

```

int main(){

    node *start = NULL;

    int n,t;

    cin>>n;

    while(n--){

        cin>>t;

        push(&start,t);

    }

    cout<<"Linked List:";

    printList(start);

    return 0;

    cout<<"p1->next=start; void display()";

}

```

## STACKS:-

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Stack	Question Information
				Level 1 Challenge 41

**Question description**

Hassan enjoys jumping from one building to the next. However, he merely jumps to the next higher building and stops when there are none accessible. The amount of stamina necessary for a voyage is equal to the xor of all the heights Hassan leaps till he comes to a halt.

If heights are [1 2 4], and he starts from 1, goes to 2 stamina required is  $1 \oplus 2 = 3$ , then from 2 to 3. Stamina for the entire journey is  $1 \oplus 2 \oplus 4 = 7$ . Find the maximum stamina required if can start his journey from any building.

**Constraints**

$1 \leq N \leq 10^5$   
 $1 \leq \text{Height} \leq 10^9$

**Input**

First line:  $N$ , no of buildings.  
 Second line:  $N$  integers, defining heights of buildings.

**Output**

Single Integer is the maximum stamina required for any journey.

**Logical Test Cases**

Test Case 1	Test Case 2
INPUT (STDIN) 5 1 2 3 8 6	INPUT (STDIN) 8 1 2 3 8 8 4 7 9

```
#include <stdio.h>
```

```
int main() {
```

```
    int i, j, arr[1000000], n, temp=0, st[1000000]= {0};
```

```
    scanf("%d",&n);
```

```

for(i=0;i<n;i++){
    scanf("%d",&arr[i]);
}
st[n-1] = arr[n-1];
temp = arr[n-1];
for(i=n-2;i>=0;i--) {
    for(j=i+1;j<n;j++)
        if(arr[i]<arr[j]) {
            st[i]=arr[i]^st[j];
            break;
        }
    if(st[i] == 0)
        st[i] = arr[i];
    if(st[i] > temp)
        temp = st[i];
}
printf("%d",temp);
return 0;
}

```

The screenshot shows a web browser window with the URL `care.srmup.in/srmncretelab/#/srmncretelab/student/home`. The page content includes a notification bar stating "You have already solved this challenge! Though you can run the code with different logic!". Below this, a table lists the course (DS), session, stack, and question information (Level 1, Challenge 42). The main section is titled "Problem" and contains the following text:

**Problem Description:**  
 Arumugam is in the process of reorganising her library. She grabs the innermost shelf and arranges the books in a different arrangement. She shatters the shelf's walls. There will be no shelf barriers and simply books in the end. Make a printout of the book order.

Opening and closing walls of shelves are shown by '/' and '\' respectively whereas books are represented by lower case alphabets.

**Constraints:**  
 $2 \leq |S| \leq 10^4$

**Input format**  
 The first line contains string `S` displaying her library.

**Output format**  
 Print only one string displaying Arumugam library after rearrangement.

**Note**  
 The first character of the string is '/' and the last character of the string is '\' indicating outermost walls of the shelf.

Below the problem description, there are two test cases:

Test Case 1	Test Case 2
INPUT (STDIN) /u/hate\l\	INPUT (STDIN) /u/slap\l\
EXPECTED OUTPUT	EXPECTED OUTPUT

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    string s,temp="";

    cin>>s;

    stack<string> stk;

    for (unsigned int i = 0; i < s.size(); i++) {

        if(s[i]==47 | | s[i]==92){

            if(!temp.empty()){

                stk.push(temp);

                temp.clear();

            }

        }

        else{

            temp.push_back(s[i]);

        }

    }

    while(!stk.empty()){

        cout<<stk.top();

        stk.pop();

    }


    return 0;

    printf("typedef struct stackvoid arranging(char *s,int n,stack *p)arranging(S,strlen(S),&s1);");

}

```

srnncrrelab

care.srnup.in/srnncrrelab/#/srnncrrelab/student/home

You have already solved this challenge! Though you can run the code with different logic!

Course DS Session Stack Question Information Level 1 Challenge 43

Question description

Given a permutation of numbers from 1 to N. Among all the subarrays, find the number of unique pairs  $(a, b)$  such that  $a \neq b$  and a is maximum and b is the second maximum in that subarray.

Input:

First line contains an integer,  $N$  ( $1 \leq N \leq 10^5$ ). The second line contains N space-separated distinct integers,  $A_i$  ( $1 \leq A_i \leq N$ ), denoting the permutation.

Output:

Print the required answer.

Explanation:

Sample Input

5

1 2 3 4 5

Sample output

4

Problem

All the possible subarrays are:

1

12

123

1234

12345

2

23

234

2345

```
#include <stdio.h>

int main(){

    int num,i,count=0,a[100001],stck[100001],top=-1;

    scanf("%d", &num);

    for (i=0;i<num;i++) {

        scanf("%d",&a[i]);

        while(top!=-1 && stck[top]<a[i]) {

            top--;

            count++;

        }

        if (top!=-1) {

            count++;

        }

        stck[++top]=a[i];

    }

    printf("%d",count);

    return 0;

}
```

care.srmup.in/srmncrrelab/#/srmncrrelab/student/home

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Stack	Question Information
				Level 1 • Challenge 44

**Problem**

**Problem Description:**  
 You are given an array  $A$  of  $Q$  integers and  $Q$  queries. In each query, you are given an integer  $i$  ( $1 \leq i \leq N$ ).  
 Your task is to find the minimum index greater than  $i$  ( $1 \leq i \leq N$ ) such that:

- Sum of digits of  $A_i$  is greater than the sum of digits of  $A_j$
- $A_i < A_j$

If there is no answer, then print -1.

**Constraints**  
 $1 \leq N, Q \leq 10^5$   
 $1 \leq A_i \leq 10^9$   
 $1 \leq Q_i \leq N$

**Input format**

- The first line contains two numbers  $N$  and  $Q$ .
- The next line contains  $N$  numbers.
- Next  $Q$  lines contain  $Q$  queries.

**Output format**  
 Print the answer as described in the problem

Logical Test Cases

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

```
using namespace std;
```

```
int main(){
```

```
    int n,q;
```

```
    cin>>n>>q;
```

```
    int *a=new int [n];
```

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

```
        cin>>a[i];
```

```
    }
```

```
    int *arr=new int[n];
```

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

```
        int z=a[i];
```

```
        int sum=0;
```

```
        while(z>0){
```

```
            sum+=(z%10);
```

```
            z=z/10;
```

```
        }
```

```
        arr[i]=sum;
```

```
    }
```

```
    while(q--){
```

```
        int Q;
```

```
        cin>>Q;
```

```

int ans=-1;

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

    if(a[i]>a[Q-1] && arr[i]<arr[Q-1]){

        ans=i+1;

        break;

    }else{

        continue;

    }

}

cout<<ans<<' ';

}

return 0;

cout<<"if(arr[x]<arr[y]) if(arr2[x]>arr2[y]) ";

}

```

The screenshot shows a web browser window with the URL `care.srmup.in/srmncretelab/#/srmncretelab/student/home`. A green notification bar at the top states: "You have already solved this challenge ! Though you can run the code with different logic !". Below this, a table with headers "Course", "Session", "Stack", and "Question Information" is visible. The "Question Information" column shows "Level 1" and "Challenge 45".

The main content area is titled "Problem" and contains the following text:

**Question description**  
 You're given a stack of N numbers, with the first component representing the stack's top and the final component being the stack's bottom.  
 At least one piece from the stack must be removed. You can turn the stack into a queue at any time.  
 The front of the line is represented by the bottom of the stack.  
 You cannot convert the queue back into a stack. Your task is to remove exactly K elements such that the sum of the K removed elements is maximized.

**constraints:**  
 $1 \leq N \leq 10^5$   
 $1 \leq K \leq N$   
 $1 \leq A_i \leq 10^9$

**Input format :**

- The first line consists of two space-separated integers N and K.
- The second line consists of N space-separated integers denoting the elements of the stack.

**Output format :**

- Print the maximum possible sum of the K removed elements

Below the problem description, there is a section for "Logical Test Cases" with two tabs: "Test Case 1" and "Test Case 2". Both tabs show "INPUT (STDIN)".

```

#include <bits/stdc++.h>

using namespace std;

int main()

{

    int n,k,i;

    cin>>n>>k;

    int sum = 0;

```

```

int arr[n];

stack<int>st, st2;

for(i=0;i<n;i++){
    cin >> arr[i];
    st.push(arr[i]);
}

for(i=0;i<k;i++){
    st2.push(arr[i]);
    sum += arr[i];
}

int maxs = sum;

while(k-- > 1){
    sum -= st2.top();
    st2.pop();
    sum += st.top();
    st.pop();
    if(sum > maxs) maxs = sum;
}

cout << maxs;

return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Stack	Question Information
				Level 1 • Challenge 46

**Problem Description:**  
 You are given an array  $A$  of  $n$  integers.  
 You have to make a queue and stack the given integers.  
 The queue should contain only prime numbers and the stack should contain only composite numbers.  
 All numbers in the array will be  $> 1$ .  
 The rule to form the stack and queue is that you should be able to generate the array using the pop and dequeue operations.  
 Note: Please read this explanation carefully

Let the array  $A$  contains 5 integers: 7, 21, 18, 3, 12 then the content of queue and stack will be :  
 Queue : 7, 3  
 Stack : 12, 18, 21  
 Now if you follow the rules of stack and queue then you see that you can generate the array using the pop operations of stack and dequeue operation of the queue as follows :

**Problem**

dequeue from the queue: 7  
 pop from stack: 7, 21  
 pop from stack: 7, 21, 18  
 dequeue from queue: 7, 21, 18, 3  
 pop from stack: 7, 21, 18, 3, 12

Thus for every array  $A$  you have to print the contents of the queue in the first line and contents of the stack in the second line.

**Constraints:**  
 $1 \leq n \leq 10^6$   
 $2 \leq A[i] \leq 10^6$

```

#include<bits/stdc++.h>

using namespace std;

bool isPrime(int n)
{
    if(n<=1)

    return false;

    for(int i=2;i<n;i++)

    if(n%i==0)

    return false;

    return true;
}

int main(){

    stack<int> stack;

    int n;

    cin>>n;

    int a[n];

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

        cin>>a[i];

        if(isPrime(a[i]))

        cout<<a[i]<<" ";

        else

        stack.push(a[i]);

    }

    cout<<endl;

    while(!stack.empty()){

        cout<<stack.top()<<" ";

        stack.pop();

    }

    return 0;

    cout<<"int read_int() void push(int stack[],int data) top++;";

}

```



srnncrrelab

care.srnup.in/srnncrrelab/#/srnncrrelab/student/home

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Stack	Question Information
				Level 1 • Challenge 47

**Question description**

Rajnikanth organised technical round interview in Animation company for the set of computer science candidates. the task is to implement stack operations for two stacks and merge the stacks into one. Rajnikanth have given the deadline of only 15 minutes to complete the problem. Can you Help the candidates to complete the problem within the specified time limit ?

**Function Description**

a) push(): Adds the new item at the beginning of linked list using the first pointer.  
b) pop(): Removes an item from the beginning using the first pointer.  
c) merge(): Links the first pointer second stack as next of the last pointer of the first list.

**Problem**

**Constraints**

$0 < n, m < N$   
 $1 < arr[i] < 1000$

**Input Format:**

First line indicates n & m, where n is the number of elements to be pushed into stack and m is the number of pop operation need to be performed  
next line indicates the n number stack elements

**Output Format:**

First line indicates top of the element of the stack  
second line indicates the top of the element after the pop operation

Logical Test Cases

```
#include <iostream>

using namespace std;

class node {
public:
    int data;
    node* next;
};

class mystack {
public:
    node* head;
    node* tail;

    mystack()
    {
        head = NULL;
        tail = NULL;
    }
};

mystack* create()
{
    mystack* ms = new mystack();
    return ms;
}
```

```

}

void push(int data,mystack* ms)
{
    node* temp = new node();
    temp->data = data;
    temp->next = ms->head;
    if (ms->head == NULL)
        ms->tail = temp;

    ms->head = temp;
}

int pop(mystack* ms)
{
    if (ms->head == NULL) {
        cout << "stack underflow" << endl;
        return 0;
    }
    else {
        node* temp = ms->head;
        ms->head = ms->head->next;
        int popped = temp->data;
        delete temp;
        return popped;
    }
}

void merge(mystack* ms1,mystack* ms2)
{
    if (ms1->head == NULL)
    {
        ms1->head = ms2->head;
        ms1->tail = ms2->tail;
        return;
    }

    ms1->tail->next = ms2->head;

```

```

ms1->tail = ms2->tail;
}

void display(mystack* ms)
{
    node* temp = ms->head;
    while (temp != NULL) {
        cout << temp->data << " ";
        temp = temp->next;
    }
}

int main()
{
    mystack* ms1 = create();
    mystack* ms2 = create();
    int n,m,t;
    cin>>n>>m;
    for(int i=0;i<n;i++)
    {
        cin>>t;
        push(t,ms1);
    }
    for(int i=0;i<m;i++)
    {
        cin>>t;
        push(t,ms2);
    }
    merge(ms1, ms2);
    for(int i=0;i<n+m;i++)
        cout<<pop(ms1)<<" ";
}

```

srnmcretelab

care.srmup.in/srnmcretelab/#/srnmcretelab/student/home

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Stack	Question Information	Level 1 • Challenge 49				
<p><b>Problem</b></p> <p><b>Question description:</b>            First off, some definitions.            An array of length at least 2 having distinct integers is said to be <b>fantabulous</b> iff the second highest element lies <b>strictly to the left</b> of the highest value.            For example, <math>[1, 2, 13, 10, 15]</math> is fantabulous as the second-highest value <math>13</math> lies to the left of the highest value <math>15</math>.            For every fantabulous array, we define a fantabulous pair <math>(a, b)</math> where <math>a</math> denotes the index of the second-highest value (1-indexed) and <math>b</math> denotes the index of the highest value (1-indexed).            In the above array, the fantabulous pair is <math>(3, 5)</math>.            Mancunian challenges you to solve the following problem.            Given an array, find the total number of <b>distinct</b> fantabulous pairs overall its subarrays.</p> <p><b>Constraints:</b>  <math>1 \leq N \leq 10^6</math>  <math>1 \leq \text{array elements} \leq 10^9</math>            Array elements are distinct.</p> <p><b>Input:</b>            The first line contains an integer <math>N</math> denoting the length of the array. The next line contains <math>N</math> distinct integers denoting the elements of the array.</p> <p><b>Output:</b>            Output a single integer which is the answer to the problem.</p> <p>Logical Test Cases</p> <table border="1"> <thead> <tr> <th>Test Case 1</th> <th>Test Case 2</th> </tr> </thead> <tbody> <tr> <td>INPUT (STDIN)</td> <td>INPUT (STDIN)</td> </tr> </tbody> </table>						Test Case 1	Test Case 2	INPUT (STDIN)	INPUT (STDIN)
Test Case 1	Test Case 2								
INPUT (STDIN)	INPUT (STDIN)								

22:27 18-11-2021

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

```
using namespace std;
```

```
#define sci(x) scanf("%d", &x)
```

```
#define scl(x) scanf("%lld", &x)
```

```
int arr[1000001], cnt[1000001];
```

```
int v[1000001];
```

```
stack <int> st;
```

```
void don(){
```

```
    cout<<"void push(llint num)stack[top++]=num;pop()";
```

```
}
```

```
int main()
```

```
{
```

```
    int n, i, x;
```

```
    sci(n);
```

```
    for (i = 1; i <= n; ++i) sci(arr[i]);
```

```
    for (i = n; i > 0; --i) {
```

```

        while (!st.empty() && arr[i] > arr[st.top()]) {
            cnt[st.top()] = st.top() - i;
            st.pop();
        }
        st.push(i);
    }

    while (!st.empty()) {
        cnt[st.top()] = st.top();
        st.pop();
    }

    for (i = 1; i <= n; ++i) {
        while (!st.empty() && arr[st.top()] < arr[i]) {
            x = i - st.top() + 1;
            v[x] = max(v[x], cnt[st.top()]);
            st.pop();
        }
        st.push(i);
    }

    int k = 0;
    for (i = 2; i <= n; ++i) {
        k += v[i];
    }

    cout << k << endl;

    return 0;
}

```

srnncrrelab

care.srmup.in/srnncrrelab/#/srnncrrelab/student/home

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Stack	Question Information
				Level 1 Challenge 48

**Question description**

Hassan gets a job in a software company in Hyderabad. The training period for the first three months is 20000 salary. Then incremented to 25000 salaries. Training is great but they will give you a programming task every day in three months. Hassan must finish it in the allotted time. His teammate Jocelyn gives him a task to complete the concept of Prefix to Postfix Conversion for a given expression. can you help him?

**Functional Description:**

- Read the Prefix expression in reverse order (from right to left)
- If the symbol is an operand, then push it onto the Stack
- If the symbol is an operator, then pop two operands from the Stack

Create a string by concatenating the two operands and the operator after them.  
 $string = operand1 + operand2 + operator$   
 And push the resultant string back to Stack

- Repeat the above steps until end of Prefix expression.

**Constraints**

the input should be a expressions

**Input Format**

Single line represents the prefixed expressions

**Output Format**

Single line represents the postfix expression

Logical Test Cases

Test Case 1

Test Case 2

23:10 09-11-2021

```
#include <iostream>

#include <stack>

using namespace std;

bool isOperator(char x)
{
    switch (x) {
        case '+':
        case '-':
        case '/':
        case '*':
            return true;
    }
    return false;
}

string preToPost(string pre_exp)
{
    stack<string> s;

    int length = pre_exp.size();
    for (int i = length - 1; i >= 0; i--)
    {
        if (isOperator(pre_exp[i]))
        {

```

```

        string op1 = s.top();

        s.pop();

        string op2 = s.top();

        s.pop();

        string temp = op1 + op2 + pre_exp[i];

        s.push(temp);

    }

    else {

        s.push(string(1, pre_exp[i]));

    }

}

return s.top();

}

int main()

{

    string pre_exp;

    cin>>pre_exp;

    cout << "Postfix:" << preToPost(pre_exp);

    return 0;

}

```

The screenshot shows a web browser window with the URL `care.srmup.in/srmncretelab/#/srmncretelab/student/home`. A notification at the top states: "You have already solved this challenge! Though you can run the code with different logic!". Below this is a table with columns: Course, Session, Stack, Question Information, and a progress indicator showing "Level 1" and "Challenge 50".

Course	Session	Stack	Question Information	Progress
DS				Level 1 Challenge 50

**Question description**  
 Hassan gets a job in a software company in Hyderabad. The training period for the first three months is 20000 salary. Then incremented to 25000 salaries. Training is great but they will give you a programming task every day in three months. Hassan must finish it in the allotted time. His teammate Jocelyn gives him a task to complete the concept of Postfix to Infix Conversion for a given expression. can you help him?

**Functional Description:**

- Read the next symbol from the input.
- Push it onto the stack.
- the symbol is an operator.
- Pop the top 2 values from the stack.
- Put the operator, with the values as arguments and form a string.
- Push the resulted string back to stack.
- If there is only one value in the stack
- That value in the stack is the desired infix string.

**Constraints**  
 the input should be a expressions

**Input Format**  
 Single line represents the postfix expressions

**Output Format**  
 Single line represents the infix expression

Logical Test Cases

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

```

#include<iostream>

#include<string.h>

using namespace std;

bool isOperand(char x){
    return (x>='a' && x<='z') || (x >= 'A' && x <= 'Z');
}

string getInfix(string exp)
{
    stack<string> s;

    for(int i=0; exp[i]!='\0'; i++)
    {
        if(isOperand(exp[i]))
        {
            string op(1, exp[i]);
            s.push(op);
        }

        else
        {
            string op1 = s.top();
            s.pop();

            string op2=s.top();
            s.pop();

            s.push("(" + op2 + exp[i] + op1 + ")");
        }
    }

    return(s.top());
}

int main()
{
    string exp;
    cin>>exp;

```



```

cout<<getInfix(exp);

return 0;

}

```

## QUEUES:-

The screenshot shows a web browser window with the URL `care.srmup.in/srmncretelab/#/srmncretelab/student/home`. A green notification bar at the top says: "You have already solved this challenge ! Though you can run the code with different logic !". Below this, there's a navigation bar with tabs: "Course", "DS", "Session", "Queue", and "Question Information". The "Queue" tab is active. The main content area shows a "Question description" and a "Function Description".

**Question description**

Sathya is an DS expert training youngsters struggling in DS to make them better. Sathya usually gives interesting problems to the youngsters to make them love the DS. One such day Sathya provided to the youngsters to solve the task such that, insert an element in a Queue in FIFO order. Youngsters were lacking the idea to solve the problem. Being an exciting youngster can you solve it?

**Function Description**

1. Define the maximum size of queue and initialize front and rear as -1.
2. In the main function we will initialize two variables that will store the data and the size of the queue.
3. Accept the data that we want to enter in a queue using a for loop.
4. After accepting the data use `enqueue()` function to insert the data in a queue.

```

#include <stdio.h>

#define SIZE 100

void enqueue(int);

void display();

int items[SIZE], front = -1, rear = -1;

int main() {

    int n,data,i;

    scanf("%d",&n);

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

    {

        scanf("%d",&data);

        enqueue(data);

        display();

    }

}

```