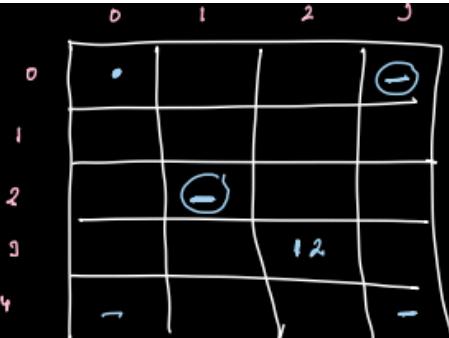


## 1. 2D Arrays - Collection of 1D Arrays (rows- N, cols - M)

2D Array	
↳ Collection of 1D Arrays	
↳ Rows (N)	
↳ Cols (M)	
Top Left → (0,0)	
Top Right → (0,3)	
Bottom Left → (4,0)	
Bottom Right → (4,3)	
	
(2, 1)	
(0, 3)	
(Rows Cols)	
arr[3][2] = 12;	

## 2. How to take input in 2D array

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

N  
A<sub>0</sub> A<sub>1</sub> A<sub>2</sub> ... A<sub>N-1</sub>

```
int arr[N][M];
for( i=0; i< N; i++)
{
    for( j=0; j< M; j++)
    {
        cin >> arr[i][j];
    }
}
```

N M  
A<sub>0,0</sub> A<sub>0,1</sub> A<sub>0,2</sub> ... A<sub>0,M-1</sub>  
A<sub>1,0</sub> A<sub>1,1</sub> A<sub>1,2</sub> ... A<sub>1,M-1</sub>

⋮

A<sub>N-1,0</sub> A<sub>N-1,1</sub> A<sub>N-1,2</sub> ... A<sub>N-1,M-1</sub>

int arr[10];



int arr[5][2].  
↑  
#rows  
↑  
#cols

arr[2][2] = 10;

arr[3][1] = 15;

cout << arr[2][1] << endl;

cin >> arr[2][1];

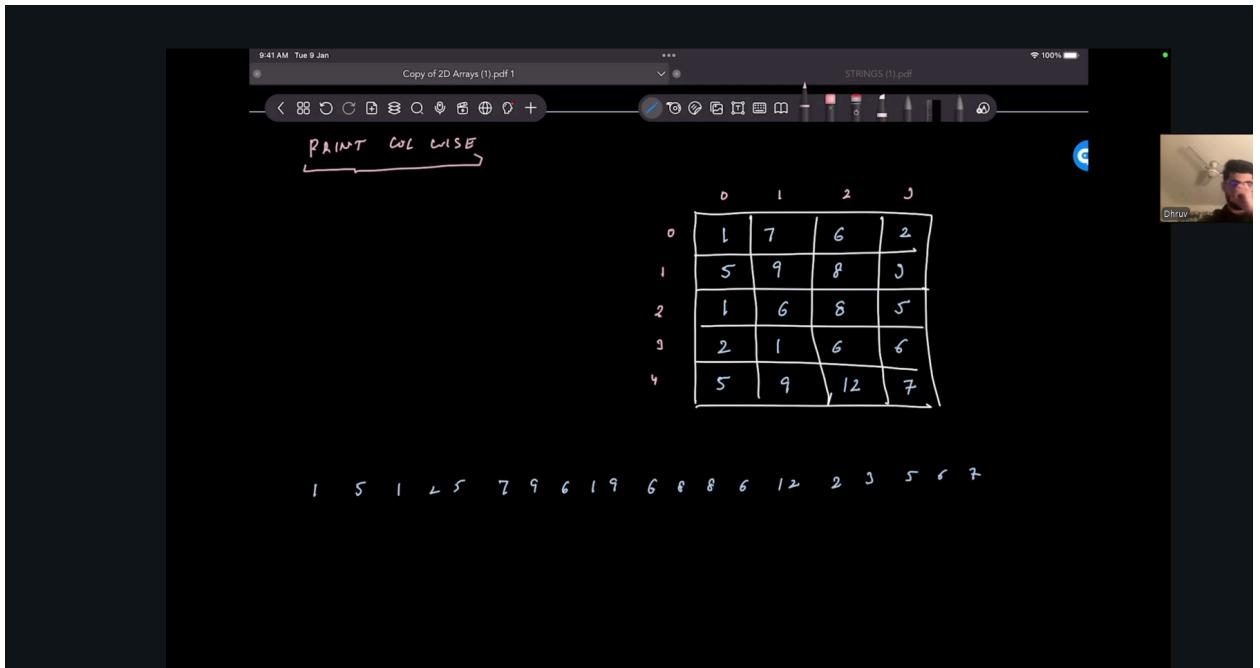
	0	1	2	3
0	?	?	?	?
1	?	?	?	?
2	?	?	10	?
3	?	15	?	?
4	?	?	?	?

5

### 3. Print 2D Array

```
PRINT 2D ARRAY (Row wise)
{
    for( i=0; i< N; i++)
    {
        for( j=0; j< M; j++)
        {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
}
```

#### 4. Print column-wise 2D array



1 5 1 2 5 7 9 6 1 9 6 8 8 6 12 2 3 5 6 7

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS STANDINGS ADM. EDIT CUSTOM INVOCATION

Source:

```
6 | int n, m;
7 | cin >> n >> m;
8 |
9 | int arr[n][m];
10 |
11 | for(int i = 0; i < n; i++)
12 | {
13 |     for(int j = 0; j < m; j++)
14 |     {
15 |         cin >> arr[i][j];
16 |     }
17 |
18 |     for(int i = 0; i < n; i++)
19 |     {
20 |         for(int j = 0; j < m; j++)
21 |         {
22 |             cout << arr[i][j] << " ";
23 |         }
24 |         cout << endl;
25 |
26 |     }
27 }
```

Language: GNU G++23 14.2 (64 bit, msys2)

Input:

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

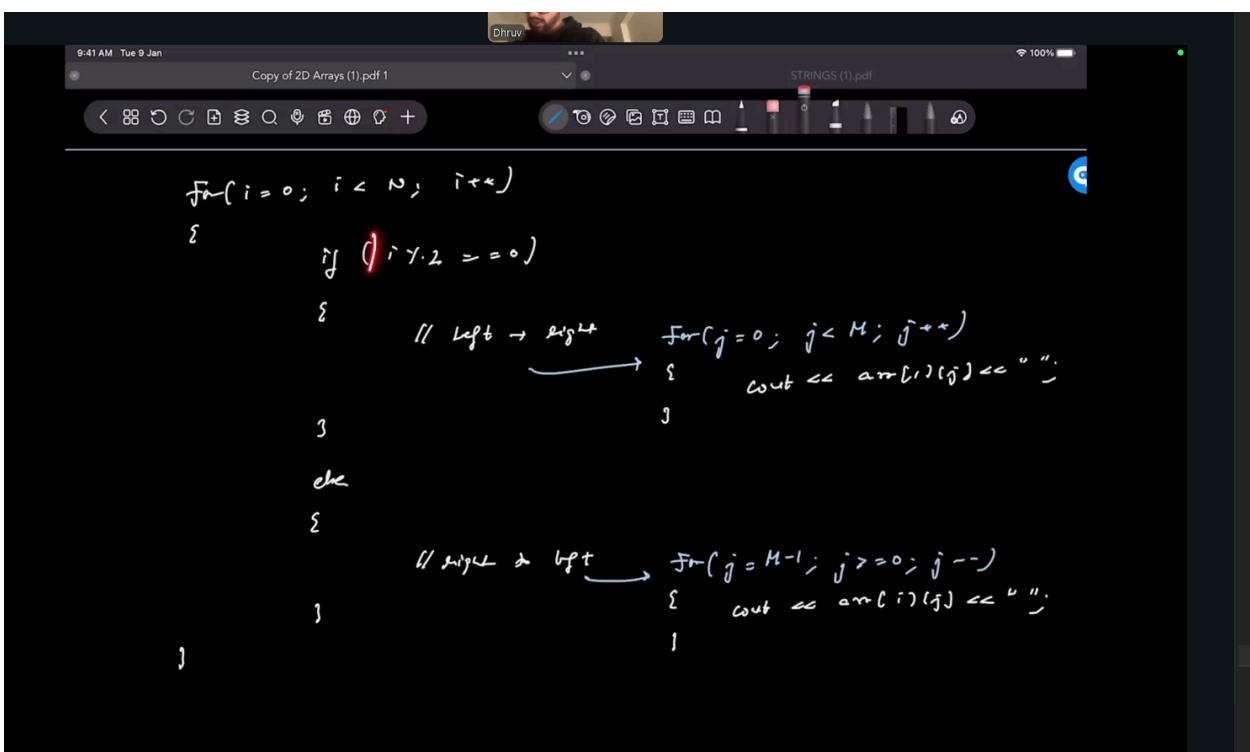
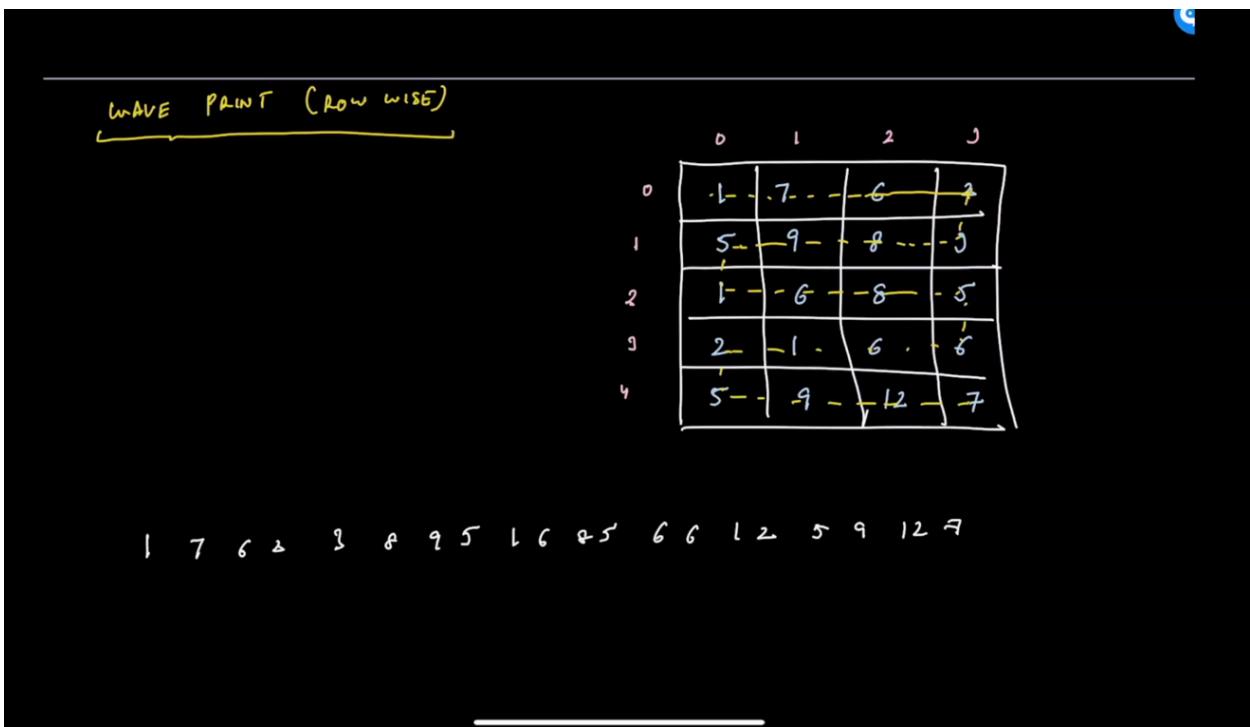
Output:

```
1 2 3 1
4 5 6 2
7 8 9 3
=====
Used: 15 ms, 0 KB
```

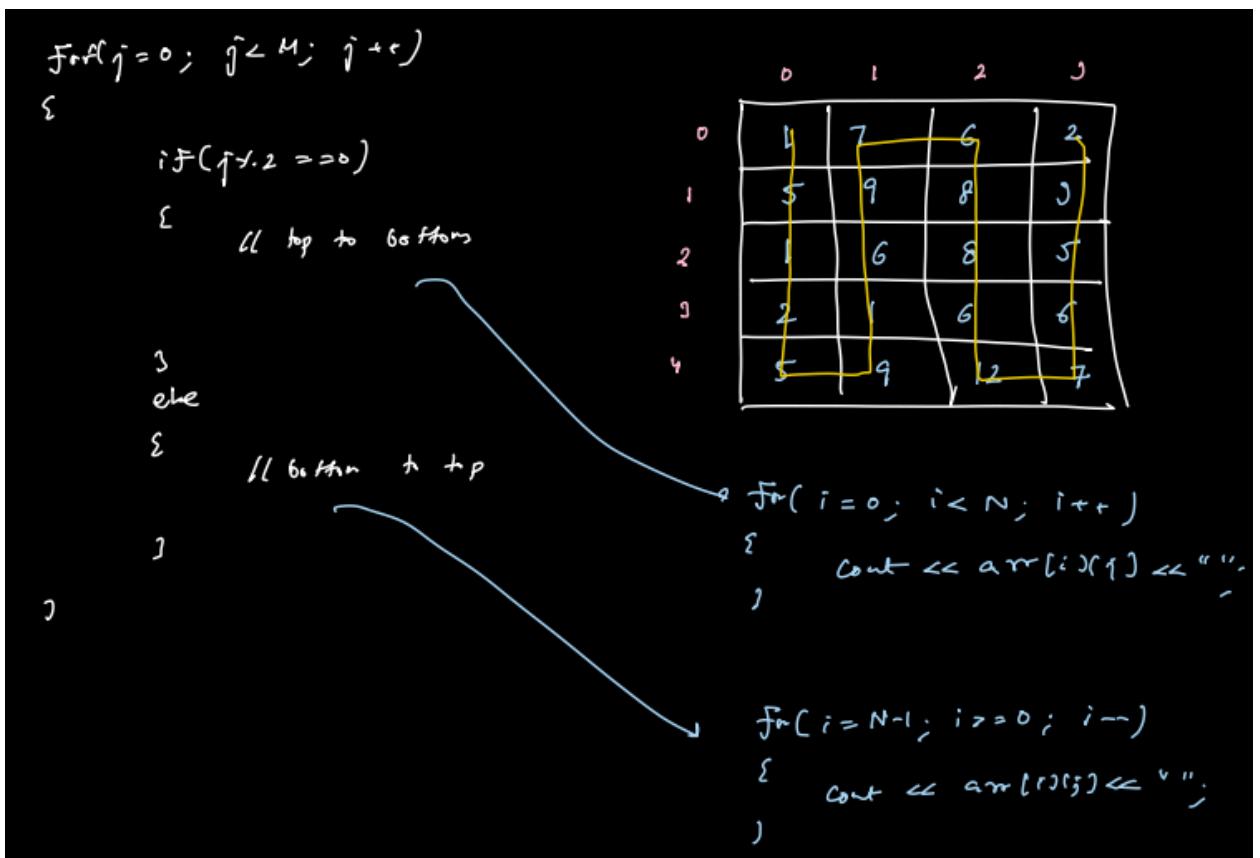
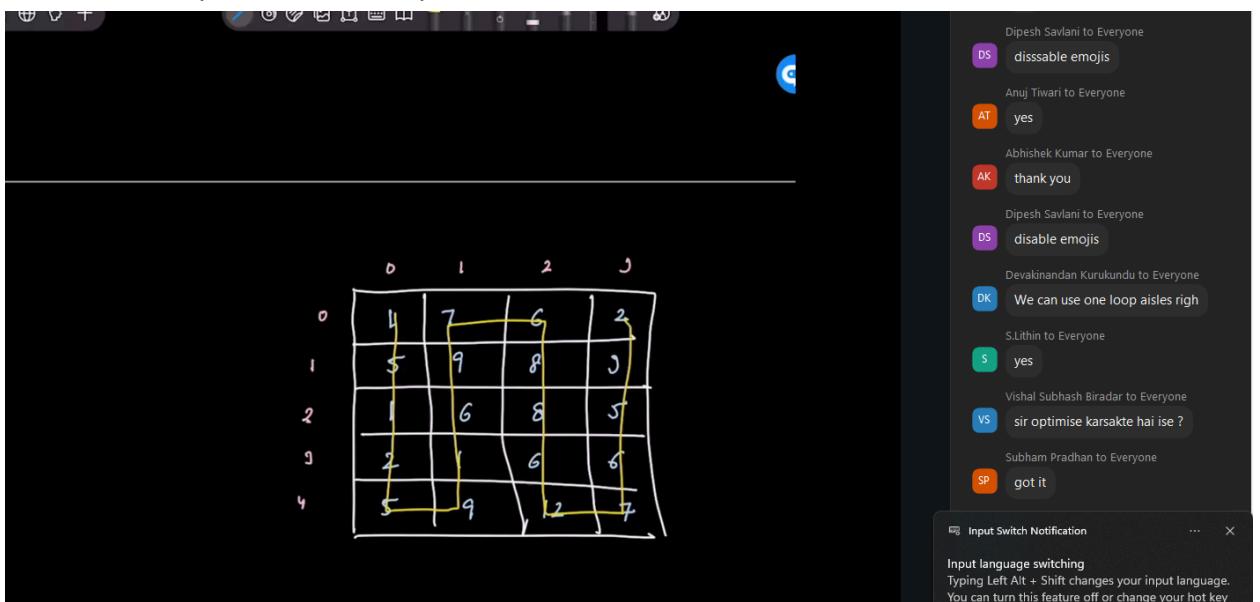
First 255 bytes only

Run

#### 4. Wave Print (Row wise)



## 5. Wave Print (Column wise)



## 6. Print Boundary of 2D Array

2) Col wise

3) Row (Row)

4) Row (Col)

5) Boundary

	0	1	2	3
0	1	7	6	2
1	5	9	8	3
2	1	6	8	5
3	2	1	6	6
4	5	9	12	7

$N \times M$

// First Row ( $row = 0$ )

```
for( j=0; j < M; j++)
```

```
{ cout << arr[0][j] << " ";
```

```
}
```

x x x x

x x x

x x x

y x x x

// Last Col ( $Col = M-1$ )

```
for( i=1; i < N; i++)
```

```
{ cout << arr[i][M-1] << " ";
```

```
}
```

// Last Row ( $Row = N-1$ )

```
for( j=M-2; j >= 0; j--)
```

```
{ cout << arr[N-1][j] << " ";
```

```
}
```

// First Col ( $Col = 0$ )

```
for( i=N-2; i >= 1; i--)
```

```
{ cout << arr[i][0] << " ";
```

```
}
```

## 7. Find Maximum in 2d array

Submit solution  
2D Arrays

Problem: C - Find Maximum  
standard input/output 1 s, 256 MB

Language: GNU G++23 14.2 (64 bit, msys2)

Source code:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int n, m;
7     cin >> n >> m;
8
9     int arr[n][m];
10
11    for(int i = 0; i < n; i++)
12    {
13        for(int j = 0; j < m; j++)
14        {
15            cin >> arr[i][j];
16        }
17    }
18
19    int ans = arr[0][0];
20
21    for(int i = 0; i < n; i++)
22    {
23        for(int j = 0; j < m; j++)
24        {
25            if(arr[i][j] > ans)
26            {
27                ans = arr[i][j];
28            }
29        }
30    }
31
32    cout << ans;
33 }
```

100xSchool Bootcamp 1.0  
Progress  
Sc Prob → A harkirat singh  
100X school

8.

0	0	1	1
1	1	0	0
2	1	0	1

2  
1  
2

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

2  
1  
2  
3  
1  
2

```

int bestRow = -1;
int maxCount = 0;

for( i=0; i< N; i++ )
{
    int count = 0;
    for( j=0; j< N; j++ )
    {
        if( arr[i][j] == 1 )
        {
            count++;
        }
    }
}

```

↗  
 $\boxed{arr = 3}$

$\boxed{\cancel{1}}$	$\boxed{\cancel{0}}$	$\boxed{3}$
$\cancel{0}$	$\cancel{1}$	$\cancel{0}$
$\cancel{1}$	$\cancel{0}$	$\cancel{1}$

```

if( count > maxCount )
{
    maxCount = count;
    bestRow = i;
}

cout << bestRow << endl;

```

# Strings

1.

**CHECK SAME**

*BINARY NUMBER*

*int x=20;  
10100*

*Given two characters in the input, check whether they are same or not.*

**a b**      **a a**

**NO**      **YES**

*char ch1, ch2;*

*cin >> ch1 >> ch2;*

*if (ch1 == ch2)*

*{ cout << "YES";*

*}*

*else*

*{ cout << "No";*

*}*

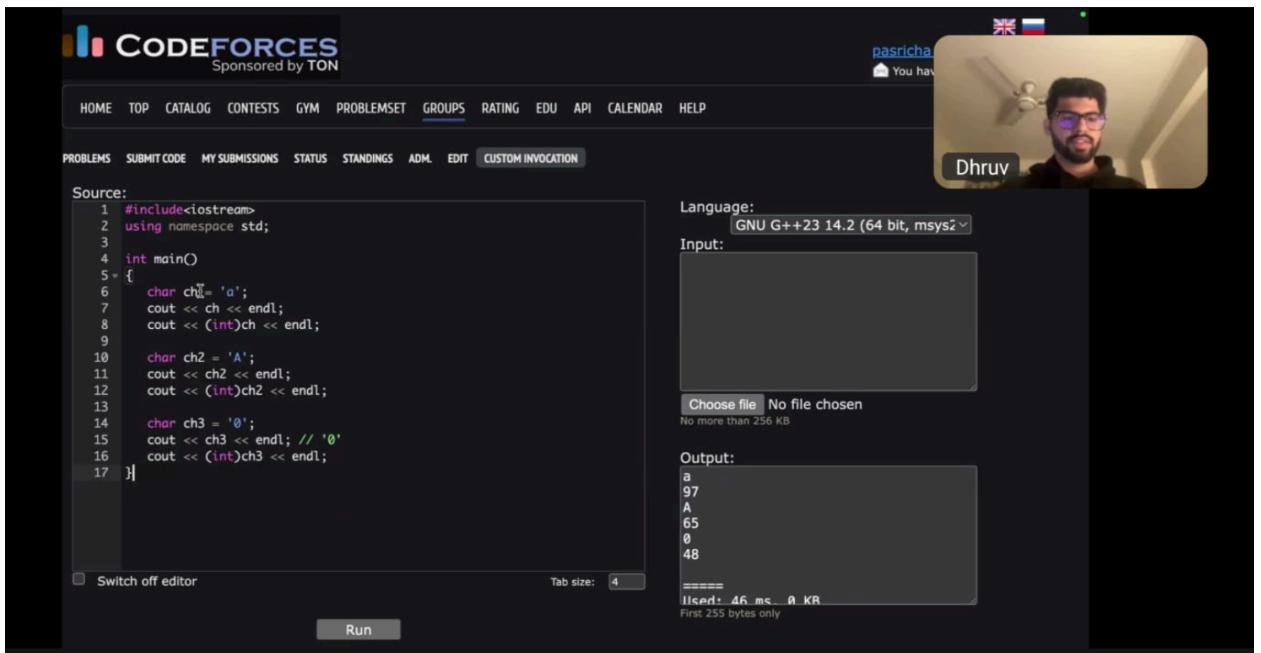
## 2. ASCII value

screen

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	'
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	\$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	:	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	-
63	3F	?	95	5F	-	127	7F	[DEL]

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	'
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	\$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	:	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	-
63	3F	?	95	5F	-	127	7F	[DEL]

### 3. Character ka ASCII value output (char to integer conversion)



Source:

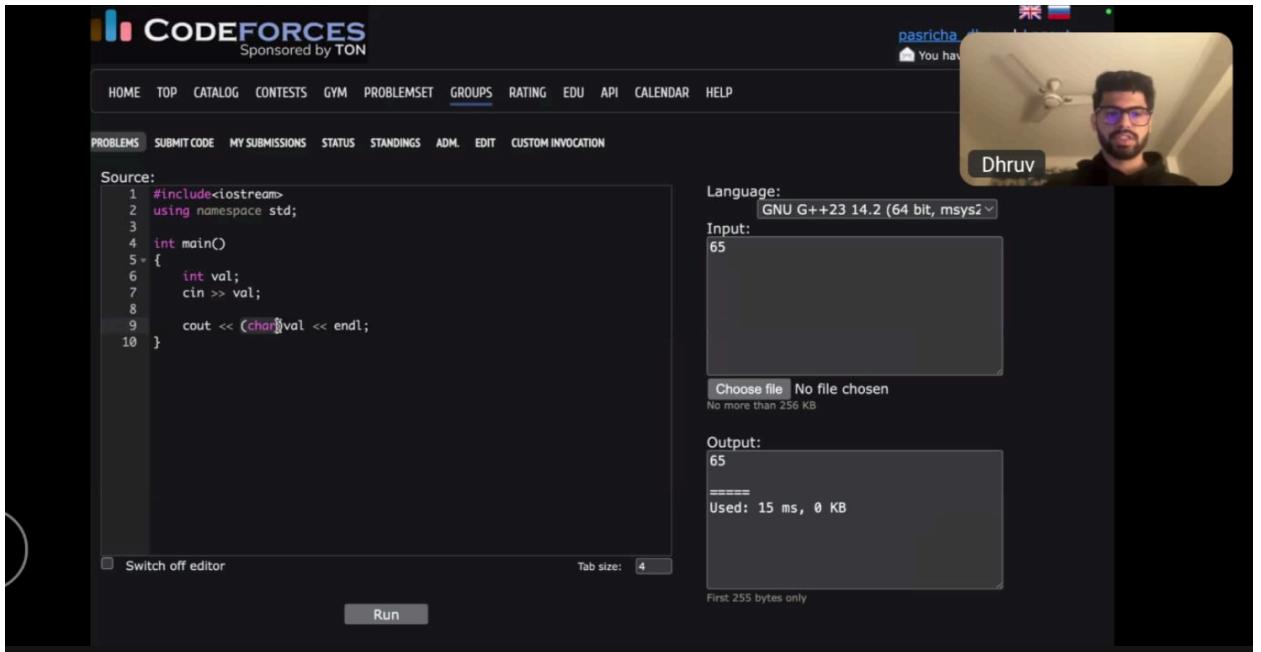
```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     char ch1= 'a';
7     cout << ch1 << endl;
8     cout << (int)ch1 << endl;
9
10    char ch2 = 'A';
11    cout << ch2 << endl;
12    cout << (int)ch2 << endl;
13
14    char ch3 = '0';
15    cout << ch3 << endl; // '0'
16    cout << (int)ch3 << endl;
17 }
```

Language: GNU G++23 14.2 (64 bit, msys2)

Input: Choose file No file chosen  
No more than 256 KB

Output:  
a  
97  
A  
65  
0  
48  
=====  
Used: 46 ms, 0 KB  
First 255 bytes only

### 4. Integer to character conversion



Source:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int val;
7     cin >> val;
8
9     cout << (char)val << endl;
10 }
```

Language: GNU G++23 14.2 (64 bit, msys2)

Input: Choose file No file chosen  
No more than 256 KB

Output:  
65  
=====  
Used: 15 ms, 0 KB  
First 255 bytes only

## 5. Check character is lowercase or not

The screenshot shows a programming environment with a dark theme. In the top navigation bar, the 'PROBLEMS' tab is selected. Below it, there are tabs for 'SUBMIT CODE', 'MY SUBMISSIONS', 'STATUS', 'STANDINGS', 'ADM.', 'EDIT', and 'CUSTOM INVOCATION'. On the right side of the interface, there is a video feed of a person with glasses and a beard, identified as 'Dhruv'. The main area is titled 'Source:' and contains the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     char ch;
7     cin >> ch;
8
9     if(ch >= 'a' and ch <= 'z')
10    {
11        cout << "Lowercase";
12    }
13    else
14    {
15        cout << "Not a lowercase";
16    }
17 }
```

Below the code editor, there are buttons for 'Switch off editor' and 'Run'. A 'Tab size:' dropdown is set to 4. To the right, the 'Language:' dropdown is set to 'GNU G++23'. The 'Input:' field contains the character 'j'. Under 'Output:', the result is displayed as 'Lowercase'. At the bottom, it says 'Used: 15 ms, 4 KB' and 'First 255 bytes only'.

## 6.

The screenshot shows a challenge titled 'ASCII TO CHARACTER' in large blue letters. Below the title, a subtitle reads 'Given an ASCII Value, print the corresponding character'. There are two blue rectangular boxes, each containing a white number: '65' and '97'. To the right of these boxes is a sample C++ code snippet:

```
int val;
cin >> val;
cout << (char) val;
```

At the bottom, there are two orange rectangular boxes, each containing a white letter: 'A' and 'a'.

7.

## CONVERT TO LOWERCASE

Given an alphabet (lowercase/uppercase), convert into lowercase.

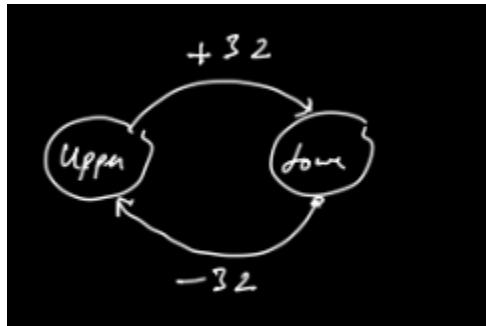


```
char ch;  
cin >> ch;
```

```
if( ch >= 'A' and ch <= 'Z')
```

```
{  
    ch += 32;  
}
```

```
cout << ch;
```



8.

## CHECK CHARACTER

Check whether a given character is lowercase, uppercase , digit or a special character.

a

B

7

@

lower

upper

digit

special

```
char ch;
cin >> ch;

if( ch >= 'a' and ch <= 'z')
{
    cout << "lower ";
}
else if( ch >= 'A' and ch <= 'Z')
{
    cout << "upper ";
}
else if(ch >= '0' and ch <= '9')
{
    cout << "digit ";
}
else
{
    cout << "special ";
}

(int) ch;
```

9.

## STRING

- **Collection / Sequence of characters** stored at continuous memory locations.

- In C++, there are 2 ways to implement it:

### 1) Char Array (C style)

- `char s[100];`
- Ends with '\0'
- **Manual memory handling**
- **More error-prone**

### 2) String (STL)

- `string s;`
- **Easier**
- **Automatic memory management**
- **Less error-prone**
- **More inbuilt functions available**

10. Size of string `.size()`

The screenshot shows a programming contest interface with the following details:

- Source:** A code editor containing the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     string s; // ""
7     cin >> s;
8
9     int n = s.size();
10
11    for(int i = 0; i < n; i++)
12    {
13        cout << s[i] << ' ';
14    }
15
16 }
```
- Language:** GNU G++2
- Input:** dhruv
- Output:** d h r u v  
=====
- Used:** 31 ms, 0 KB
- Notes:** First 255 bytes only

## 11. String input and output

The screenshot shows a competitive programming interface. In the top right corner, there is a video feed of a person with glasses and a beard, identified as "Dhruv". The main area contains a code editor with the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     string s; // ""
7     cin >> s; // "dhruv"
8
9     cout << s; // dhruv
10 }
```

Below the code editor are two buttons: "Switch off editor" and "Run". To the right of the code editor, the "Tab size:" dropdown is set to 4. On the far right, there is a vertical sidebar with a thumbs-up icon.

On the right side of the interface, there are several sections:

- Language:** GNU G++23 1.4.2 (64 bit, msys2)
- Input:** dhruv
- Output:** dhruv  
=====
- Used:** 15 ms, 0 KB

At the bottom of the right sidebar, it says "First 255 bytes only".

## 12. getline(string)

The screenshot shows the Codeforces platform. In the top right corner, there is a video feed of a person with glasses and a beard, identified as "Dhruv". The main area contains a code editor with the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     string s; // ""
7     getline(cin, s);
8     cout << s;
9 }
10
```

Below the code editor are two buttons: "Switch off editor" and "Run". To the right of the code editor, the "Tab size:" dropdown is set to 4. On the far right, there is a vertical sidebar with a thumbs-up icon.

On the right side of the interface, there are several sections:

- Language:** GNU G++23 14.2 (64 bit, msys2)
- Input:** hello world hello hello
- Output:** hello world hello hello  
=====
- Used:** 15 ms, 0 KB

At the bottom of the right sidebar, it says "First 255 bytes only".

13.

 **CODEFORCES**  
Sponsored by TON

---

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

---

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS STANDINGS CUSTOM INVOCATION

**E. Brothers**

time limit per test: 1 second  
memory limit per test: 256 megabytes

You are given the names of two people.

Each name is given as two words: `FirstName LastName`.

Determine whether they are brothers.

They are considered brothers if and only if their last names are the same.

**Input**  
The input consists of two lines.  
Each line contains two strings: `FirstName` and `LastName`. All strings contain only English letters and have length from 1 to 100.

**Output**  
Print `YES` if they are brothers, otherwise print `NO`.

**Examples**

<b>input</b>	<code>Ali Hassan</code>	<code>Omar Hassan</code>	<input type="button" value="Copy"/>
<b>output</b>	<code>YES</code>		
<b>input</b>	<code>Alice Singh</code>	<code>Bob Kumar</code>	<input type="button" value="Copy"/>
<b>output</b>	<code>NO</code>		

Source:

```

1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     string s1, s2;
7     cin >> s1 >> s2;
8
9     string t1, t2;
10    cin >> t1 >> t2;
11
12    if(t1 == t2)
13    {
14        cout << "YES";
15    }
16    else
17    {
18        cout << "NO";
19    }
20
21 }
22

```

Switch off editor      Tab size: 4

**Run**

Language: GNU G++23 14.2 (64 bit, msys2)

Input:  
Ali Hassan  
Omar Hassan

Choose file No file chosen  
No more than 256 KB

Output:  
helloworldhellohello  
=====

Used: 15 ms, 4 KB

First 255 bytes only

14.

## F. Lexicographical Order

time limit per test: 1 second  
memory limit per test: 256 megabytes

You are given two words *A* and *B*.

A word *A* is said to come before *B* in **lexicographical order** (dictionary order) if:

- at the first position where they differ, *A* has a smaller character than *B*, or
- *A* is a prefix of *B* and *A*  $\neq$  *B*.

Determine which word appears first in the dictionary.

### Input

The first line contains a word *A*.

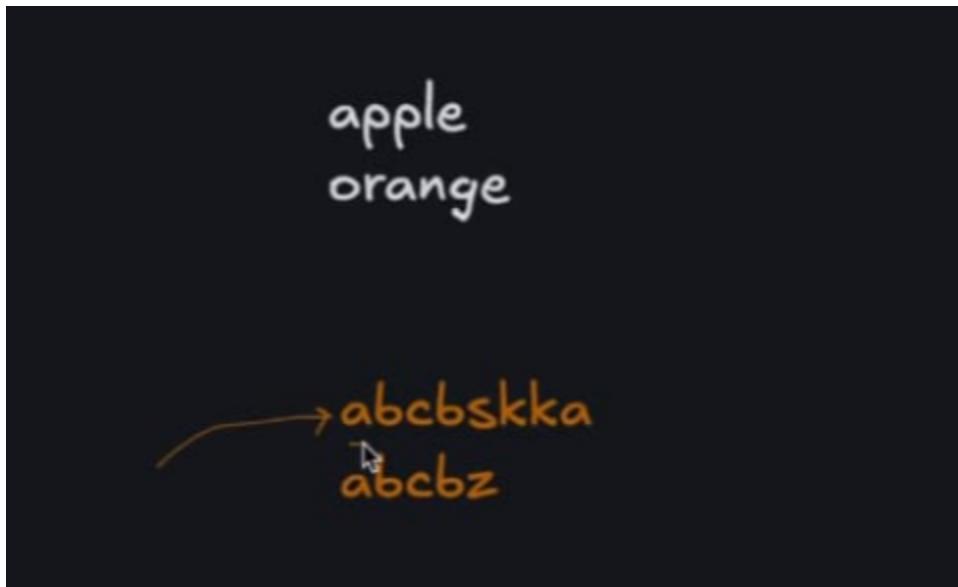
The second line contains a word *B*.

Both words contain only lowercase English letters and have length from 1 to 1000.

### Output

- Print *A* if *A* comes before *B*.
- Print *B* if *B* comes before *A*.
- Print *Equal* if *A* = *B*.

-----



Thu 12 Feb 10:44PM

codeforces.com/group/4vcXCPx8NY/contest/671594/customtest

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     string s, t;
7     cin >> s >> t;
8
9     if(s < t)
10    {
11        cout << s;
12    }
13    else
14    {
15        cout << t;
16    }
17 }
18
19 |
```

Language: G++

Input: Dhruv

Choose file No file chosen  
No more than 256 KB

Output:

apple  
=====

Used: 15 ms, 40 KB

First 255 bytes only

's screen Run

# 15. Palindrome

## M. Palindrome

time limit per test: 1 second  
memory limit per test: 256 megabytes

You are given a string  $S$  consisting only of lowercase English letters.

A string is called a **palindrome** if it reads the same from left to right and from right to left.

Determine whether  $S$  is a palindrome.

### Input

The only line contains a string  $S$  ( $1 \leq |S| \leq 1000$ ).

It is guaranteed that  $S$  consists only of lowercase English letters.

### Output

Print YES if  $S$  is a palindrome, otherwise print NO.

### Examples

input	<input type="button" value="Copy"/>
racecar	<input type="button" value="Copy"/>
output	<input type="button" value="Copy"/>
YES	<input type="button" value="Copy"/>

input	<input type="button" value="Copy"/>
hello	<input type="button" value="Copy"/>
output	<input type="button" value="Copy"/>
NO	<input type="button" value="Copy"/>

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS STANDINGS ADM. EDIT CUSTOM INVOCATION

Source:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     string s;
7     cin >> s;
8
9     char ch1, ch2;
10    cin >> ch1 >> ch2;
11
12    int n = s.size();
13
14    for(int i = 0; i < n; i++)
15    {
16        if(s[i] == ch1)
17        {
18            s[i] = ch2;
19        }
20    }
21
22    cout << s << endl;
23 }
```

Switch off editor      Tab size: 4

Language: GN

Input: abacaba

Output: A

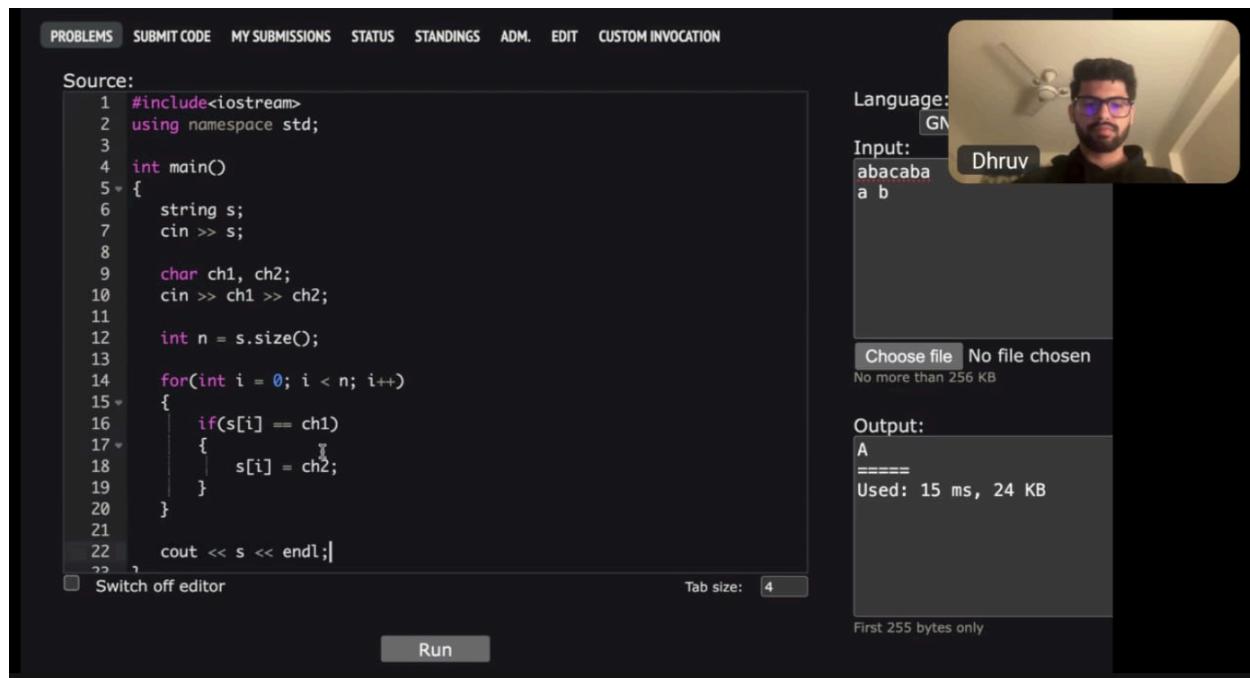
Choose file No file chosen  
No more than 256 KB

Output:  
A  
=====

Used: 15 ms, 24 KB

First 255 bytes only

Run



## 16. Count Words

### N. Count Words

time limit per test: 1 second

memory limit per test: 256 megabytes

You are given a string  $S$  consisting of English letters and spaces.

A word is a maximal contiguous sequence of letters.

It is guaranteed that words are separated by exactly one space, and the first character of  $S$  is not a space.

Your task is to count the number of words in  $S$ .

#### Input

The only line contains the string  $S$ .

It is guaranteed that  $1 \leq |S| \leq 10^5$ , and  $S$  consists only of English letters and spaces.

#### Output

Print one integer — the number of words in  $S$ .

#### Example

##### input

```
hello world this is CP
```

[Copy](#)

##### output

```
5
```

[Copy](#)

```
1 // -----  
2 using namespace std;  
3  
4 int main()  
5 {  
6     string s;  
7     getline(cin, s);  
8  
9     int n = s.size();  
10  
11    int cnt = 0;  
12    for(int i = 0; i < n; i++)  
13    {  
14        if(s[i] == ' ')  
15        {  
16            cnt++;  
17        }  
18    }  
19  
20    cout << cnt + 1 << endl;  
21  
22 }
```

Switch off editor      Tab size: 4

Run

Languag  
Choose  
No more than 1000 characters  
Output:  
bbbcbbb  
=====  
Used: 1  
First 255 bytes