



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment- 1.2

Student Name: UTKARSH JOSHI

Branch: CSE-Gen

Semester: 5th

Subject Name: Advanced Programming

UID: 21BCS9158

Section/Group: ST 802-A

Date of Performance: 09/08/23

Subject Code: 21CSP-259

1. Aim:

Solve the following problems on hackerrank:

1. Balanced Brackets
2. Down to Zero II

2. Objective: To perform different operations on stack and queue.

3. Code :

Program -1

```
def check():
    stack = []
    s = input()
    for c in s:
        #print(c)
        if c == '(':
            stack.append(0)
        elif c == ')':
            if len(stack) > 0 and stack[-1] == 0:
                stack.pop()
            else:
                return -1
        elif c == '[':
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        stack.append(2)
    elif c == ']':
        if len(stack) > 0 and stack[-1] == 2:
            stack.pop()
        else:
            return -1
    if c == '{':
        stack.append(4)
    elif c == '}':
        if len(stack) > 0 and stack[-1] == 4:
            stack.pop()
        else:
            return -1

    if len(stack) == 0:
        return 0
    else:
        return -1

def solve():
    t = int(input())
    for i in range(0,t):
        if check() == 0:
            print("YES")
        else:
            print("NO")
solve()
```

Program -2

```
import collections
lim = 10**6+1
dist = [0]*lim
```



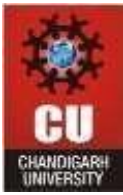
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
active = collections.deque()
active.append(0)
while active:
    n = active.popleft()
    d = dist[n]+1
    x = n + 1
    if x < lim and dist[x] == 0:
        dist[x] = d
        active.append(x)
    for m in range(2,n+1):
        x = m * n
        if x >= lim: break
        if dist[x] == 0:
            dist[x] = d
            active.append(x)
Q = int(input())
for q in range(Q):
    N = int(input())
    print(dist[N])
```

5. Output:

Program 1:



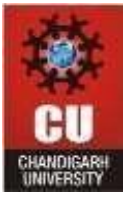
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

The screenshot shows the HackerRank interface for the 'Down to Zero II' problem. The left sidebar contains navigation links: Problem, Submissions, Leaderboard, and Discussions. The main content area on the left details the problem: given Q queries, each with a number N , you can perform two operations: 1) If $N = a \times b$ ($a \neq 1, b \neq 1$), change N to $\max(a, b)$; 2) Decrease N by 1. The goal is to find the minimum moves to reduce N to 0. Constraints are $1 \leq Q \leq 10^3$ and $0 \leq N \leq 10^6$. The sample input is 2, 3, 4, and the sample output is 2, 3, 4. The right side shows a code editor with a Python solution, a 'Run Code' button, and a 'Submit Code' button. Below the code editor, there are test cases 0 through 5, all marked as successful.

Program 2:

The screenshot shows the HackerRank interface for the 'Balanced Brackets' problem. The left sidebar contains navigation links: Problem, Submissions, Leaderboard, and Discussions. The main content area on the left details the problem: a bracket is considered to be any one of the following characters: (,), {, }, [, or]. Two brackets are considered to be a matched pair if an opening bracket (i.e., (, [, or {) occurs to the left of a closing bracket (i.e.,),], or }) of the exact same type. There are three types of matched pairs of brackets: [], {}, and (). A matching pair of brackets is not balanced if the set of brackets it encloses are not matched. For example, [{()}] is not balanced because the contents in between { and } are not balanced. The pair of square brackets encloses a single, unbalanced opening bracket, (, and the pair of parentheses encloses a single, unbalanced closing square bracket,]. By this logic, we say a sequence of brackets is balanced if the following conditions are met: It contains no unmatched brackets. The subset of brackets enclosed within the confines of a matched pair of brackets is also a matched pair of brackets. Given n strings of brackets, determine whether each sequence of brackets is balanced. If a string is balanced, return YES. Otherwise, return NO. The function description is to complete the function isBalanced in the editor below. isBalanced has the following parameter(s): string s: a string of brackets. The returns are: string: either YES or NO. The input format is not specified. The right side shows a code editor with a Python solution, a 'Run Code' button, and a 'Submit Code' button. Below the code editor, there are test cases 0 through 6, all marked as successful. A 'Hidden Test Case' is also shown, which is locked for 5 hacks.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

6. Learning outcomes:

1. Develop essential skills in array manipulation, traversal, and indexing.
2. Strengthen your logical reasoning and problem-solving abilities through array comparisons and calculations.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.