Result & Analysis

Student: Smit Barmase Test: PREP_DSA_PT_Stacks Course: 2023 Data Structure...

Attempt 1

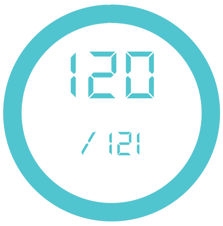
IP Address: 2401:4900:1c16:5745:a220:f449:d645:31ea **Tab switches:** 0 **OS used:** Linux

Browser used: Chrome

Test Duration: 02:22:20 **Test Start Time:** Oct 22, 2021 | 04:12 PM

Test Submit Time: Nov 6, 2021 | 05:33 PM **Resume Count:** 1

Overall score



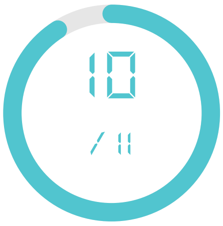
Rank: NA

Topper score: 121.00 / 121

Average score: 22.83 / 121

Least score: 0.00 / 121

MCQ



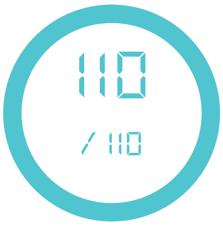
Rank: NA

Topper score: 11.00 / 11

Average score: 7.67 / 11

Least score: 0.00 / 11

Coding



Rank: NA

Topper score: 110.00 / 110

Average score: 25.38 / 110

Least score: 0.00 / 110

Overall Question Status

Total Questions: 22

Questions Attempted: 22

Questions Correct: 21

Question Wrong: 1

Partially Correct: 0

Question Not Viewed: 0

MCQ - Question Status

Total Questions: 11

Questions Attempted: 11

Questions Correct: 10

Question Wrong: 1

Partially Correct: 0

Question Not Viewed: 0

Coding - Question Status

Total Questions: 11

Questions Attempted: 11

Questions Correct: 11

Question Wrong: 0

Partially Correct: 0

Question Not Viewed: 0

Topic wise Analysis MCQ Coding

Question No: 1 **Single File Programming Question**

Ankit along with Karthick has opened new PVC pipe shop. They have 3 stacks of cylindrical pipes where each pipe has the equal diameter, but they may vary in height. They can change the height of a stack by removing and discarding its topmost pipe any number of times.

Find the maximum possible height of the stacks such that all of the stacks are exactly the same height. This means they must remove zero or more cylindrical pipes from the top of zero or more of the three stacks until they're all the same height, and then print the height. The removals must be performed in such a way as to maximize the height. Also, remember, an empty stack is still a stack.

Karthick is not clear about this arrangement. So Ankit, explained with an example. (Refer Sample Input)
There are 3 stacks with different height. To make all stacks of equal height, they remove the first two cylinders from stack 1 and one cylinder each from stack 2 and 3.

Input format

The 1st line contains three space-separated integers N_1, N_2 and N_3 (i.e Number of pipes in stacks 1,2 ,3)
The 2nd line contains N1 space-separated integers describing the pipe heights in stack 1. The first element is the top of the stack.
The 3rd line contains N2 space-separated integers describing the pipe heights in stack 2. The first element is the top of the stack.
The 4th line contains N3 space-separated integers describing the pipe heights in stack 3. The first element is the top of the stack.

Output format

Display the maximum height at which all stacks will be of equal height.

Sample testcases

Input 1

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

Output 1

```
5
```

C (17)

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

int main() {
    int n1, n2, n3;
    cin >> n1 >> n2 >> n3;
    vector<int> a(n1), b(n2), c(n3);
    for (int i = 0; i < n1; i++) cin >> a[i];
    for (int i = 0; i < n2; i++) cin >> b[i];
    for (int i = 0; i < n3; i++) cin >> c[i];
    int s1 = accumulate(a.begin(), a.end(), 0);
    int s2 = accumulate(b.begin(), b.end(), 0);
    int s3 = accumulate(c.begin(), c.end(), 0);
    int i = 0, j = 0, k= 0;
    while (true) {
        if ((s1 == s2 && s2 == s3) || s1 == 0 || s2 == 0 || s3 == 0) brea
        if (s1 >= s2 && s1 >= s3) s1 -= a[i++];
        else if (s2 >= s1 && s2 >= s3) s2 -= b[j++];
        else s3 -= c[k++];
    }
    if (s1 == 0 || s2 == 0 || s3 == 0) cout << 0;
    else cout << s1;
    return 0;
}
```

Recommended Learning Content: Introduction to Stacks Stacks Implementation Stacks

Status: Correct **Mark obtained:** 10/10 **Hints used:** 0

Times compiled: 3 **Times submitted:** 1 **Level:** Medium

Question type: Single File Programming **Subject:** Programming

Subject: Data Structures **Subject:** Stacks

□ Show testcase scores □ Show solution

Chandhana is given a set of numbers in an array which has to be placed in a stack. She writes a code to implement it but gets stuck halfway. Analyze her code and help her complete it by filling up the missing part.

Input format

First line has the number of array elements and the second line has the elements separated by a space.

Output format

The output prints the stack elements in first line.

The second line prints the stack elements after performing 2 pop operations.

Sample testcases

Input 1

```
5
23 569 85 12 56
```

Output 1

```
56 12 85 569 23
85 569 23
```

Header Snippet

```
struct Stack
{
    int top;
    int stack[50];
};
```

C (17)

```c
void push(struct Stack *S, int arr[], int n) {
    for (int i = 0; i < n; i++) {
        S->top++;
        if (S->top >= 50) break;
        S->stack[S->top] = arr[i];
    }
}

void pop(struct Stack *S) {
    if (S->top < 0) return;
    else S->top--;
}
```

Footer Snippet

```c
void show(struct Stack *S)
{
    for (int i=S->top;i>=0;i--)
    {
        printf("%d ",S->stack[i]);
    }
    if(S->top == -1)
    {
        printf("Stack is empty");
```

Recommended Learning Content: Introduction to Stacks Stacks Implementation Stacks

Status: Correct **Mark obtained:** 10/10 **Hints used:** 0

Times compiled: 4 **Times submitted:** 1 **Level:** Easy

Question type: Single File Programming **Subject:** Programming

Subject: Data Structures **Subject:** Stacks

□ Show testcase scores □ Show solution

Question No: 3 Single File Programming Question

Write a program to check whether the string is palindrome or not using Stack

Input format

Input consist of a string

Output format

Output prints whether the string is palindrome or not

Refer Sample output for exact format

Sample testcases

Input 1	Output 1
`malayalam`	`malayalam is palindrome`
Input 2	**Output 2**
`raser`	`raser is not a palindrome`

C++ (17)

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

int main() {
    string s;
    cin >> s;
    int n = s.size();
    stack<char> st;
    for (int i = 0; i < n; i++) st.push(s[i]);
    int ok = 1;
    for (int i = 0; i < n; i++) {
        if (s[i] != st.top()) {
            ok = 0;
            break;
        }
        st.pop();
    }
    if (ok) cout << s << " is palindrome";
    else cout << s << " is not a palindrome";
    return 0;
}
```

Deleted element in the stack

Stack elements as shown in sample output

Recommended Learning Content: Introduction to Stacks Stacks Implementation Stacks

Sample testcases

Input 1

```
5
3 6 4 9 8
```

Output 1

```
Deleted element is 8
The elements in stack
9
4
6
3
```

□ Show testcase scores □ Show solution

C++ (17)

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

int main() {
    int n;
    cin >> n;
    stack<int> s;
    for (int i = 0; i < n; i++) {
        int x;
        cin >> x;
        s.push(x);
    }
    cout << "Deleted element is " << s.top() << '\n';
    cout << "The elements in stack\n";
    s.pop();
    while (s.size()) {
        cout << s.top() << '\n';
        s.pop();
    }
    return 0;
}
```

Recommended Learning Content: Introduction to Stacks Stacks Implementation Stacks

Question No: 5 **Single File Programming Question**

Status: Correct **Mark obtained:** 10/10 **Hints used:** 0

Times compiled: 3 **Times submitted:** 1 **Level:** Easy

Question type: Single File Programming **Subject:** Programming

Subject: Data Structures **Subject:** Stacks

□ Show testcase scores □ Show solution

Write a program to implement a stack and search for an element in the stack.

Input format

Number of stack elements in first line
Stack elements in second line separated by space
Element to be searched in third line

Output format

Output prints whether the element is found or not

Sample testcases

Input 1

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

Output 1

```
Element found
```

Input 2

```
5
1 2 3 4 5
37
```

Output 2

```
Element not found
```

Header Snippet

```
#include<stdio.h>
#include<stdlib.h>
int *stack=(int*)malloc(sizeof(int)*100);
int n,top,x,i;
void push();
void display();
int main()
{
    top=-1;
```

C (17)

Question No: 6 **Single File Programming Question**

```
void display() {
    int y;
    scanf("%d", &y);
    int ok = 0;
    while (top > -1) {
        if (stack[top] == y) {
            ok = 1;
            break;
        }
        top--;
    }
    if (ok) printf("Element found");
    else printf("Element not found");
}
```

Footer Snippet

```
void push()
{
    if(top>=n-1)
    {
        printf("STACK is over flow");

    }
    else
    {
```

Recommended Learning Content: Introduction to Stacks Stacks Implementation Stacks

Status: Correct **Mark obtained:** 10/10 **Hints used:** 0

Times compiled: 10 **Times submitted:** 2 **Level:** Easy

Header Snippet

```
#include <stdio.h>
#include<stdlib.h>
int *stack=(int*)malloc(sizeof(int)*100);
int top = -1;
void push(int *stack, int val);
int peek(int *stack);
int main()
{
int val.n:
```

C (17)

```
// You are using GCC
int peek(int *stack) {
    if (top == -1) return -1;
    return stack[top];
}
```

Footer Snippet

```
void push(int *stack, int val)
{
if(top == 100-1)
{
printf("STACK OVERFLOW");
}
else {
top++;
*(stack+top) = val:
```

Number of elements in first line

Stack elements in second line separated by space

Output format

Elements in descending order separated by space

Recommended Learning Content: Introduction to Stacks   Stacks Implementation

Stacks

Status: Correct **Mark obtained:** 10/10 **Hints used:** 0

Times compiled: 1 **Times submitted:** 1 **Level:** Easy

Sample testcases

Question type: Single File Programming **Subject:** Programming

Input 1	**Output 1**
5 8 10 4 2 7	10 8 7 4 2

□ Show testcase scores □ Show solution

C++ (17)

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

int main() {
    int n;
    cin >> n;
    stack<int> s1, s2;
    for (int i = 0; i < n; i++) {
        int x;
        cin >> x;
        while (!s1.empty() && s1.top() > x) {
            s2.push(s1.top());
            s1.pop();
        }
        s1.push(x);
        while (!s2.empty()) {
            s1.push(s2.top());
            s2.pop();
        }
    }
    while (!s1.empty()) {
        cout << s1.top() << " ";
        s1.pop();
    }
    return 0;
}
```

Recommended Learning Content: Introduction to Stacks Stacks Implementation Stacks

Status: Correct **Mark obtained:** 10/10 **Hints used:** 0

Times compiled: 2 **Times submitted:** 1 **Level:** Medium

Question type: Single File Programming **Subject:** Programming

Subject: Data Structures **Subject:** Stacks

□ Show testcase scores □ Show solution

Question No: 8 **Single File Programming Question**

Write a program to check whether the given expression has *balanced parenthesis* or not.

Input format

First line of input contanis an expression with parenthesis

Output format

Display the given expression in first line
Print **BALANCED** or **NOT BALANCED** in next line.

Code constraints

Expression length ≤ 100

Sample testcases

Input 1	Output 1
(f+d){	(f+d){ NOT BALANCED

Input 2	Output 2
{d-f}	{d-f} BALANCED

C++ (17)

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

int main() {
    string s;
    cin >> s;
    int n = s.size();
    stack<char> st;
    for (int i = 0; i < n; i++) {
        if ((s[i] == '(') || (s[i] == '{') || (s[i] == '[')) st.push(s[i]
        else if (s[i] == ')') {
            if (!st.empty() && (st.top() == '(')) st.pop();
            else st.push(s[i]);
        } else if (s[i] == '}') {
            if (!st.empty() && (st.top() == '{')) st.pop();
            else st.push(s[i]);
        } else if (s[i] == ']') {
            if (!st.empty() && (st.top() == '[')) st.pop();
            else st.push(s[i]);
        }
    }
    cout << s << '\n';
    if (st.empty()) cout << "BALANCED";
    else cout << "NOT BALANCED";
    return 0;
}
```

Recommended Learning Content: Introduction to Stacks Stacks Implementation Stacks

Status: Correct **Mark obtained:** 10/10 **Hints used:** 0

Times compiled: 2 **Times submitted:** 2 **Level:** Medium

Question type: Single File Programming **Subject:** Programming

Subject: Data Structures **Subject:** Stacks

□ Show testcase scores □ Show solution

Question No: 9 **Single File Programming Question**

Write a program to convert an infix expression to a post-fix expression using Stack concept.

Input format

Input consists of an infix expression.

Output format

Output displays the post-fix expression of the input.

Sample testcases

Input 1

```
(a+b)*(a-b)
```

Output 1

```
ab+ab-*
```

Input 2

```
A+(B*C-(D/E^F)*G)*H
```

Output 2

```
ABC*DEF^/G*-H*+
```

C++ (17)

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

int prec(char c) {
    if (c == '^') return 3;
    else if (c == '/' || c == '*') return 2;
    else if (c == '+' || c == '-') return 1;
    else return -1;
}

string get(string &s) {
    stack<char> st;
    string ans;
    int n = s.size();
    for (int i = 0; i < n; i++) {
        char c = s[i];
        if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c >= '0
            ans += c;
        } else if (c == '(') {
            st.push('(');
        } else if (c == ')') {
            while (st.top() != '(') {
                ans += st.top();
                st.pop();
            }
            st.pop();
        } else {
```

Write a program to print all the factors of a number using stack through array implementation.

Input format

First line of the input is a Number

Output format

Print all the factors of the number separated by space

Code constraints

0<N<999999

Sample testcases

Input 1	**Output 1**
`5`	`5 1`

Input 2	**Output 2**
`-1`	`WRONG INPUT`

Recommended Learning Content: Introduction to Stacks Stacks Implementation Stacks

Status: Correct **Mark obtained:** 10/10 **Hints used:** 0

Times compiled: 2 **Times submitted:** 1 **Level:** Medium

Question type: Single File Programming **Subject:** Programming

Subject: Data Structures **Subject:** Stacks

□ Show testcase scores □ Show solution

Header Snippet

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

C (17)

```c
// You are using GCC
void PFactors(int num) {
    int stack[MAX];
    int top = -1;
    for (int i = 1; i <= num; i++) {
        if ((num % i) == 0) push(i, &top, stack);
    }
    while (top != -1) {
        printf("%d ", stack[top]);
        pop(&top, stack);
    }
}
```

Footer Snippet

```c
int push(int x, int *top, int stack_arr[])
{
    if(*top == (MAX-1))
    {
        printf("Stack Overflow\n");
        return 1;
    }
    else
    {
```

Recommended Learning Content: Introduction to Stacks Stacks Implementation Stacks

Status: Correct **Mark obtained:** 10/10 **Hints used:** 0

Times compiled: 14 **Times submitted:** 1 **Level:** Medium

Question type: Single File Programming **Subject:** Programming

□ Show testcase scores □ Show solution

Subject: Data Structures **Subject:** Stacks

Question No: 11 **Single File Programming Question**

Write a program to reverse a string **s** using stack.

Input format

Input consists of a string

Output format

Display the reversed string

Code constraints

1 ≤ s ≤ 100

Sample testcases

Input 1	Output 1
`kumar`	`ramuk`
Input 2	**Output 2**
`Sample Input`	`tupnI elpmaS`

C++ (17)

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

int main() {
    string s;
    getline(cin, s);
    int n = s.size();
    stack<char> st;
    for (int i = 0; i < n; i++) st.push(s[i]);
    while (!st.empty()) {
        cout << st.top();
        st.pop();
    }
    return 0;
}
```

Recommended Learning Content: Introduction to Stacks Stacks Implementation Stacks

Status: Correct **Mark obtained:** 10/10 **Hints used:** 0

Times compiled: 2 **Times submitted:** 1 **Level:** Medium

Question type: Single File Programming **Subject:** Programming

Subject: Data Structures **Subject:** Stacks

□ Show testcase scores □ Show solution