



Experiment : 1.2

Student Name: SANJIV GUPTA

Branch: CSE

Semester: 5th

Subject Name: Advanced Programming LAB

UID: 21BCS-3478

Section/Group: 21BCS-IOT-602B

Date: 25/08/23

Subject Code: 21CSP-314

AIM: *To implement the concept of Stack and Queues.*

OBJECTIVE:

1) *You have three stacks of cylinders where each cylinder has the same diameter, but they may vary in height. You can change the height of a stack by removing and discarding its topmost cylinder 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 you 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.

2) *Alexa has two stacks of non-negative integers, stack $a[n]$ and stack $b[m]$ where index 0 denotes the top of the stack. Alexa challenges Nick to play the following game:*

- In each move, Nick can remove one integer from the top of either stack a or stack b .*
- Nick keeps a running sum of the integers he removes from the two stacks.*
- Nick is disqualified from the game if, at any point, his running sum becomes greater than some integer $maxSum$ given at the beginning of the game.*
- Nick's final score is the total number of integers he has removed from the two stacks.*

CODE:

Code 1:

```
def solve(petrol, distance, n):  
    start, tank = 0, 0  
    for i in range(n):  
        start = i  
        tank = 0  
        for j in range(n):  
            current = (start + j) % n  
            tank += petrol[current] - distance[current]  
            if tank < 0:
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        break
    if j == n - 1:
        return start
    return -1

n = int(input())
petrol, distance = [], []
for i in range(n):
    p, d = [int(x) for x in input().split()]
    petrol.append(p)
    distance.append(d)
print(solve(petrol, distance, n))
```

Code 2:

```
def brute(a, b):
    ans = 0
    a = [0] + a
    b = [0] + b
    for i in range(len(a)):
        for j in range(len(b)):
            sa = sum(a[:i + 1])
            sb = sum(b[:j + 1])
            if sa + sb > x:
                break
            ans = max(ans, i + j)
    return ans
```

```
def solve(a, b):
    total = 0
    i = len(a)
    ans = i
    j = 1
    for i in range(len(a)):
        total += a[i]
        if total > x:
            ans = i
            break
    ans_total = x + 1
    total = sum(a[:i])
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
while (total <= x and i > 0) or j < len(b):
```

```
    if total < x:
```

```
        total += b[j - 1]
```

```
        j += 1
```

```
    elif total > x:
```

```
        total -= a[i - 1]
```

```
        i -= 1
```

```
    else:
```

```
        total -= a[i - 1]
```

```
        i -= 1
```

```
        total += b[j - 1]
```

```
        j += 1
```

```
    if i < 1 and total > x:
```

```
        break
```

```
    ans = max(ans, i + j - 2)
```

```
    ans_total = min(total, ans_total)
```

```
return ans if ans_total <= x else ans - 1
```

```
for _ in range(int(input())):
```

```
    n, m, x = map(int, input().split())
```

```
    a = [int(x) for x in input().split()]
```

```
    b = [int(x) for x in input().split()]
```

```
    _sum_a, _sum_b = sum(a), sum(b)
```

```
    if _sum_a + _sum_b <= x:
```

```
        print(n + m)
```

```
    elif n <= 200 and m <= 200:
```

```
        print(brute(a, b))
```

```
    else:
```

```
        ans = solve(a, b) if _sum_a < _sum_b else solve(b, a)
```

```
        print(ans)
```

OUTPUT:

OUTPUT 1

Congratulations

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)

[Next Challenge](#)

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

Success

Input (stdin)

1	3
2	1 5
3	10 3
4	3 4

Download

Expected Output

1	1
---	---

Download

Activate Windows
Go to Settings to activate Windows.

OUTPUT 2

Congratulations

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)

[Next Challenge](#)

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

Success

Input (stdin)

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

Download

Expected Output

1	4
---	---

Download

Activate Windows
Go to Settings to activate Windows.

LEARNING OUTCOMES:

1. *Learn about stack manipulation technique.*
2. *Learn about stack and queues conditional logic.*
3. *Learn about algorithm thinking*
4. *Learn about mathematical logic*