output as specified in the question.
            */
    int N;
     cin >> N;
    int i = 1;
    while(i <= N){
        int j = 1;
        while(j <= i){
            cout << "*";
            j++;
        }
        cout<<endl;

        i++;
    }
}
```

3-Tut : Code : Triangle Number Pattern

Send Feedback

Print the following pattern for the given N number of rows.

Pattern for N = 4

```
1
22
333
4444
```

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints 0 <= N <= 50

Sample Input 1: 5

Sample Output 1:

```
1
22
333
4444
55555
```

Sample Input 2: 6

Sample Output 2:

```
1
22
333
4444
```

55555
666666

```
#include<iostream>
using namespace std;
int main(){

      /*  Read input as specified in the question.
           * Print output as specified in the question.
           */
   int N;
     cin >> N;
     int i =1;
     while(i <= N){
        int j =1;
        while(j <= i){

            cout<< i;
            j++;
        }
        cout << endl;

        i++;
    }
}
```

4-Tut : Code : Reverse Number Pattern

Send Feedback

Print the following pattern for the given N number of rows.

Pattern for N = 4

1
21
321
4321

Input format : Integer N (Total no. of rows)
Output format : Pattern in N lines
Constraints 0 <= N <= 50
Sample Input 1: 5
Sample Output 1:

1
21
321
4321
54321

Sample Input 2: 6
Sample Output 2:

1

21
321
4321
54321
654321

```cpp
#include<iostream>
using namespace std;


int main(){

        /*  Read input as specified in the question.
            * Print output as specified in the question.
            */
   int N;
     cin >> N;
     int i =1;
     while(i <= N){
        int j =1;
        int p =i;
        while(j <= i){
            cout << p;
            p--;
            j++;
        }
        cout << endl;
        i++;
    }
}
```

5-Tut : Code : Alpha Pattern

Send Feedback

Print the following pattern for the given N number of rows.

Pattern for N = 3

A
BB
CCC

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints 0 <= N <= 26

Sample Input 1: 7

Sample Output 1:

A
BB
CCC
DDDD

EEEEE
FFFFFF
GGGGGGG

Sample Input 2:6

Sample Output 2:

A
BB
CCC
DDDD
EEEEE

FFFFFF

```
#include<iostream>
using namespace std;
int main(){

    /*  Read input as specified in the question.
     * Print output as specified in the question.
     */
    int N ;
    cin >> N;
    int i =1;
    while(i <= N){
        int j =1;
        char ch = 'A' + i-1;
        while(j <= i){
            cout << ch;
            j++;
        }
        cout << endl;
        i++;
    }
}
```

6-Tut : Code : Character Pattern

Send Feedback

Print the following pattern for the given N number of rows.

Pattern for N = 4

A
BC
CDE
DEFG

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints0 <= N <= 13

Sample Input 1:5
Sample Output 1:
A
BC
CDE
DEFG
EFGHI
Sample Input 2:6
Sample Output 2:
A
BC
CDE
DEFG
EFGHI
FGHIJK

```
#include<iostream>
using namespace std;
int main(){

        /*  Read input as specified in the question.
            * Print output as specified in the question.
            */
    int N;
     cin >> N;
     int i =1;
     while(i <= N){
        int j =1;
        char ch = 'A'+i-1;
        while(j <= i){
          char c = ch + j-1;
          cout << c;
           //ch = ch + j-1;
          // cout << ch;
           j++;

        }
        cout << endl;

        i++;
    }
}
```

7-Tut : Code : Interesting Alphabets

Send Feedback

Print the following pattern for the given number of rows.

Pattern for N = 5

E
DE
CDE
BCDE
ABCDE

Input format : N (Total no. of rows)

Output format : Pattern in N lines

Constraints

0 <= N <= 26

Sample Input 1:

8

Sample Output 1:

H
GH
FGH
EFGH
DEFGH
CDEFGH
BCDEFGH
ABCDEFGH

Sample Input 2:

7

Sample Output 2:

G
FG
EFG
DEFG
CDEFG
BCDEFG
ABCDEFG

```
#include<iostream>
using namespace std;
int main() {

    /*  Read input as specified in the question.
     * Print output as specified in the question.
     */
    int N;
    cin >> N;
    int i =1;
    while(i <= N){
        int j =1;
        char start = 'A' + N- i;
        while(j <= i){
            char c = start + j -1;
            cout << c;
            j++;
        }
```

```
19.         cout << endl;
20.         i++;
21.     }
22. }
```

L5 : Pattern 2 Practice Questions

1-Tut : Code : Mirror Number Pattern

Send Feedback

Print the following pattern for the given N number of rows.

Pattern for N = 4

```
···1
··12
·123
1234
```

The dots represent spaces.

Input format : Integer N (Total no. of rows)
Output format : Pattern in N lines
Constraints 0 <= N <= 50
Sample Input 1: 3
Sample Output 1:

```
  1
 12
123
```

Sample Input 2: 4
Sample Output 2:

```
   1
  12
 123
1234
```

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


int main(){

    /*  Read input as specified in the question.
        * Print output as specified in the question.
        */
  int N;
   cin >> N;
   int i = 1;
   while(i <= N){

     int spaces = 1;
```

```
        while(spaces <= N-i){
            cout << " ";
            spaces++;
        }

        int n =1;
        while(n <= i){
            cout << n;
            n ++;
        }
        cout << endl;
        i++;
    }
}
```

2-Tut : Code : Inverted Number Pattern

Send Feedback

Print the following pattern for the given N number of rows.

Pattern for N = 4

```
4444
333
22
1
```

Input format : Integer N (Total no. of rows)

Output format : Pattern in N lines

Constraints : 0 <= N <= 50

Sample Input 1: 5

Sample Output 1:

```
55555
4444
333
22
1
```

Sample Input 2: 6

Sample Output 2:

```
666666
55555
4444
333
22
1
```

```
#include<iostream>
using namespace std;
int main(){

    /*  Read input as specified in the question.
```

```
6.              * Print output as specified in the question.
7.              */
8.     int N ;
9.       cin >> N;
10.     int i =1;
11.     while(i <= N){
12.        int p = N-i+1;
13.        int k =1;
14.
15.        while(k <= p){
16.           cout << p;
17.           k++;
18.        }
19.        cout << endl;
20.        i++;
21.     }
22. }
```

3-Tut : Code : Star Pattern

Send Feedback

Print the following pattern

Pattern for N = 4

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

The dots represent spaces.

Input Format : N (Total no. of rows)
Output Format : Pattern in N lines
Constraints : 0 <= N <= 50
Sample Input 1 : 3
Sample Output 1 :

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

Sample Input 2 : 4
Sample Output 2 :

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

*******
```

```cpp
#include<iostream>
using namespace std;
int main(){

      /*  Read input as specified in the question.
           * Print output as specified in the question.
           */
   int N ;
    cin >> N;
    int i =1;
    while(i <= N){
       int spaces = 1;
       while(spaces <= N-i){
          cout << " ";
          spaces ++ ;
       }
       int star = 1;
       while(star <= 2*i-1){
          cout << "*";
          star ++;
       }
       cout << endl;
       i++;
   }
}
```

4-Tut : Code : Triangle of Numbers

Send Feedback

Print the following pattern for the given number of rows.

Pattern for N = 4

```
···1
··232
·34543
4567654
```

The dots represent spaces.

Input format : Integer N (Total no. of rows)
Output format : Pattern in N lines
Constraints : 0 <= N <= 50
Sample Input 1: 5
Sample Output 1:

```
    1
   232
  34543
 4567654
567898765
```

Sample Input 2:4

Sample Output 2:

```
   1
  232
 34543
4567654
```

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

int main() {
    /*  Read input as specified in the question.
     * Print output as specified in the question.
     */
    int N;
    cin >> N;
    int i =1;
    while(i <= N){
        int space= 1;
        while(space <= N-i){
            cout << " ";
            space ++;
        }

        int k =i;
        int j = 1;
        while(j <= i){
            cout << k;
            j++;
            k++;
        }

        j =1;
        k = 2*(i-1);
        while(j <= i-1){
            cout << k;
            k--;
            j++;
        }
        cout << endl;
        i++;
    }
}

```

5-Tut : Code : Diamond of stars

Send Feedback

Print the following pattern for the given number of rows.

Note: N is always odd.

Pattern for N = 5

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

The dots represent spaces.

Input format :

N (Total no. of rows and can only be odd)

Output format :

Pattern in N lines

Constraints :

1 <= N <= 49

Sample Input 1:

5

Sample Output 1:

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

Sample Input 2:

3

Sample Output 2:

```
 *
***
 *
```

```
#include<iostream>
using namespace std;
int main() {

    /*  Read input as specified in the question.
     * Print output as specified in the question.
     */
    int N;
    cin >> N;
    int i = 1;

    int n1 = N /2 + 1;
    while(i <= n1){
        int space = 1;
        while(space <= n1-i){
            cout << " ";
            space ++;
        }
        int star = 1;
        while(star <= 2*i -1){
            cout << "*";
            star ++ ;
        }
        cout << endl;
        i++;
    }

    i = N/2;
    while(i >= 1){

        int space = 1;
        while(space <= n1-i){
            cout << " ";
            space ++;
        }

        int star = 1;
        while(star <= 2*i -1){
            cout << "*";
            star ++ ;
        }
        cout << endl;
        i--;
     }
}
```

L6: Operator For Loops Practice Questions

1-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main()
{
        int x, y = 1;
        x = 10;
        if (x != 10 && x / 0 == 0)
           cout << y;
        else
            cout << ++y;

}
```

Options

This problem has only one correct answer

2
1
Error
None of these

Correct Answer: 2

Solution Description

Even though you get the output as '2', you'll also get a "division by zero" warning. The reason you only get a warning but not an error is short-circuit evaluation. In the if statement, once the first condition (x != 10) is evaluated to be false, then the second condition (x / 0 == 0) is not even executed because the result of the overall condition (x != 10 && x / 0 == 0) will be false. Even if the second condition were true, it would not change the overall result of the two conditions.

However, if the value of x or the first condition itself is changed so that it evaluates to true, then you will get an error because in that case the second condition would also be checked (or executed).

2-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main()
{
   int x = 15;
   int y = x++;
   int z = ++x;
   cout << y << " " << z;

}
```

Options

This problem has only one correct answer

15 16
16 17
15 17
16 16

Correct Answer : c

3-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main()
{
        int g = 3;
        cout << (++g * 8);

}
```

Answer

Type here : 32

Correct Answer

4-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main()
{
   int x = 10;
   int y = 20;
   if(x++ > 10 && ++y > 20 ){
   cout << "Inside if ";
   } else{
   cout << "Inside else ";
   }
   cout << x << " " << y;

}
```

Options

This problem has only one correct answer

Inside if 11 21
Inside if 10 21
Inside else 11 20
Inside else 11 21

Correct Answer: c

5-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main()
{
   int x = 10;
   int y = 20;
   if(x++ > 10 || ++y > 20 ){
   cout << "Inside if ";
   } else{
   cout << "Inside else ";
   }
   cout << x << " " << y;

}
```

Options

This problem has only one correct answer

Inside if 11 21
Inside if 10 21
Inside else 11 20
Inside else 11 21

Correct Answer : A

6-Tut : What is the output

Send Feedback

What will be the output the following code ?

```
for(int i = 0; i < 5; i = i + 1){
   cout << i << " ";
   i = i + 1;
}
```

Options

This problem has only one correct answer

0 1 2 3 4
0 2 4
1 3
1 2 3 4 5

Correct Answer : B

7-Tut : What is the output

Send Feedback

What will be the output of the following code?

```
for(int i = 1; i < 5; i = i + 1){
    cout << i << " ";
    i = i - 1;
}
```

Options

This problem has only one correct answer

1 2 3 4
Infinite 1s
Compilation error
None of these

Correct Answer B

8-Tut : What is the output

Send Feedback

What will be the output ?

```
for(int i = 0; i < 2; i = i + 1) {
     for(int j = 0; j < 2; j = j + 1) {
          if (j == 1)
               break;
          cout << j << " ";
      }

 }
```

Options

This problem has only one correct answer

0 1 0 1
0 0 0 0
0 0
0 1

Correct Answer : C

9-Tut : What is the output

Send Feedback

What will be the output ?

```
for(int i = 0; i < 5; i = i + 1) {
     if(i == 2)
          continue;
    cout << i << " ";
 }
```

Options

This problem has only one correct answer

0 1
0 1 3 4
0 1 2 3 4
0 1 2 3 4 5

Correct Answer : B

10-Tut : Nth Fibonacci Number

Send Feedback

Nth term of Fibonacci series F(n), where F(n) is a function, is calculated using the following formula -

F(n) = F(n-1) + F(n-2),
Where, F(1) = F(2) = 1
Provided N you have to find out the Nth Fibonacci Number.

Input Format :

The first line of each test case contains a real number 'N'.

Output Format :

For each test case, return its equivalent Fibonacci number.

Constraints:

1 <= N <= 10000
Where 'N' represents the number for which we have to find its equivalent Fibonacci number.
Time Limit: 1 second
Sample Input 1:6

Sample Output 1:8

Explanation of Sample Input 1:

Now the number is '6' so we have to find the "6th" Fibonacci number
So by using the property of the Fibonacci series i.e
[1, 1, 2, 3, 5, 8]
So the "6th" element is "8" hence we get the output.

```
#include<iostream>
using namespace std;
int fib(int n){
    if(n == 1 || n == 2){
        return 1;
    }
    return fib(n-1) + fib(n-2);

}

int main(){
    //Write your code here.
    int N;
    cin >> N;
```

```
15.     cout << fib(N);
16. }
```

11-Tut : Skip iteration

Send Feedback

Which of these jump statements can skip processing the remainder of code in its body for a particular iteration ?

Options

This problem has only one correct answer

break
return
continue

Correct Answer : C

Solution Description

####"break" is used to exit from the current loop.
####"return" statement is used to exit from the current function.

####"continue" is used to skip the current iteration of a loop and continue with the next iteration.

12-Tut : What is the output

Send Feedback

```
int i = 1;
while(i < 5) {
   if(i == 3) {
      break;
   }
   cout << i << " ";
   i++;
}
```

Options

This problem has only one correct answer

1 2 3 4
1 2
1 2 4
Infinite loop

Correct Answer : B

13-Tut : What is the output

Send Feedback

```
int i = 1;
while(i < 5) {
   if(i == 3) {
      continue;
   }
   cout << i << " ";
   i++;
```

```
}
```

Options

This problem has only one correct answer

1 2 3 4
1 2
1 2 4
1 2 Infinite loop

Correct Answer : D

14-Tut : What is the output

Send Feedback

```
int i = 1;
while(i < 3) {
    int j = 1;
    while(j < 5) {
        if(j == 3) {
            break;
        }
        cout << j << " ";
        j++;
    }
    i++;

}
```

Options

This problem has only one correct answer

1 2 1 2
1 2
1 2 4 1 2 4
Infinite loop

Correct Answer : A

15-Tut : What is the output

Send Feedback

```
int i = 1;
while(i < 3) {
    int j = 0;
    while(j < 5) {
        j++;
        if(j == 3) {
            continue;
        }
        cout << j << " ";
    }
    i++;

}
```

Options

This problem has only one correct answer

1 2 1 2
1 2 3 4 1 2 3 4
1 2 4 5 1 2 4 5
1 2 4 1 2 4

Correct Answer : C

16-Tut : All Prime Numbers

Send Feedback

Given an integer N, print all the prime numbers that lie in the range 2 to N (both inclusive).

Print the prime numbers in different lines.

Input Format :

Integer N

Output Format :

Prime numbers in different lines

Constraints :

1 <= N <= 100

Sample Input 1: 9

Sample Output 1:

2
3
5
7

Sample Input 2: 20

Sample Output 2:

2
3
5
7
11
13
17
19

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

int main(){

    /* Read input as specified in the question.
            * Print output as specified in the question.
            */
    int N;
     cin >> N;
     int CN = 2;
     while(CN <= N){
```

```
13.         int div = 2;
14.         bool divided = false;
15.
16.         while(div < CN){
17.             if(CN % div == 0){
18.                 divided = true;
19.                 break;
20.             }
21.             div++;
22.         }
23.         if(!divided){
24.             cout << CN<< endl;
25.         }
26.
27.         CN++;
28.     }
29. }
```

17-Tut : Check error

Send Feedback

Will the following code generate error ?

```
#include <iostream>
using namespace std;
int main() {
    int a = 10;
    if(a > 5) {
        int b = 10;
    }
    cout << b << endl;

}
```

Options

This problem has only one correct answer

Yes
No

Correct Answer : A

18-Tut : Check error

Send Feedback

Will following code generate error ?

```
#include <iostream>
using namespace std;
int main() {
    int a = 10;
    if(a > 5) {
```

```
        int a = 100;
    }
    cout << a << endl;

}
```

Options

This problem has only one correct answer

Yes
No

Correct Answer : B

19-Tut : Fill the output

Send Feedback

What is the output ?

```
#include <iostream>
using namespace std;
int main() {
    int a = 10;
    if(a > 5) {
        int a = 100;
    }
    else {
        int a = 110;
    }
    cout << a << endl;

}
```

Answer

Type here : 10

Correct Answer

20-Tut : Check the error

Send Feedback

Will following code generate error ?

```
#include <iostream>
using namespace std;
int main() {
    for(int i = 0; i < 3; i++) {
        cout << i << " ";
    }
    cout << i << " ";
}
```

Options

This problem has only one correct answer

Yes
No

Correct Answer : A

21-Tut : What is the output

Send Feedback

What is the output ?

```
#include <iostream>
using namespace std;
int main() {
   int a = 10;
   while(a > 5) {
      int a = 1;
      cout << a << " ";
      a--;
   }
}
```

Options

This problem has only one correct answer

10 9 8 7 6
1 1 1 1 1
Error
Infinite loop

Correct Answer : D

22-Tut : Count Characters

Send Feedback

Write a program to count and print the total number of characters (lowercase english alphabets only), digits (0 to 9) and white spaces (single space, tab i.e. '\t' and newline i.e. '\n') entered till '$'.

That is, input will be a stream of characters and you need to consider all the characters which are entered till '$'.

Print count of characters, count of digits and count of white spaces respectively (separated by space).

Input Format :

A stream of characters terminated by '$'

Output Format :

3 integers i.e. count_of_characters count_of_digits count_of_whitespaces (separated by space)

Sample Input :

abc def4 5$

Sample Output :

6 2 2

Sample Output Explanation :

Number of characters : 6 (a, b, c, d, e, f)

Number of digits : 2 (4, 5)

Number of white spaces : 2 (one space after abc and one newline after 4)

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

int main(){

    /* Read input as specified in the question.
            * Print output as specified in the question.
            */
    int ch =0;
    int digit =0;
    int space =0;


    char c;
    c = cin.get();
    while(c!= '$'){

        if(c >= 'a' && c <= 'z'){
            ch++;
        }

        if(c >= '0' && c <= '9'){
            digit++;
        }
        if(c == ' ' || c == '\n' || c == '\t'){
            space++;
        }
        c = cin.get();
    }
      cout << ch << " " << digit << " " << space <<endl;

}
```

23-Ass : Sum or Product

Send Feedback

Write a program that asks the user for a number N and a choice C. And then give them the possibility to choose between computing the sum and computing the product of all integers in the range 1 to N (both inclusive).

If C is equal to -

1, then print the sum
2, then print the product
Any other number, then print '-1' (without the quotes)

Input format :

Line 1 : Integer N
Line 2 : Choice C

Output Format :

Sum or product according to user's choice

Constraints :

1 <= N <= 12

Sample Input 1 :

10
1

Sample Output 1 :

55

Sample Input 2 :

10
2

Sample Output 2 :

3628800

Sample Input 3 :

10
4

Sample Output 3 :

-1

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

int main() {
        // Write your code here
        int N;
    char c;
    cin >> N >> c;
    if(c == '1'){
        cout << (N*(N+1))/2;

    }else if(c == '2'){
```

```
        long int mul = 1;
        int i =1;
        while(i <= N){
            mul = mul * i;
            i++;
        }
        cout << mul;
    }else{
        cout << -1;
    }
}
```

24-Ass : Terms of AP

Send Feedback

Write a program to print first x terms of the series 3N + 2 which are not multiples of 4.

Input format : Integer x
Output format : Terms of series (separated by space)
Constraints : 1 <= x <= 1,000
Sample Input 1 : 10
Sample Output 1 :
5 11 14 17 23 26 29 35 38 41
Sample Input 2 : 4
Sample Output 2 : 5 11 14 17

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

int main() {
        // Write your code here
        int x;
    cin >> x;
    int count = 1;
    for(int i =1; count <= x; i++){
        if( ((3*i)+2) % 4 != 0){
            cout << (3*i)+2<<" ";
            count ++;
        }


    }
}
```

25-Ass : Reverse of a number

Send Feedback

Write a program to generate the reverse of a given number N. Print the corresponding reverse number.

Note : If a number has trailing zeros, then its reverse will not include them. For e.g., reverse of 10400 will be 401 instead of 00401.

Input format :

Integer N

Output format :

Corresponding reverse number

Constraints:0 <= N < 10^8

Sample Input 1 :

1234

Sample Output 1 :

4321

Sample Input 2 :

1980

Sample Output 2 :

891

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

int main() {
        // Write your code here
        int N;
    cin >> N;
    int rev = 0;
    while(N!=0){
        int r = N % 10;
        rev = rev *10 +r;
        N = N/10;
    }

     cout << rev;

}
```

26-Ass : Binary to decimal

Send Feedback

Given a binary number as an integer N, convert it into decimal and print.

Input format :

An integer N in the Binary Format

Output format :

Corresponding Decimal number (as integer)

Constraints :

0 <= N <= 10^9

Sample Input 1 :

1100

Sample Output 1 :

12

Sample Input 2 :

111

Sample Output 2 :

7

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

int main() {
        // Write your code here
        int N;
    cin >> N;
    int pv=1;
    int decimal = 0;
    while(N != 0){
        int r = N % 10;
        decimal = decimal + r*pv;
        pv = pv*2;
        N = N/10;
    }
    cout << decimal;
}
```

27-Ass : Decimal to Binary

Send Feedback

Given a decimal number (integer N), convert it into binary and print.

The binary number should be in the form of an integer.

Note: The given input number could be large, so the corresponding binary number can exceed the integer range. So you may want to take the answer as long for CPP and Java.

Note for C++ coders: Do not use the inbuilt implementation of "pow" function. The implementation of that function is done for 'double's and it may fail when used for 'int's, 'long's, or 'long long's. Implement your own "pow" function to work for non-float data types.

Input format : Integer N

Output format : Corresponding Binary number (long)

Constraints : 0 <= N <= 10^5

Sample Input 1 : 12

Sample Output 1 : 1100

Sample Input 2 : 7

Sample Output 2 : 111

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

int main() {
        // Write your code here
        int N;
    cin >> N;
     long int answer =0;
    long int pv =1;

    while(N>0){
        int r = N % 2;
        answer = answer + r * pv;
        pv = pv*10;
        N = N/2;
    }
    cout << answer;
    return 0;
}
```

28-Ass : Square Root (Integral)

Send Feedback

Given a number N, find its square root. You need to find and print only the integral part of square root of N.

For eg. if number given is 18, answer is 4.

Input format : Integer N

Output Format :

Square root of N (integer part only)

Constraints :

0 <= N <= 10^8

Sample Input 1 : 10

Sample Output 1 : 3

Sample Input 2 :4
Sample Output 2 :2

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

int main() {
        // Write your code here
        int N;
    cin >> N;
    if(N == 0){
        cout << 0;
    }else{
        int i =1;
        while(i*i < N){
            i++;
        }
        if(i*i == N){
            cout<<i;
        }else
          if(i*i > N){
            cout << i-1;
        }
    }

}
```

29-Ass : Check Number sequence

Send Feedback

You are given S, a sequence of n integers i.e. S = s1, s2, ..., sn. Compute if it is possible to split S into two parts : s1, s2, ..., si and si+1, si+2, ….., sn (0 <= i <= n) in such a way that the first part is strictly decreasing while the second is strictly increasing one.

Note : We say that x is strictly larger than y when x > y.
So, a strictly increasing sequence can be 1 4 8. However, 1 4 4 is NOT a strictly increasing sequence.
That is, in the sequence if numbers are decreasing, they can start increasing at one point. And once the sequence of numbers starts increasing, they cannot decrease at any point further.

Sequence made up of only increasing numbers or only decreasing numbers is a valid sequence. So, in both the cases, print true.
You just need to print true/false. No need to split the sequence.
Input format :
Line 1 : Integer 'n'
Line 2 and Onwards : 'n' integers on 'n' lines(single integer on each line)
Output Format :"true" or "false" (without quotes)
Constraints :1 <= n <= 10^7

Sample Input 1 :

5
9
8
4
5
6

Sample Output 1 :true

Sample Input 2 :

3
1
2
3

Sample Output 2 :true

Sample Input 3 :

3
8
7
7

Sample Output 3 :false

Explanation for Sample Format 3 :

8 7 7 is not strictly decreasing, so output is false.

Sample Input 4 :

6
8
7
6
5
8
2

Sample Output 4 :false

Explanation for Sample Input 4 :

The series is :
8 7 6 5 8 2

It is strictly decreasing first (8 7 6 5). Then it's strictly increasing (5 8). But then it starts strictly decreasing again (8 2). Therefore, the output for this test case is 'false'

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

int main() {
    // Write your code here
    int N;
    cin >> N;
    int n1, n2;
    cin >> n1 >>n2;
```

```
10.
11.     bool isdec = true;
12.     if(n2 > n1){
13.         isdec = false;
14.     }
15.     int flag =1;
16.    int i =1;
17.     while(i <= N-2){
18.
19.         if(n1 == n2){
20.             cout << "false";
21.             flag = 0;
22.             break;
23.         }
24.
25.         n1 = n2;
26.         cin >> n2;
27.
28.         if(isdec && n2 < n1){
29.             i++;
30.             continue;
31.         }
32.
33.         if(!isdec && n2 > n1 ){
34.             i++;
35.             continue;
36.         }
37.
38.         if(isdec && n2 > n1){
39.             isdec = false;
40.             i++;
41.             continue;
42.         }
43.         if(!isdec && n2 < n1){
44.             flag =0;
45.             cout << "false";
46.             break;
47.         }
48.     }
49.
50.     if(flag == 1){
51.         cout << "true";
52.     }
53.
54. }
```

L7A : Functions Practice Questions

1-Tut : Return type

Send Feedback

What is the return type of a method that does not returns any value ?

Options

This problem has only one correct answer

int
double
char
void

Correct Answer : D

Solution Description

####The functions which doesn't return any value, their return type is "void".

2-Tut : Return type

Send Feedback

Let’s say the problem is - You will be given two numbers(both integers) and you need to return their sum.

For this problem, what should be the return type of function -

Options

This problem has only one correct answer

int
boolean
char
void

Correct Answer: A

Solution Description

####We need to return the sum of two integers, which is again an integer. So the sum that we want to return is of type "int". Hence the return type should be "int" for this function.

3-Tut : What is the output

Send Feedback

What will be the output of the following code ?

```
  void func(int a, int b){
   cout << (a + b);
   }

   int main() {
   int a = 7;
```

```
    func(a, 12);
    }
```

Answer

Type here : 19

Correct Answer

4-Tut : What is the output

Send Feedback

What will be the output of the following code ?

```
void demo(int a, int b){
    cout << a << " " << b;
}

int main() {
    int a = 5;
    int b = 15;
    demo(a);
}
```

Options

This problem has only one correct answer

5 0

Compilation Error

5 15

None of these

Correct Answer : B

5-Tut : Fahrenheit to Celsius Table

Send Feedback

Given three values - Start Fahrenheit Value (S), End Fahrenheit value (E) and Step Size (W), you need to convert all Fahrenheit values from Start to End at the gap of W, into their corresponding Celsius values and print the table.

Input Format : 3 integers - S, E and W respectively

Output Format :

Fahrenheit to Celsius conversion table. One line for every Fahrenheit and Celsius Fahrenheit value. Fahrenheit value and its corresponding Celsius value should be separate by tab ("\t")

Constraints :

0 <= S <= 1000

0 <= E <= 1000

0 <= W <= 1000

Sample Input 1:

0

100

20

Sample Output 1:

0 -17

20 -6
40 4
60 15
80 26
100 37

Sample Input 2:

120
200
40

Sample Output 2:

120 48
160 71
200 93

Explanation for Sample Output 2 :

Start value is 120, end value is 200 and step size is 40. Therefore, the values we need to convert are 120, 120 + 40 = 160, and 160 + 40 = 200.
The formula for converting Fahrenheit to Celsius is:
Celsius Value = (5/9)*(Fahrenheit Value - 32)
Plugging 120 into the formula, the celsius value will be (5 / 9)*(120 - 32) => (5 / 9) * 88 => (5 * 88) / 9 => 440 / 9 => 48.88

But we'll only print 48 because we are only interested in the integral part of the value.

```
1.  void printTable(int start, int end, int step) {
2.      /* Don't write main().
3.       * Don't read input, it is passed as function argument.
4.       * Print output and don't return it.
5.       * Taking input is handled automatically.
6.       */
7.      int cf = start;
8.      while(cf <= end){
9.          int result = (5.0/9)*(cf-32);
10.         cout << cf << "\t" << result;
11.         cout<<endl;
12.         cf = cf + step;
13.     }
14.
15. }
```

6-Tut : Fibonacci Number

Send Feedback

Given a number N, figure out if it is a member of fibonacci series or not. Return true if the number is member of fibonacci series else false.

Fibonacci Series is defined by the recurrence

F(n) = F(n-1) + F(n-2)
where F(0) = 0 and F(1) = 1

Input Format : Integer N
Output Format : true or false
Constraints : 0 <= n <= 10^4
Sample Input 1 : 5
Sample Output 1 : true
Sample Input 2 : 14
Sample Output 2 : false

```
bool checkMember(int n){

    /* Don't write main().
     * Don't read input, it is passed as function argument.
     * Return output and don't print it.
     * Taking input and printing output is handled automatically.
     */
     int f0 = 0, f1 = 1, f2 = 0;
      bool fib = false;

     while(f2 <= n){

        if(f2 == n){
            fib = true;
            break;
        }

        f2 = f0 +f1;
        f0 = f1;
        f1 = f2;
     }

    return fib;
}
```

7-Tut : Check the error

Send Feedback

Will the following code generate any error ?

```
#include <iostream>
using namespace std;
void func(int a) {
    int b = a;
    b = b + 10;
}

int main() {
    int a = 10;
    func(a);
    cout << b << endl;
```

```
}
```

Options

This problem has only one correct answer

Yes
No

Correct Answer : A

8-Tut : Check the error

Send Feedback

Will the following code generate any error ?

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

void func(int a) {
   int b = 10;
   a = b + 10;
   cout << a << " ";
}

int main() {
   int a = 10;
   func(a);
   cout << a << " ";
}
```

Options

This problem has only one correct answer

Yes
No

Correct Answer : B

9-Tut : Fill the output

Send Feedback

What will be the output of the following code ?

```
void doubleValue(int a) {
   a = a * 2;
}
int main() {
   int a = 8;
   doubleValue(a);
   cout << a;

}
```

Answer

Type here : 8

Correct Answer

10-Tut : Fill the output

Send Feedback

What will be the output of the following code ?

```
int func(int a){
    a += 10;
    return a;
}
int main() {
    int a = 5;
    func(a);
    cout << a;

}
```

Answer

Type here : 5

Note : why not give errors if we are not storing returned values ?

https://stackoverflow.com/questions/28142017/what-happens-when-i-call-a-function-without-assigning-its-returned-value-to-a-va

Correct Answer

11-Tut : Fill the output

Send Feedback

What will be the output of the following code ?

```
int square(int a){
    int ans = a * a;
    return ans;
}

int main() {
    int a = 4;
    a = square(a);
    cout << a;

}
```

Answer

Type here : 16

Correct Answer

L7 : Test 1 Questions

Q1: If & Else

Send Feedback

Find the output.

```
int p = 100;
if(p > 20) {
    if(p < 20) {
        printf("coding");
    }
} else {
    printf("ninjas");

}
```

Options

This problem may have one or more correct answers

coding
ninjas
codingninjas
No Output

Correct Answer: D

Q2: Order the code

Send Feedback

Write the correct order(s) of statements.

We want to print "Coding Ninjas" 10 times.

```
1.  while(x <= 10) {
2.  int x = 1;
3.  x += 1;
4.  cout<<"Coding Ninjas"<<endl;
5.  }
```

Options

This problem may have one or more correct answers

2 3 1 4 5
2 1 3 4 5
2 1 4 3 5
1 2 4 3 5

Correct Answer : B , C

Q3: Predict the output

Send Feedback

Figure out the output

```
int x = 10;
while(x >= 0) {
   x = x - 1;
   print(x);
   x--;

}
```

Options

This problem may have one or more correct answers

97531
97531-1
10864
10987654321

Correct Answer : B

Q4 : And/Or Operator

Send Feedback

Please select the correct statement(s) about && and || operators.

Options

This problem may have one or more correct answers

a && b is true if either a or b is true
(a || b) && c is true if c is true and either a or b is true
a && b is false if both a and b are true
a || b is true if either a or b is true

Correct Answer : B , D

Q5 : Pyramid Number Pattern

Send Feedback

Print the following pattern for the given number of rows.

Pattern for N = 4

```
 1
 212
 32123
4321234
```

Input format : N (Total no. of rows)
Output format : Pattern in N lines

Sample Input :

5

Sample Output :

```
    1
   212
  32123
 4321234
543212345
```

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

int main(){

  // Write your code here
int N;
    cin >> N;
    int i =1;
    while(i <= N){
        int space = 1;
        while(space <= N-i){
            cout << " ";
            space++;
        }
        int num =i;
        while(num >= 1){
            cout << num;
            num--;
        }
        num =2;
        int k =1;
        while(k<= i-1){
            cout << num;
            num++;
            k++;
        }
        cout<<endl;

        i++;
    }

}
```

Q6 : Number Star Pattern

Send Feedback

Print the following pattern for a given number of rows.

Input format :

Line 1 : N (Total number of rows)

Sample Input :

```
5
```

Sample Output :

```
1234554321
1234**4321
123****321
12******21

1********1
```

```
#include<iostream>
using namespace std;
int main(){

  // Write your code here
int N;
    cin >> N;
    int i =1;
    while(i <= N){
        int j =1;
        while(j <= N-i+1){
            cout << j;
            j++;
        }
        int stars = 1;
        while(stars <= 2*(i-1)){
            cout << "*";
            stars ++;
        }
        j = N-i+1;
        while(j >= 1){
            cout << j;
            j--;

        }
        cout << endl;
        i++;
    }
}
```

Q7 : Second Largest

Send Feedback

Take input a stream of n integer elements, find the second largest element present.

The given elements can contain duplicate elements as well. If only 0 or 1 element is given, the second largest should be INT_MIN (- 2^31).

Input format :

Line 1 : Total number of elements (n)

Line 2 : N elements (separated by space)

Sample Input 1:

```
4
3 9 0 9
```

Sample Output 1:

```
3
```

Sample Input 2 :

```
2
4 4
```

Sample Output 2:

```
-2147483648
```

Sample Output Explanation:

Since both the elements are equal here, hence second largest element is INT_MIN i.e. (-2^31)

```
#include<iostream>
using namespace std;
#include <climits>
int main(){

    // Write your code here
    int N;
    long int min = -2147483648;
    cin >> N;
    int a[N+1];
    for(int i =1; i<= N; i++){
        cin >> a[i];
    }
    int fl = a[1];
    int sl = min;
    int count = 0;
    if(N == 2 && a[1] == a[2]){
        cout << min;
    }
    else if(N == 0 || N == 1){
        cout << min;
    }
    else if(N >= 2){
          for (int i =1; i <= N; i ++)
```

```
25.             {
26.                 if(a[i]> fl)
27.                 {
28.                     sl = fl;
29.                     fl = a[i];
30.                 }
31.                 else if(a[i] > sl&&a[i]!=fl)//
32.                 {
33.                     sl = a[i];
34.                 }
35.
36.             }
37.             cout<<sl;
38.
39.         }
40. }
```

L8 : Arrays Practice Questions

1-Tut : Array declaration

Send Feedback

Which of the following correctly declares an array of size 10 ?

Options

This problem has only one correct answer

int array[10];
int array;
array{10};
array array[10];

Correct Answer : A

2-Tut : What is the output

Send Feedback

What will be the output of the following code ?

```
int arr[100];
cout << arr[0];
```

Options

This problem has only one correct answer

0
Garbage value
Error

Correct Answer : B

3-Tut : Fill the output

Send Feedback

What is the index number of the last element of an array with 5 elements ?

Answer

Type here : 4

Correct Answer

4-Tut : Access element

Send Feedback

Which of the following accesses the third element stored in array?

Options

This problem has only one correct answer

array[3]
array[2]
array(3)
array

Correct Answer : B

5-Tut : Array Sum

Send Feedback

Given an array of length N, you need to find and print the sum of all elements of the array.

Input Format :

Line 1 : An Integer N i.e. size of array
Line 2 : N integers which are elements of the array, separated by spaces

Output Format :

Sum

Constraints :

1 <= N <= 10^6

Sample Input :

3
9 8 9

Sample Output :

26

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

int arraysum(int n,int a[]){
    int sum =0;
    for(int i = 0; i<n; i++){
        sum = sum + a[i];
    }
    return sum;
}
int main(){
    /* Read input as specified in the question.
     * Print output as specified in the question.
     */
    int n;
    cin >> n;
    int a[n];

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

6-Tut : Function calling

Send Feedback

What is the correct syntax for passing array to a function -

```
void func(int a[]) {
}

int main() {
    int a[10];
    // Call function "func" and pass array a

}
```

Options

This problem has only one correct answer

func(a[]);
func(a[10]);
func(int a[10]);
func(a);

Correct Answer : D

7-Tut : What is the output

Send Feedback

```
#include <iostream>
using namespace std;
int main() {
    int a[10];
    cout << sizeof(a) << endl;
}
```

Options

This problem has only one correct answer

10
40
4
8

Correct Answer : B

8-Tut : What is the output

Send Feedback

Assume an integer takes 4 bytes and a pointer takes 8 bytes.

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

void func(int a[]) {
    cout << sizeof(a) << endl;
}

int main() {
    int a[10];
    func(a);
```

}

Options

This problem has only one correct answer

10
40
4
8

Correct Answer : D

9-Tut : Linear Search

Send Feedback

You have been given a random integer array/list(ARR) of size N, and an integer X. You need to search for the integer X in the given array/list using 'Linear Search'.

You have been required to return the index at which X is present in the array/list. If X has multiple occurrences in the array/list, then you need to return the index at which the first occurrence of X would be encountered. In case X is not present in the array/list, then return -1.

'Linear search' is a method for finding an element within an array/list. It sequentially checks each element of the array/list until a match is found or the whole array/list has been searched.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Third line contains the value of X(integer to be searched in the given array/list)

Output format :

For each test case, print the index at which X is present or -1, otherwise.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^5
-2 ^ 31 <= X <= (2 ^ 31) - 1
Time Limit: 1 sec

Sample Input 1:

1
7
2 13 4 1 3 6 28
3

Sample Output 1:

4

Sample Input 2:

```
2
7
2 13 4 1 3 6 28
9
5
7 8 5 9 5
5
```

Sample Output 2:

```
-1

2
```

```
1.  int linearSearch(int *arr, int n, int x)
2.  {
3.      //Write your code here
4.      for(int i = 0; i<n; i++){
5.          if(arr[i] == x){
6.              return i;
7.          }
8.
9.      }
10.
11.     return -1;
12. }
```

10-Tut : Arrange Numbers in Array

Send Feedback

You have been given an empty array(ARR) and its size N. The only input taken from the user will be N and you need not worry about the array.

Your task is to populate the array using the integer values in the range 1 to N(both inclusive) in the order - 1,3,.....4,2.

Note:

You need not print the array. You only need to populate it.

Input Format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first and the only line of each test case or query contains an integer 'N'.

Output Format :

For each test case, print the elements of the arra/listy separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2

0 <= N <= 10^4

Time Limit: 1sec

Sample Input 1 :

1
6

Sample Output 1 :

1 3 5 6 4 2

Sample Input 2 :

2
9
3

Sample Output 2 :

1 3 5 7 9 8 6 4 2

1 3 2

```
1.  void arrange(int *arr, int n)
2.  {
3.      //Write your code here
4.      int start = 0;
5.      int end = n-1;
6.      int val = 1;
7.      while(start < end){
8.          arr[start] = val;
9.          val++;
10.         arr[end] = val;
11.         val++;
12.         start ++;
13.         end--;
14.
15.     }
16.     if(start == end){
17.
18.         arr[start] = n;
19.     }
20. }
```

11-Tut : Swap Alternate

Send Feedback

You have been given an array/list(ARR) of size N. You need to swap every pair of alternate elements in the array/list.

You don't need to print or return anything, just change in the input array itself.

Input Format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the elements of the resulting array in a single row separated by a single space.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^5
Time Limit: 1sec

Sample Input 1:

1
6
9 3 6 12 4 32

Sample Output 1 :

3 9 12 6 32 4

Sample Input 2:

2
9
9 3 6 12 4 32 5 11 19
4
1 2 3 4

Sample Output 2 :

3 9 12 6 32 4 11 5 19

2 1 4 3

```
1.  void swapAlternate(int *arr, int size)
2.  {
3.      //Write your code here
4.      int i =0;
5.      int j = 1;
6.      while(j < size){
7.          int temp = arr[i];
8.          arr[i] = arr[j];
9.          arr[j] = temp;
10.         i=i+2;
11.         j=j+2;
12.
13.     }
14.
15. }
```

12-Ass : Find Unique

Send Feedback

You have been given an integer array/list(ARR) of size N. Where N is equal to [2M + 1].

Now, in the given array/list, 'M' numbers are present twice and one number is present only once.

You need to find and return that number which is unique in the array/list.

Note:

Unique element is always present in the array/list according to the given condition.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the array/list.
Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the unique element present in the array.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^3
Time Limit: 1 sec

Sample Input 1:

1
7
2 3 1 6 3 6 2

Sample Output 1:

1

Sample Input 2:

2
5
2 4 7 2 7
9
1 3 1 3 6 6 7 10 7

Sample Output 2:

4

10

```
1.  int findUnique(int *arr, int size)
2.  {
3.      //Write your code here
4.      int i = 0;
5.      while(i < size-1){
6.          int j = 0;
7.          int flag = 0;
8.
9.          while(j < size ){
10.
11.             if(j == i){
12.                 j++;
13.                 continue;
14.             }
15.
16.             if(arr[i] == arr[j]){
17.                flag = 1;
18.                 break;
```

```
            }

            j++;
        }

        if(flag == 0){
            return arr[i];
        }

        i++;
    }
    return arr[size-1];
}
```

13-Ass : Find Duplicate

Send Feedback

You have been given an integer array/list(ARR) of size N which contains numbers from 0 to (N - 2). Each number is present at least once. That is, if N = 5, the array/list constitutes values ranging from 0 to 3 and among these, there is a single integer value that is present twice. You need to find and return that duplicate number present in the array.

Note :

Duplicate number is always present in the given array/list.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the array/list.
Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the duplicate element in the array/list.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^3
Time Limit: 1 sec

Sample Input 1:

```
1
9
0 7 2 5 4 7 1 3 6
```

Sample Output 1:

```
7
```

Sample Input 2:

```
2
5
0 2 1 3 1
7
```

0 3 1 5 4 3 2

Sample Output 2:

1

3

```
1.  int duplicateNumber(int *arr, int size)
2.  {
3.      //Write your code here
4.      int i =0;
5.      while(i < size){
6.          int j = 0;
7.          while(j < size){
8.              if(i == j){
9.                  j++;
10.                 continue;
11.             }
12.
13.             if(arr[i] == arr[j]){
14.                 return arr[i];
15.             }
16.             j++;
17.         }
18.
19.         i++;
20.     }
21.
22. }
```

14-Ass : Array Intersection

Send Feedback

You have been given two integer arrays/list(ARR1 and ARR2) of size N and M, respectively. You need to print their intersection; An intersection for this problem can be defined when both the arrays/lists contain a particular value or to put it in other words, when there is a common value that exists in both the arrays/lists.

Note :

Input arrays/lists can contain duplicate elements.
The intersection elements printed would be in the order they appear in the first array/list(ARR1)

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the first array/list.
Second line contains 'N' single space separated integers representing the elements of the first the array/list.
Third line contains an integer 'M' representing the size of the second array/list.

Fourth line contains 'M' single space separated integers representing the elements of the second array/list.

Output format :

For each test case, print the intersection elements in a row, separated by a single space.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^5
0 <= M <= 10^5
Time Limit: 1 sec

Sample Input 1 :

2
6
2 6 8 5 4 3
4
2 3 4 7
2
10 10
1
10

Sample Output 1 :

2 4 3
10

Sample Input 2 :

1
4
2 6 1 2
5
1 2 3 4 2

Sample Output 2 :

2 1 2

Explanation for Sample Output 2 :

Since both input arrays have two '2's, the intersection of the arrays also have two '2's. The first '2' of the first array matches with the first '2' of the second array. Similarly, the second '2' of the first array matches with the second '2' if the second array.

```
#include <bits/stdc++.h>
void intersection(int *input1, int *input2, int size1, int size2)
{
    //Write your code here
    int i = 0;
    while(i < size1){
        int j = 0;
        while(j < size2){
            if(input1[i] == input2[j]){
                cout << input1[i] << " ";
                input2[j] = INT_MIN;
```

```
            break;
        }

        j++;
    }


    i++;
}


}
```

15-Ass : Pair Sum

Send Feedback

You have been given an integer array/list(ARR) and a number X. Find and return the total number of pairs in the array/list which sum to X.

Note:

Given array/list can contain duplicate elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the first array/list.
Second line contains 'N' single space separated integers representing the elements in the array/list.
Third line contains an integer 'X'.

Output format :

For each test case, print the total number of pairs present in the array/list.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^3
0 <= X <= 10^9
Time Limit: 1 sec

Sample Input 1:

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

Sample Output 1:

```
7
```

Sample Input 2:

```
2
9
1 3 6 2 5 4 3 2 4
12
6
```

2 8 10 5 -2 5
10

Sample Output 2:

0
2

Explanation for Input 2:

Since there doesn't exist any pair with sum equal to 12 for the first query, we print 0.

For the second query, we have 2 pairs in total that sum up to 10. They are, (2, 8) and (5, 5).

```
1.  int pairSum(int *input, int size, int x)
2.  {
3.              //Write your code here
4.      int count = 0;
5.      int i =0;
6.      while(i< size-1){
7.          int j = i+1;
8.          while(j < size){
9.              if(input[j] == x-input[i]){
10.                 count ++;
11.             }
12.             j++;
13.         }
14.
15.
16.         i++;
17.     }
18.     return count;
19. }
```

16-Ass :Triplet Sum

Send Feedback

You have been given a random integer array/list(ARR) and a number X. Find and return the number of triplets in the array/list which sum to X.

Note :

Given array/list can contain duplicate elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the first array/list.
Second line contains 'N' single space separated integers representing the elements in the array/list.
Third line contains an integer 'X'.

Output format :

For each test case, print the total number of triplets present in the array/list.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 50
0 <= N <= 10^2
0 <= X <= 10^9
Time Limit: 1 sec

Sample Input 1:

1
7
1 2 3 4 5 6 7
12

Sample Output 1:

5

Sample Input 2:

2
7
1 2 3 4 5 6 7
19
9
2 -5 8 -6 0 5 10 11 -3
10

Sample Output 2:

0
5

Explanation for Input 2:

Since there doesn't exist any triplet with sum equal to 19 for the first query, we print 0.

For the second query, we have 5 triplets in total that sum up to 10. They are, (2, 8, 0), (2, 11, -3), (-5, 5, 10), (8, 5, -3) and (-6, 5, 11)

```
1.  int tripletSum(int *input, int size, int x)
2.  {
3.              //Write your code here
4.      int count = 0;
5.      int i = 0;
6.      while(i < size-2){
7.          int j = i+1;
8.              while(j < size -1){
9.                  int k = j+1;
10.                 while(k < size){
11.                     if(input[i] + input[j] + input[k] == x){
12.                         count ++;
13.                     }
14.
15.
16.                     k++;
17.                 }
18.
19.
20.                 j++;
```

```
            }

        i++;
    }
    return count;
}
```

17-Ass : Sort 0 1

Send Feedback

You have been given an integer array/list(ARR) of size N that contains only integers, 0 and 1. Write a function to sort this array/list. Think of a solution which scans the array/list only once and don't require use of an extra array/list.

Note:

You need to change in the given array/list itself. Hence, no need to return or print anything.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the array/list.
Second line contains 'N' single space separated integers(all 0s and 1s) representing the elements in the array/list.

Output format :

For each test case, print the sorted array/list elements in a row separated by a single space.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^5
Time Limit: 1 sec

Sample Input 1:

```
1
7
0 1 1 0 1 0 1
```

Sample Output 1:

```
0 0 0 1 1 1 1
```

Sample Input 2:

```
2
8
1 0 1 1 0 1 0 1
5
0 1 0 1 0
```

Sample Output 2:

```
0 0 0 1 1 1 1 1

0 0 0 1 1
```

```
1.  void sortZeroesAndOne(int *input, int size)
2.  {
3.      //Write your code here
4.      int i = 0;
5.      while(i < size){
6.          if(input[i] == 1){
7.              break;
8.          }
9.          i++;
10.     }
11.
12.     int j = size -1;
13.     while(input[j] == 1 && j > i ){
14.         j--;
15.     }
16.
17.
18.     while(j > i){
19.
20.         if(input[i] == 1 && input[j] == 0){
21.             input[j] = 1;
22.             input[i] = 0;
23.             i++;
24.             j--;
25.             continue;
26.         }
27.         if(input[i] == 1 && input[j] == 1){
28.             j--;
29.             continue;
30.         }
31.         if(input[i] == 0 && input[j] == 0){
32.             i++;
33.             continue;
34.         }
35.
36.         if(input[i] == 0 && input[j] == 1){
37.             i++;
38.             j--;
39.             continue;
40.         }
41.     }
42.
43. }
```

L9: Searching Sorting Practice Questions

1-Tut : Code Binary Search

Send Feedback

You have been given a sorted(in ascending order) integer array/list(ARR) of size N and an element X. Write a function to search this element in the given input array/list using 'Binary Search'. Return the index of the element in the input array/list. In case the element is not present in the array/list, then return -1.

Input format :

The first line contains an Integer 'N' which denotes the size of the array/list.
Second line contains 'N' single space separated integers representing the elements in the array/list.
Third line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow..
All the 't' lines henceforth, will take the value of X to be searched for in the array/list.

Output Format :

For each test case, print the index at which X is present, -1 otherwise.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^4
0 <= N <= 10^6
0 <= X <= 10^9
Time Limit: 1 sec

Sample Input 1:

```
7
1 3 7 9 11 12 45
1
3
```

Sample Output 1:

```
1
```

Sample Input 2:

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

Sample Output 2:

```
-1

6
```

```
1.  int binarySearch(int *input, int n, int val)
2.  {
3.      //Write your code here
4.      int s = 0;
5.      int end = n-1;
6.      int mid = (s + end) / 2;
7.
```

```
8.
9.      while(s <= end){
10.
11.         if(input[mid] == val){
12.             return mid;
13.         }
14.
15.         if(input[mid] < val){
16.             s = mid + 1;
17.             mid = (s + end) / 2;
18.             continue;
19.
20.         }
21.
22.         if(input[mid] > val){
23.             end = mid -1;
24.             mid = (s + end) / 2;
25.             continue;
26.         }
27.
28.     }
29.
30.     if( s > end ){
31.
32.         return -1;
33.     }
34.
35. }
```

2-Tut : Code Bubble Sort

Send Feedback

Provided with a random integer array/list(ARR) of size N, you have been required to sort this array using 'Bubble Sort'.

Note:

Change in the input array/list itself. You don't need to return or print the elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the array/list.
Second line contains 'N' single space separated integers representing the elements in the array/list.

Output format :

For each test case, print the elements of the array/list in sorted order separated by a single space.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^3

Time Limit: 1 sec

Sample Input 1:

1
7
2 13 4 1 3 6 28

Sample Output 1:

1 2 3 4 6 13 28

Sample Input 2:

2
5
9 3 6 2 0
4
4 3 2 1

Sample Output 2:

0 2 3 6 9

1 2 3 4

```
1.  void bubbleSort(int *input, int size)
2.  {
3.      //Write your code here
4.      for(int i = 0; i < size-1; i++){
5.
6.          for(int j = 1; j < size - i; j++){
7.
8.              if(input[j] < input[j-1]){
9.                  int temp = input[j-1];
10.                 input[j-1] = input[j];
11.                     input[j] = temp;
12.             }
13.         }
14.     }
15. }
```

3-Tut : Code Insertion Sort

Send Feedback

Provided with a random integer array/list(ARR) of size N, you have been required to sort this array using 'Insertion Sort'.

Note:

Change in the input array/list itself. You don't need to return or print the elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the array/list.
Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the elements of the array/list in sorted order separated by a single space.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^3
Time Limit: 1 sec

Sample Input 1:

1
7
2 13 4 1 3 6 28

Sample Output 1:

1 2 3 4 6 13 28

Sample Input 2:

2
5
9 3 6 2 0
4
4 3 2 1

Sample Output 2:

0 2 3 6 9

1 2 3 4

```
1.  void insertionSort(int *input, int size)
2.  {
3.      //Write your code here
4.      int i , j ,key;
5.
6.      for( i = 1; i < size; i++)
7.      {
8.          key = input[i];
9.          j = i-1;
10.
11.         while(j >= 0 && key < input[j]){
12.             input[j+1] = input[j];
13.             j--;
14.
15.         }
16.         input[j+1] = key;
17.         //correct index
18.        // shifting
19.     }
20. }
```

4-Tut : Code Merge Two Sorted Arrays

Send Feedback

You have been given two sorted arrays/lists(ARR1 and ARR2) of size N and M respectively, merge them into a third array/list such that the third array is also sorted.

Input Format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the first array/list.
Second line contains 'N' single space separated integers representing the elements of the first array/list.
Third line contains an integer 'M' representing the size of the second array/list.
Fourth line contains 'M' single space separated integers representing the elements of the second array/list.

Output Format :

For each test case, print the sorted array/list(of size N + M) in a single row, separated by a single space.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^5
0 <= M <= 10^5
Time Limit: 1 sec

Sample Input 1 :

```
1
5
1 3 4 7 11
4
2 4 6 13
```

Sample Output 1 :

```
1 2 3 4 4 6 7 11 13
```

Sample Input 2 :

```
2
3
10 100 500
7
4 7 9 25 30 300 450
4
7 45 89 90
0
```

Sample Output 2 :

```
4 7 9 10 25 30 100 300 450 500

7 45 89 90
```

```
void merge(int *arr1, int size1, int *arr2, int size2, int *ans)
{
    //Write your code here
    int i = 0, j=0, k = 0;
    while(i < size1 && j < size2){
        if(arr1[i] < arr2[j]){
            ans[k] = arr1[i];
            i++;
            k++;
            continue;
        }
        if(arr1[i] >= arr2[j]){
            ans[k] = arr2[j];
            j++;
            k++;
            continue;
        }

    }

    while(i < size1){
        ans[k]= arr1[i];
        k++;
        i++;
    }

    while(j < size2){
        ans[k] = arr2[j];
        k++;
        j++;
    }
}
```

5-Ass : Push Zeros to end

Send Feedback

You have been given a random integer array/list(ARR) of size N. You have been required to push all the zeros that are present in the array/list to the end of it. Also, make sure to maintain the relative order of the non-zero elements.

Note:

Change in the input array/list itself. You don't need to return or print the elements.
You need to do this in one scan of array only. Don't use extra space.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the array/list.
Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the elements of the array/list in the desired order separated by a single space. Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2

0 <= N <= 10^5

Time Limit: 1 sec

Sample Input 1:

1

7

2 0 0 1 3 0 0

Sample Output 1:

2 1 3 0 0 0 0

Explanation for the Sample Input 1 :

All the zeros have been pushed towards the end of the array/list. Another important fact is that the order of the non-zero elements have been maintained as they appear in the input array/list.

Sample Input 2:

2

5

0 3 0 2 0

5

9 0 0 8 2

Sample Output 2:

3 2 0 0 0

9 8 2 0 0

```
1.  void pushZeroesEnd(int *input, int size)
2.  {
3.      //Write your code here
4.      int i = 0 , k = 0;
5.      while(i < size){
6.
7.          if (input[i] != 0){
8.              input[k] = input[i];
9.              k++;
10.         }
11.         i++;
12.     }
13.
14.     while(k < size){
15.         input[k] = 0;
16.         k++;
17.     }
18. }
```

6-Ass : Rotate array

Send Feedback

You have been given a random integer array/list(ARR) of size N. Write a function that rotates the given array/list by D elements(towards the left).

Note:

Change in the input array/list itself. You don't need to return or print the elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the array/list.
Second line contains 'N' single space separated integers representing the elements in the array/list.
Third line contains the value of 'D' by which the array/list needs to be rotated.

Output Format :

For each test case, print the rotated array/list in a row separated by a single space.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^4
0 <= N <= 10^6
0 <= D <= N
Time Limit: 1 sec

Sample Input 1:

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

Sample Output 1:

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

Sample Input 2:

```
2
7
1 2 3 4 5 6 7
0
4
1 2 3 4
2
```

Sample Output 2:

```
1 2 3 4 5 6 7

3 4 1 2
```

```
1.  void rotate(int *input, int d, int n)
2.  {
3.      //Write your code here
4.      int index = d;
5.
6.      int a[n];
```

```
7.      int k = 0;
8.
9.      if(index > -1){
10.
11.
12.         for(int i = index; i < n ; i++){
13.             a[k] = input[i];
14.             k++;
15.         }
16.
17.         for(int i = 0; i < index; i++){
18.             a[k]= input[i];
19.             k++;
20.         }
21.
22.         for(int i = 0 ; i < n; i++){
23.             input[i] = a[i];
24.         }
25.
26.     }
27.
28. }
```

7-Ass : Second Largest in array

Send Feedback

You have been given a random integer array/list(ARR) of size N. You are required to find and return the second largest element present in the array/list.

If N <= 1 or all the elements are same in the array/list then return -2147483648 or -2 ^ 31(It is the smallest value for the range of Integer)

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
The first line of each test case or query contains an integer 'N' representing the size of the array/list.
The second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the second largest in the array/list if exists, -2147483648 otherwise.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^5
Time Limit: 1 sec

Sample Input 1:

1
7
2 13 4 1 3 6 28

Sample Output 1:

13

Sample Input 2:

```
1
5
9 3 6 2 9
```

Sample Output 2:

```
6
```

Sample Input 3:

```
2
2
6 6
4
90 8 90 5
```

Sample Output 3:

```
-2147483648

8
```

```
#include <bits/stdc++.h>
int findSecondLargest(int *input, int n)
{
    //Write your code here
    if(n <= 1){
        return INT_MIN;
    }

    long int sl = INT_MIN;
    int fl = input[0];

    int i = 1;
    while(i < n){
        if(input[i] > fl ){
            fl = input[i];
        }
        i++;
    }
     i = 0;
    while(i < n){
        int key = input[i];
        if(key > sl && key != fl){
            sl = key;
        }

        i++;
    }


    return sl;

}
```

8-Ass : Check Array Rotation

Send Feedback

You have been given an integer array/list(ARR) of size N. It has been sorted(in increasing order) and then rotated by some number 'K' in the right hand direction.

Your task is to write a function that returns the value of 'K', that means, the index from which the array/list has been rotated.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the array/list.
Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the value of 'K' or the index from which which the array/list has been rotated.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^5
Time Limit: 1 sec

Sample Input 1:

```
1
6
5 6 1 2 3 4
```

Sample Output 1:

```
2
```

Sample Input 2:

```
2
5
3 6 8 9 10
4
10 20 30 1
```

Sample Output 2:

```
0

3
```

```
1.  int arrayRotateCheck(int *input, int size)
2.  {
3.      //Write your code here
4.      int ans = 0;
5.
6.      for(int i = 1; i < size; i++){
7.
8.          if(input[i] < input[i-1]){
9.              ans = i;
10.             break;
11.
```

```
        }
    }

    return ans;
}
```

9-Ass : **Sort 0 1 2**

Send Feedback

You are given an integer array/list(ARR) of size N. It contains only 0s, 1s and 2s. Write a solution to sort this array/list in a 'single scan'.

'Single Scan' refers to iterating over the array/list just once or to put it in other words, you will be visiting each element in the array/list just once.

Note:

You need to change in the given array/list itself. Hence, no need to return or print anything.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the array/list.
Second line contains 'N' single space separated integers(all 0s, 1s and 2s) representing the elements in the array/list.

Output Format :

For each test case, print the sorted array/list elements in a row separated by a single space.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^5
Time Limit: 1 sec

Sample Input 1:

```
1
7
0 1 2 0 2 0 1
```

Sample Output 1:

```
0 0 0 1 1 2 2
```

Sample Input 2:

```
2
5
2 2 0 1 1
7
0 1 2 0 1 2 0
```

Sample Output 2:

```
0 1 1 2 2

0 0 0 1 1 2 2
```

```
1.  void sort012(int *arr, int n)
2.  {
3.      //Write your code here
4.      int nz = 0;
5.      int nt = n-1;
6.      int i = 0;
7.
8.      while(i <= nt){
9.
10.         if(arr[i] == 0){
11.            int temp = arr[nz];
12.             arr[nz] = 0;
13.             arr[i] = temp;
14.             nz ++;
15.             i++;
16.            continue;
17.         }
18.
19.         if(arr[i] == 1)
20.         {
21.             i++;
22.             continue;
23.         }
24.
25.
26.         if (arr[i] == 2){
27.             int temp = arr[nt];
28.             arr[nt] = 2;
29.             arr[i] = temp;
30.             nt--;
31.            continue;
32.         }
33.     }
34. }
```

10-Ass : Sum of Two Arrays

Send Feedback

Two random integer arrays/lists have been given as ARR1 and ARR2 of size N and M respectively. Both the arrays/lists contain numbers from 0 to 9(i.e. single digit integer is present at every index). The idea here is to represent each array/list as an integer in itself of digits N and M.

You need to find the sum of both the input arrays/list treating them as two integers and put the result in another array/list i.e. output array/list will also contain only single digit at every index.

Note:

The sizes N and M can be different.

Output array/list(of all 0s) has been provided as a function argument. Its size will always be one more than the size of the bigger array/list. Place 0 at the 0th index if there is no carry.

No need to print the elements of the output array/list.
Using the function "sumOfTwoArrays", write the solution to the problem and store the answer inside this output array/list. The main code will handle the printing of the output on its own.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains an integer 'N' representing the size of the first array/list.
Second line contains 'N' single space separated integers representing the elements of the first array/list.
Third line contains an integer 'M' representing the size of the second array/list.
Fourth line contains 'M' single space separated integers representing the elements of the second array/list.

Output Format :

For each test case, print the required sum of the arrays/list in a row, separated by a single space.
Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^5
0 <= M <= 10^5
Time Limit: 1 sec

Sample Input 1:

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

Sample Output 1:

```
1 3 8 0
```

Sample Input 2:

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

Sample Output 2:

```
0 8 6 5

1 0 2 2 0
```

```
1.  void sumOfTwoArrays(int *input1, int size1, int *input2, int size2, int *output)
2.  {
3.      //Write your code here
4.      int i = size1-1, j = size2-1;
5.      int k = 0;
6.
7.      if(size1 < size2){
8.          k = size2;
9.      }
10.     else {
11.         k = size1;
12.     }
13.
14.
15.     while(i >= 0 && j >= 0){
16.
17.         int sum = input1[i] + input2[j] + output[k];
18.
19.         if(sum <= 9){
20.             output[k] = sum;
21.             k--;
22.         }
23.         else
24.         {
25.             int carry = sum / 10;
26.             output[k] = sum % 10;
27.             output[k-1] = carry;
28.             k--;
29.
30.         }
31.         i--;
32.         j--;
33.
34.
35.     }
36.
37.      while(i >= 0){
38.
39.         int sum = input1[i] + output[k];
40.
41.         if(sum < 9){
42.             output[k] = sum;
43.             k--;
44.         }
45.         else
46.         {
47.             int carry = sum / 10;
48.             output[k] = sum % 10;
49.             output[k-1] = carry;
```

```
            k--;

        }
        i--;
    }

     while(j >= 0){

        int sum = input2[i] + output[k];

        if(sum < 9){
            output[k] = sum;
            k--;
        }
        else
        {
            int carry = sum / 10;
            output[k] = sum % 10;
            output[k-1] = carry;
            k--;

        }
        j--;
    }

}
```

L10A : Test 2

Q1 : Print 2D Array

Send Feedback

Given a 2D integer array with n rows and m columns. Print the 0th row from input n times, 1st row n-1 times…..(n-1)th row will be printed 1 time.

Input format :

Line 1 : No of rows (n) and no of columns (m) (separated by single space)
Line 2 : Row 1 elements (separated by space)
Line 3 : Row 2 elements (separated by space)
Line 4 : and so on

Sample Input 1:

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

Sample Output 1 :

```
1 2 3
1 2 3
1 2 3
4 5 6
4 5 6

7 8 9
```

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

void print2DArray(int **input, int row, int col) {
        // Write your code here
    for(int i = 0; i < row; i ++){

        for(int k = 0; k< row-i; k++){
            for(int j = 0; j < col ; j++){
                cout << input[i][j] << " ";
            }
            cout << endl;
        }

    }
}
```

Q2 : Minimum Length Word

Send Feedback

Given a string S (that can contain multiple words), you need to find the word which has minimum length.

Note : If multiple words are of same length, then answer will be first minimum length word in the string.
Words are seperated by single space only.

Input Format :

String S

Output Format :

Minimum length word

Constraints :

1 <= Length of String S <= 10^5

Sample Input 1 :

this is test string

Sample Output 1 :

is

Sample Input 2 :

abc de ghihjk a uvw h j

Sample Output 2 :

a

```
/* input - Input String
*  output - Save the result in the output array (passed as argument). You donâ ™t have to
*  print or return the result
*/
#include<bits/stdc++.h>
#include<string>
#include<cstring>
void minLengthWord(char input[], char output[]){

        // Write your code here
    int min = 1e6;
    int si = 0,ei = 0,i = 0;
    int k = 0;
    int n = 0;
    while(input[i] != '\0'){
        n++;
        i++;
    }
    i = 0;


    while(i < n){

```

```
24.         if(input[i] == ' '){
25.             ei = i;
26.             int curlen = ei - si;
27.             if(curlen < min){
28.                 min = curlen;
29.                  k = 0;
30.                 while(si < ei){
31.                     output[k++] = input[si];
32.                     si++;
33.                 }
34.               if(input[si] == ' '){
35.                   si++;
36.               }
37.
38.             }else if(curlen >= min){
39.                  si = ei + 1;
40.             }
41.
42.
43.         }
44.
45.
46.         i++;
47.     }
48.
49.      ei = n;
50.     int minlast = ei-si;
51.     if(minlast < min){
52.         min = minlast;
53.         k = 0;
54.         while(si < ei){
55.                     output[k++] = input[si];
56.                     si++;
57.                 }
58.     }
59.
60.
61.     output[k] = '\0';
62.
63.
64. }
```

Q3 : Leaders in array

Send Feedback

Given an integer array A of size n. Find and print all the leaders present in the input array. An array element A[i] is called Leader, if all the elements following it (i.e. present at its right) are less than or equal to A[i].

Print all the leader elements separated by space and in the same order they are present in the input array.

Input Format :

Line 1 : Integer n, size of array
Line 2 : Array A elements (separated by space)

Output Format :

leaders of array (separated by space)

Constraints :

1 <= n <= 10^6

Sample Input 1 :

6
3 12 34 2 0 -1

Sample Output 1 :

34 2 0 -1

Sample Input 2 :

5
13 17 5 4 6

Sample Output 2 :

17 6

```
1.  void Leaders(int* arr,int len)
2.  {
3.              /* Don't write main().
4.               * Don't read input, it is passed as function argument.
5.               * Print your output exactly in the same format as shown.
6.               * Don't print any extra line.
7.              */
8.      for(int i = 0; i< len; i++){
9.                  int flag = 1;
10.                 int curr_leader = arr[i];
11.
12.                 for(int j = i; j < len; j++){
13.                     if(arr[j] > curr_leader){
14.                         flag = 0;
15.                         break;
16.                     }
17.                 }
18.                 if(flag == 1){
```

```
19.                     cout << curr_leader <<" ";
20.                 }
21.
22.
23.             }
24. }
25.
26.
```

L10 : Character Arrays and 2d Arrays Practice Questions

1-Tut : Check Palindrome

Send Feedback

Given a string, determine if it is a palindrome, considering only alphanumeric characters.

Palindrome

A palindrome is a word, number, phrase, or other sequences of characters which read the same backwards and forwards.

Example:

If the input string happens to be, "malayalam" then as we see that this word can be read the same as forward and backwards, it is said to be a valid palindrome.
The expected output for this example will print, 'true'.
From that being said, you are required to return a boolean value from the function that has been asked to implement.

Input Format:

The first and only line of input contains a string without any leading and trailing spaces. All the characters in the string would be in lower case.

Output Format:

The only line of output prints either 'true' or 'false'.

Note:

You are not required to print anything. It has already been taken care of.

Constraints:

0 <= N <= 10^6
Where N is the length of the input string.
Time Limit: 1 second

Sample Input 1 :

abcdcba

Sample Output 1 :

true

Sample Input 2:

coding

Sample Output 2:

false

```
1.  bool checkPalindrome(char str[]) {
2.      // Write your code here
3.      int len = 0;
4.      while(str[len] != '\0'){
5.          len++;
6.      }
7.      int i = 0, j = len-1;
8.      while(i < j){
```

```
9.
10.         if(str[i] == str[j]){
11.             i++;
12.             j--;
13.             continue;
14.         }
15.         else if(str[i] != str[j])
16.         {
17.             return 0;
18.         }
19.     }
20.     return 1;
21. }
```

2-Tut : Replace Character

Send Feedback

Given an input string S and two characters c1 and c2, you need to replace every occurrence of character c1 with character c2 in the given string.

Input Format :

Line 1 : Input String S
Line 2 : Character c1 and c2 (separated by space)

Output Format :

Updated string

Constraints :

1 <= Length of String S <= 10^6

Sample Input :

abacd
a x

Sample Output :

xbxcd

```
1.  void replaceCharacter(char input[], char c1, char c2) {
2.      // Write your code here
3.      int i = 0;
4.      while(input[i] != '\0'){
5.
6.          if(input[i] == c1){
7.              input[i] = c2;
8.          }
9.
10.         i++;
11.     }
12. }
```

3-Tut : Trim Spaces

Send Feedback

Given an input string S that contains multiple words, you need to remove all the spaces present in the input string.

There can be multiple spaces present after any word.

Input Format :

String S

Output Format :

Updated string

Constraints :

1 <= Length of string S <= 10^6

Sample Input :

abc def g hi

Sample Output :

abcdefghi

```
1.  void trimSpaces(char input[]) {
2.      // Write your code here
3.      int i = 0 , j = 0;
4.      while(input[i] != '\0'){
5.
6.          if(input[i] != ' '){
7.              input[j++] = input[i];
8.
9.          }
10.
11.         i++;
12.     }
13.
14.     input[j] = '\0';
15. }
```

4-Tut : Reverse Word Wise

Send Feedback

Reverse the given string word wise. That is, the last word in given string should come at 1st place, last second word at 2nd place and so on. Individual words should remain as it is.

Input format :

String in a single line

Output format :

Word wise reversed string in a single line

Constraints :

0 <= |S| <= 10^7

where |S| represents the length of string, S.

Sample Input 1:

Welcome to Coding Ninjas

Sample Output 1:

Ninjas Coding to Welcome

Sample Input 2:

Always indent your code

Sample Output 2:

code your indent Always

```
void reverseStringWordWise(char input[]) {
    // Write your code here
    int len = 0;
    int i = 0;
    while(input[i] != '\0'){
        len++;
        i++;
    }

    i = 0;
    int j = len - 1;

    // Reverse the string
    while(i < j){
        char temp = input[i];
        input[i] = input[j];
        input[j] = temp;
        i++;
        j--;
    }

    i = 0;
    j = 0;
    while(input[i] != '\0')
    {

                int k = 0;
                if(input[i] == ' ')
                {
                    k = i-1;
                    while(j < k){
                      char temp = input[j];
                       input[j] = input[k];
                       input[k] = temp;
                       j++;
                       k--;
                      }
```

```
                    j = i+1;

                            }

        i++;
    }

    i = len -1;

    while(j < i){

                     char temp = input[j];
                      input[j] = input[i];
                      input[i] = temp;
                      j++;
                      i--;

    }

}
```

5-Tut : Print All Substrings

Send Feedback

For a given input string(str), write a function to print all the possible substrings.

Substring

A substring is a contiguous sequence of characters within a string.
Example: "cod" is a substring of "coding". Whereas, "cdng" is not as the characters taken are not contiguous

Input Format:

The first and only line of input contains a string without any leading and trailing spaces. All the characters in the string would be in lower case.

Output Format:

Print the total number of substrings possible, where every substring is printed on a single line and hence the total number of output lines will be equal to the total number of substrings.

Note:

The order in which the substrings are printed, does not matter.

Constraints:

0 <= N <= 10^6
Where N is the length of the input string.
Time Limit: 1 second

Sample Input 1:

abc

Sample Output 1:

a
ab
abc

b
bc
c

Sample Input 2:

co

Sample Output 2:

c
co

o

```
1.  void printSubstrings(char input[]) {
2.      // Write your code here
3.
4.      for(int i = 0; input[i] != '\0'; i++)
5.
6.      {
7.
8.          for(int j = i; input[j] != '\0'; j++){
9.
10.             for(int k = i; k <= j; k++){
11.                 cout << input[k];
12.             }
13.             cout<< endl;
14.         }
15.           cout<< endl;
16.
17.     }
18. }
```

6-Tut : Column Wise Sum

Send Feedback

Given a 2D integer array of size M*N, find and print the sum of ith column elements separated by space.

Input Format :

First and only line of input contains M and N, followed by M * N space separated integers representing the elements in the 2D array.

Output Format :

Sum of every ith column elements (separated by space)

Constraints :

1 <= M, N <= 10^3

Sample Input :

4 2 1 2 3 4 5 6 7 8

Sample Output :

16 20

```
#include<iostream>
using namespace std;
int main(){

        /*  Read input as specified in the question.
             * Print output as specified in the question.
             */
   int m ,n;
     cin >> m >> n;
    int a[m][n];

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

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

    }

    for(int j = 0; j < n; j++){
      int sum = 0;
       for(int i = 0; i< m; i++){
           sum = sum + a[i][j];
       }
       cout << sum << " ";
    }
}
```

7-Tut : Largest Row or Column

Send Feedback

For a given two-dimensional integer array/list of size (N x M), you need to find out which row or column has the largest sum(sum of all the elements in a row/column) amongst all the rows and columns.

Note :

If there are more than one rows/columns with maximum sum, consider the row/column that comes first. And if ith row and jth column has the same largest sum, consider the ith row as answer.

Input Format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains two integer values, 'N' and 'M', separated by a single space. They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.
Second line onwards, the next 'N' lines or rows represent the ith row values.
Each of the ith row constitutes 'M' column values separated by a single space.

Output Format :

For each test case, If row sum is maximum, then print: "row" <row_index> <row_sum>
OR
If column sum is maximum, then print: "column" <col_index> <col_sum>

It will be printed in a single line separated by a single space between each piece of information.
Output for every test case will be printed in a seperate line.

Consider :

If there doesn't exist a sum at all then print "row 0 -2147483648", where -2147483648 or -2^31 is the smallest value for the range of Integer.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^3
0 <= M <= 10^3
Time Limit: 1sec

Sample Input 1 :

1
2 2
1 1
1 1

Sample Output 1 :

row 0 2

Sample Input 2 :

2
3 3
3 6 9
1 4 7
2 8 9
4 2
1 2
90 100
3 40
-10 200

Sample Output 2 :

column 2 25

column 1 342

```
#include <bits/stdc++.h>
void findLargest(int **input, int nRows, int mCols)
{
    //Write your code here
    if(nRows == 0 && mCols == 0){
        cout << "row" << " " << 0 <<" " << INT_MIN;
        return;
    }

    int row[nRows] = {0};
    int col[mCols] = {0};

    for(int i = 0; i < nRows; i++){
        int rsum = 0;

```

```
16.         for(int j = 0; j < mCols; j++){
17.             rsum = rsum + input[i][j];
18.         }
19.         row[i] = rsum;
20.
21.     }
22.
23.     for(int j = 0; j < mCols; j++){
24.         int colsum = 0;
25.
26.         for(int i = 0; i < nRows; i++){
27.             colsum = colsum + input[i][j];
28.         }
29.         col[j] = colsum;
30.
31.     }
32.     int rmax = row[0], rindex = 0,rcount = 0;
33.     int colmax = col[0], colindex = 0,colcount = 0;
34.
35.     for(int i = 0; i < nRows; i++){
36.
37.         if(row[i] > rmax ){
38.             rmax = row[i];
39.             rindex = i;
40.         }
41.     }
42.
43.      for(int i = 0; i < mCols; i++){
44.
45.         if(col[i] > colmax ){
46.             colmax = col[i];
47.             colindex = i;
48.         }
49.     }
50.
51.     if(rmax > colmax){
52.         cout << "row" << " " << rindex << " " << rmax;
53.     }
54.     if(rmax < colmax){
55.         cout << "column" << " " << colindex << " " << colmax;
56.     }
57.
58.     if(rmax == colmax){
59.         cout << "row" << " " << rindex << " " << rmax;
60.     }
61.
62.
63. }
```

8-Tut : Wave Print

Send Feedback

For a given two-dimensional integer array/list of size (N x M), print the array/list in a sine wave order, i.e, print the first column top to bottom, next column bottom to top and so on.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.
First line of each test case or query contains two integer values, 'N' and 'M', separated by a single space. They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.
Second line onwards, the next 'N' lines or rows represent the ith row values.
Each of the ith row constitutes 'M' column values separated by a single space.

Output format :

For each test case, print the elements of the two-dimensional array/list in the sine wave order in a single line, separated by a single space.
Output for every test case will be printed in a seperate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^3
0 <= M <= 10^3
Time Limit: 1sec

Sample Input 1:

```
1
3 4
1  2  3  4
5  6  7  8
9 10 11 12
```

Sample Output 1:

```
1 5 9 10 6 2 3 7 11 12 8 4
```

Sample Input 2:

```
2
5 3
1 2 3
4 5 6
7 8 9
10 11 12
13 14 15
3 3
10 20 30
40 50 60
70 80 90
```

Sample Output 2:

```
1 4 7 10 13 14 11 8 5 2 3 6 9 12 15

10 40 70 80 50 20 30 60 90
```

```
1.  void wavePrint(int **input, int nRows, int mCols)
2.  {
3.      //Write your code here
4.      for(int j = 0; j < mCols; j++){
5.          //int k = j;
6.
7.                      if(j % 2 == 0){
8.
9.                         for(int i = 0; i < nRows ; i++){
10.                           cout << input[i][j] << " ";
11.                        }
12.                     }
13.                     else {
14.                           for(int i = nRows-1; i >= 0 ; i--){
15.                              cout << input[i][j] << " ";
16.                           }
17.
18.                     }
19.     }
20. }
```

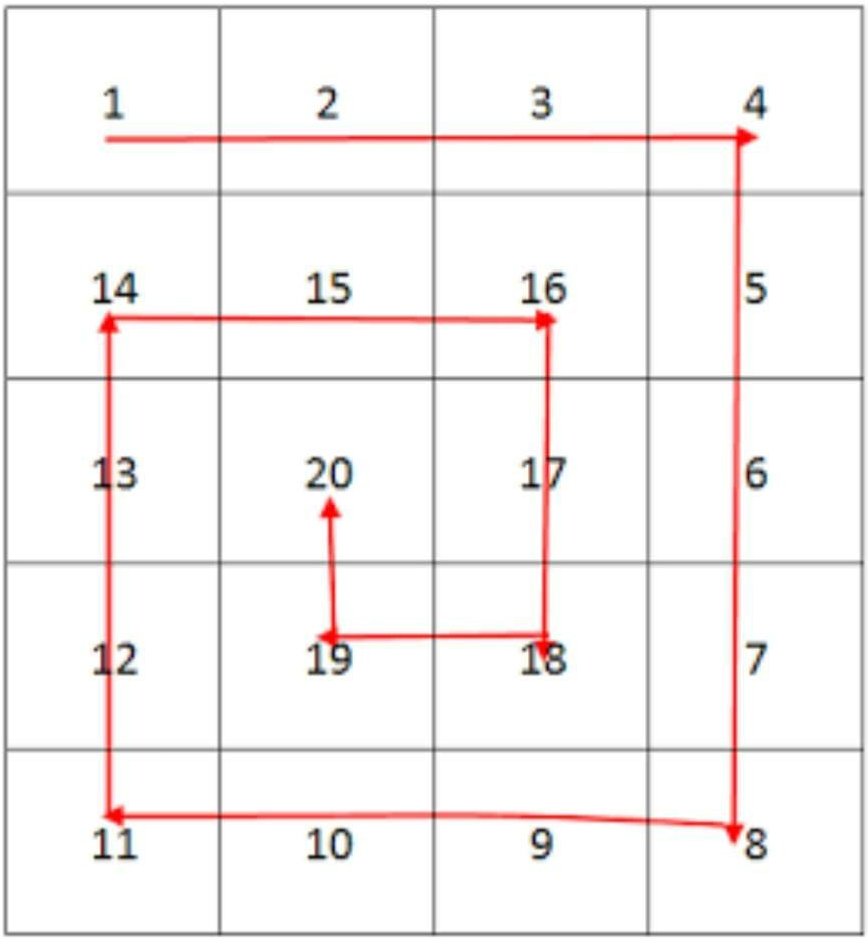
9-Tut : Spiral Print

Send Feedback

For a given two-dimensional integer array/list of size (N x M), print it in a spiral form. That is, you need to print in the order followed for every iteration:

a. First row(left to right)
b. Last column(top to bottom)
c. Last row(right to left)
d. First column(bottom to top)
Mind that every element will be printed only once.

Refer to the Image:



Output :	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains two integer values, 'N' and 'M', separated by a single space. They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.

Second line onwards, the next 'N' lines or rows represent the ith row values.

Each of the ith row constitutes 'M' column values separated by a single space.

Output format :

For each test case, print the elements of the two-dimensional array/list in the spiral form in a single line, separated by a single space.

Output for every test case will be printed in a seperate line.

Constraints :

1 <= t <= 10^2
0 <= N <= 10^3
0 <= M <= 10^3
Time Limit: 1sec

Sample Input 1:

```
1
4 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

Sample Output 1:

```
1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10
```

Sample Input 2:

```
2
3 3
1 2 3
4 5 6
7 8 9
3 1
10
20
30
```

Sample Output 2:

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

10 20 30
```

```
1.  void spiralPrint(int **input, int nRows, int nCols)
2.  {
3.      //Write your code here
4.
5.      int rs = 0, re = nRows-1;
6.      int cs = 0, ce = nCols -1;
7.      int count = 0,total = nRows * nCols;
8.      while(count < total){
9.
10.         // 1.
11.         for(int i = cs; i <= ce; i++){
12.             if(count == total){
13.                 break;
14.             }
15.
16.             cout << input[rs][i] << " ";
17.             count++;
18.
19.         }
```

```
        rs ++;

        // 2.
        for(int i = rs; i<= re; i++){
            if(count == total){
                break;
            }
            cout << input[i][ce] << " ";
            count++;

        }
        ce--;
        // 3

        for(int i = ce; i >= cs; i--){
           if(count == total){
                break;
            }

            cout << input[re][i] << " ";
            count++;
        }


        re--;

        // 4


        for(int i = re; i >= rs; i--){

            if(count == total){
                break;
            }
            cout << input[i][cs]<<" ";
            count ++;
        }

        cs ++;


    }

}
```

10-Ass : **Check Permutation**

Send Feedback

For a given two strings, 'str1' and 'str2', check whether they are a permutation of each other or not.

Permutations of each other

Two strings are said to be a permutation of each other when either of the string's characters can be rearranged so that it becomes identical to the other one.

Example:

str1= "sinrtg"

str2 = "string"

The character of the first string(str1) can be rearranged to form str2 and hence we can say that the given strings are a permutation of each other.

Input Format:

The first line of input contains a string without any leading and trailing spaces, representing the first string 'str1'.

The second line of input contains a string without any leading and trailing spaces, representing the second string 'str2'.

Note:

All the characters in the input strings would be in lower case.

Output Format:

The only line of output prints either 'true' or 'false', denoting whether the two strings are a permutation of each other or not.

You are not required to print anything. It has already been taken care of. Just implement the function.

Constraints:

0 <= N <= 10^6

Where N is the length of the input string.

Time Limit: 1 second

Sample Input 1:

abcde

baedc

Sample Output 1:

true

Sample Input 2:

abc

cbd

Sample Output 2:

false

```
1.  bool isPermutation(char input1[], char input2[]) {
2.      // Write your code here
3.
4.      int freq[256] = {0,0};
5.
6.      if(strlen(input1) != strlen(input2) ){
7.          return 0;
8.      }
9.
10.     int i = 0;
11.     while(input1[i] != '\0'){
12.         int c = input1[i];
13.         freq[c] = freq[c] + 1;
```

```
        i++;
    }

    i = 0;

    while(input2[i] != '\0'){
        int c = input2[i];
        freq[c] = freq[c] - 1;
        if(freq[c] < 0){

            return 0;
        }

        i++;
    }

     return 1;
}
```

11-Ass : Remove Consecutive Duplicates

Send Feedback

For a given string(str), remove all the consecutive duplicate characters.

Example:

Input String: "aaaa"
Expected Output: "a"
Input String: "aabbbcc"
Expected Output: "abc"

Input Format:

The first and only line of input contains a string without any leading and trailing spaces. All the characters in the string would be in lower case.

Output Format:

The only line of output prints the updated string.

Note:

You are not required to print anything. It has already been taken care of.

Constraints:

0 <= N <= 10^6
Where N is the length of the input string.
Time Limit: 1 second

Sample Input 1:

aabccbaa

Sample Output 1:

abcba

Sample Input 2:

xxyyzxx

Sample Output 2:

xyzx

```
1.  void removeConsecutiveDuplicates(char input[]) {
2.      // Write your code here
3.      char lastchar = input[0];
4.      int i = 1, j = 1;
5.
6.      while(input[i] != '\0'){
7.
8.          if(input[i] == lastchar){
9.              i++;
10.             continue;
11.         }
12.         else{
13.         input[j] = input[i];
14.         lastchar = input[i];
15.         j++;
16.         i++;
17.         }
18.
19.     }
20.     input[j] = '\0';
21. }
```

12-Ass : Reverse Each Word

Send Feedback

Aadil has been provided with a sentence in the form of a string as a function parameter. The task is to implement a function so as to print the sentence such that each word in the sentence is reversed.

Example:

Input Sentence: "Hello, I am Aadil!"
The expected output will print, ",olleH I ma !lidaA".

Input Format:

The first and only line of input contains a string without any leading and trailing spaces. The input string represents the sentence given to Aadil.

Output Format:

The only line of output prints the sentence(string) such that each word in the sentence is reversed.

Constraints:

0 <= N <= 10^6
Where N is the length of the input string.
Time Limit: 1 second

Sample Input 1:

Welcome to Coding Ninjas

Sample Output 1:

emocleW ot gnidoC sajniN

Sample Input 2:

Always indent your code

Sample Output 2:

```
syawlA tnedni ruoy edoc
```

```
1.  void reverseEachWord(char input[]) {
2.      // Write your code here
3.      int s = 0;
4.      int i = 0;
5.
6.      while(input[i] != '\0'){
7.
8.                  if(input[i] == ' '){
9.                      int e = i-1;
10.                             while(s < e){
11.                                 char temp = input[s];
12.                                 input[s] = input[e];
13.                                 input[e] = temp;
14.                                 s++;
15.                                 e--;
16.                             }
17.                     s = i+1;
18.
19.                 }
20.
21.         if(input[i+1] == '\0'){
22.                         int e = i;
23.                             while(s < e){
24.                                 char temp = input[s];
25.                                 input[s] = input[e];
26.                                 input[e] = temp;
27.                                 s++;
28.                                 e--;
29.                              }
30.
31.
32.         }
33.
34.
35.
36.         i++;
37.     }
38.
39. }
```

13-Ass : Remove character

Send Feedback

For a given a string(str) and a character X, write a function to remove all the occurrences of X from the given string.

The input string will remain unchanged if the given character(X) doesn't exist in the input string.

Input Format:

The first line of input contains a string without any leading and trailing spaces.
The second line of input contains a character(X) without any leading and trailing spaces.

Output Format:

The only line of output prints the updated string.

Note:

You are not required to print anything explicitly. It has already been taken care of.

Constraints:

0 <= N <= 10^6
Where N is the length of the input string.
Time Limit: 1 second

Sample Input 1:

aabccbaa
a

Sample Output 1:

bccb

Sample Input 2:

xxyyzxx
y

Sample Output 2:

xxzxx

```
1.  void removeAllOccurrencesOfChar(char input[], char c) {
2.      // Write your code here
3.      int i = 0;
4.
5.      while(input[i] != '\0'){
6.
7.          if(input[i] == c){
8.              int k = i;
9.
10.             while( input[k] != '\0'){
11.                 input[k] = input[k+1];
12.                 k++;
13.             }
14.             continue;
15.         }
16.
17.
18.         i++;
19.     }
20. }
```

14-Ass : Highest Occuring Character

Send Feedback

For a given a string(str), find and return the highest occurring character.

Example:

Input String: "abcdeapapqarr"
Expected Output: 'a'
Since 'a' has appeared four times in the string which happens to be the highest frequency character, the answer would be 'a'.
If there are two characters in the input string with the same frequency, return the character which comes first.

Consider:

Assume all the characters in the given string to be in lowercase always.

Input Format:

The first and only line of input contains a string without any leading and trailing spaces.

Output Format:

The only line of output prints the updated string.

Note:

You are not required to print anything explicitly. It has already been taken care of.

Constraints:

0 <= N <= 10^6
Where N is the length of the input string.
Time Limit: 1 second

Sample Input 1:

abdefgbabfba

Sample Output 1:

b

Sample Input 2:

xy

Sample Output 2:

x

```
char highestOccurringChar(char input[]) {
    // Write your code here
    int freq[256] = {0,0};

    int i = 0;
    while( input[i] != '\0'){
        int c = input[i];
        freq[c] = freq[c] + 1;
        i++;
    }

    int index = 97;
    int max = freq[97];
```

```
14.
15.     for(int i = 97; i <= 122 ; i++){
16.
17.         if(freq[i] > max){
18.             max = freq[i];
19.                 index = i;
20.         }
21.     }
22.
23.     char c = index;
24.     return c;
25. }
```

15-Ass : Compress the String

Send Feedback

Write a program to do basic string compression. For a character which is consecutively repeated more than once, replace consecutive duplicate occurrences with the count of repetitions.

Example:

If a string has 'x' repeated 5 times, replace this "xxxxx" with "x5".
The string is compressed only when the repeated character count is more than 1.

Note:

Consecutive count of every character in the input string is less than or equal to 9.

Input Format:

The first and only line of input contains a string without any leading and trailing spaces.

Output Format:

The output contains the string after compression printed in single line.

Note:

You are not required to print anything. It has already been taken care of. Just implement the given function.

Constraints:

0 <= N <= 10^6
Where 'N' is the length of the input string.
Time Limit: 1 sec

Sample Input 1:

aaabbccdsa

Sample Output 1:

a3b2c2dsa

Explanation for Sample Output 1:

In the given string 'a' is repeated 3 times, 'b' is repeated 2 times, 'c' is repeated 2 times and 'd', 's' and 'a' and occuring 1 time hence no compression for last 3 characters.

Sample Input 2:

aaabbcddeeeee

Sample Output 2:

a3b2cd2e5

Explanation for Sample Output 2:

In the given string 'a' is repeated 3 times, 'b' is repeated 2 times, 'c' is occuring single time, 'd' is repeating 2 times and 'e' is repeating 25times.

```
#include <bits/stdc++.h>
string getCompressedString(string &input) {
    // Write your code here.
    string newstr = "";
    int n = input.length();

    for (int i = 0; i < n; i++){
        int count = 1;
        char c = input[i];

        while (i < n - 1 && input[i] == input[i + 1]) {
            count++;
            i++;
        }

        if(count > 1){
            newstr +=  c + to_string(count);
        }

        if(count == 1){
            newstr += c;
        }


    }

    return newstr;
}
```