



Experiment 2

Student Name: Ishu

Branch: BE-CSE

Semester: 5th

Subject Name: AP

UID: 22BCS15695

Section/Group: 603_FL_IOT-B

Date of Performance: 29/07/2024

Subject Code: 22CSP-314

1. Aim:

To find the maximum possible height of the stacks such that all of the stacks are exactly the same height.

2. Objective: In this experiment, we must remove zero or more cylinders from the top of zero or more of the three stacks until they are all the same height, then return the height. For this we have two approaches which are as following.

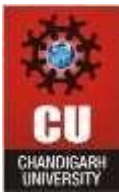
3. Implementation/Code:

A.

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

int calculateSum(stack<int>& stack1, stack<int>& stack2, int maxSum) {
    int sum = 0;
    while (!stack1.empty() && !stack2.empty()) {
        int top1 = stack1.top();
        int top2 = stack2.top();

        if (top1 < top2) {
            if (sum + top1 > maxSum) break;
            sum += top1;
            stack1.pop();
        } else {
            if (sum + top2 > maxSum) break;
            sum += top2;
            stack2.pop();
        }
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
    }  
    return sum;  
}  
  
int main() {  
    stack<int> stack1;  
    stack<int> stack2;  
  
    // Push elements into stack1  
    stack1.push(5);  
    stack1.push(3);  
    stack1.push(4);  
  
    // Push elements into stack2  
    stack2.push(3);  
    stack2.push(2);  
    stack2.push(1);  
    stack2.push(1);  
  
    int maxSum = 6; // Maximum sum allowed  
  
    int sum = calculateSum(stack1, stack2, maxSum);  
    cout << "The calculated sum is: " << sum << endl;  
  
    return 0;  
}
```

B.

```
#include <iostream>  
  
using namespace std;  
  
int getMaxEqualHeight(int stacks[3][100], int sizes[3]) {  
    // Calculate the total height of each stack  
    int heights[3] = {0, 0, 0};  
    for (int i = 0; i < 3; ++i) {  
        for (int j = 0; j < sizes[i]; ++j) {  
            heights[i] += stacks[i][j];  
        }  
    }  
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
// Use indices to keep track of the current top cylinder for each stack
int indices[3] = {0, 0, 0};

while (true) {
    // Check if all stacks have the same height
    if (heights[0] == heights[1] && heights[1] == heights[2]) {
        return heights[0];
    }

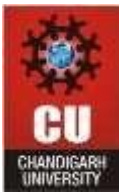
    // Find the stack with the greatest height
    int maxHeightIndex = 0;
    for (int i = 1; i < 3; ++i) {
        if (heights[i] > heights[maxHeightIndex]) {
            maxHeightIndex = i;
        }
    }

    // Remove the top cylinder from the stack with the greatest height
    if (indices[maxHeightIndex] < sizes[maxHeightIndex]) {
        heights[maxHeightIndex] -=
        stacks[maxHeightIndex][indices[maxHeightIndex]];
        ++indices[maxHeightIndex];
    } else {
        // If we cannot remove more cylinders, break the loop
        break;
    }
}

return 0; // If no common height can be found
}

int main() {
    // Example stacks (with maximum number of cylinders set to 100)
    int stacks[3][100] = {
        {5, 3, 4}, // Stack 1
        {3, 2, 1, 1}, // Stack 2
        {1, 2, 1} // Stack 3
    };

    // The number of cylinders in each stack
    int sizes[3] = {3, 4, 3};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
int maxEqualHeight = getMaxEqualHeight(stacks, sizes);  
cout << "The maximum possible height of the stacks such that all are  
equal is: " << maxEqualHeight << endl;
```

```
    return 0;  
}
```

4. Output:

A.

```
The calculated sum is: 4
```

```
...Program finished with exit code 0  
Press ENTER to exit console.□
```

B.

```
The maximum possible height of the stacks such that all are equal is: 4
```

```
...Program finished with exit code 0  
Press ENTER to exit console.□
```