Submitted Code

Language: Python 3 Open in editor

```
import os
import random
import re
import sys

#
# Complete the 'insertionSort1' function below.
#
# The function accepts following parameters:
#  1. INTEGER n
#  2. INTEGER_ARRAY arr
#

def insertionSort1(n, arr):
    key = arr[-1]
    i = n - 2
```

Test case 0

Test case 1

Test case 2

Compiler Message

Success

Input (stdin) Download

Submitted Code

Language: Python 3 Open in editor

```
#!/bin/python3

import math
import os
import random
import re
import sys

#
# Complete the 'insertionSort2' function below.
#
# The function accepts following parameters:
#  1. INTEGER n
#  2. INTEGER_ARRAY arr
#
```

Test case 0

Test case 1

Compiler Message

Success

Submitted Code

Language: Python 3 Open in editor

```
#!/bin/python3

import math
import os
import random
import re
import sys

#
# Complete the 'quickSort' function below.
#
# The function is expected to return an INTEGER_ARRAY.
# The function accepts INTEGER_ARRAY arr as parameter.
#

```

Test case 0

Test case 1

Compiler Message

Success

Submitted Code

Language: C

Open in editor

```c
int parse_int(char*);

// Name: Teja K
// USN: 1DT23CA015

int* countingSort(int arr_count, int* arr, int* result_count) {
    int* freq = calloc(100, sizeof(int));

    for (int i = 0; i < arr_count; i++) {
        freq[arr[i]]++;
    }

    *result_count = 100;

    return freq;
```

Test case 0

Test case 1

Compiler Message

Success

Language: C

Open in editor

```c
char** split_string(char*);
int parse_int(char*);
//TEJA
//1DT23CA015
int* countingSort(int arr_count, int* arr, int* result_count) {
    int* freq = calloc(100, sizeof(int));

    for (int i = 0; i < arr_count; i++) {
        freq[arr[i]]++;
    }

    int* sorted = malloc(arr_count * sizeof(int));
    int index = 0;
    for (int i = 0; i < 100; i++) {
        for (int j = 0; j < freq[i]; j++) {
            sorted[index++] = i;
```

Test case 0

Test case 1

Test case 2

Compiler Message

Success

Input (stdin)

Download

```
100
```

Submitted Code

Language: C

Open in editor

```c
    int n;
    scanf("%d",&n);
    char **s=malloc(n*sizeof(char *));
    char stemp[1000001];
    for(int i=0;i<n;i++){
        scanf("%s",stemp);
        s[i]=(char *)malloc((strlen(stemp)+1)*sizeof(char));
        strcpy(s[i],stemp);
    }
    qsort(s,n,sizeof(char *),cmp);
    for(int i=0;i<n;i++){
        printf("%s\n",s[i]);
    }
    return 0;
}

```

Test case 0

Test case 1

Compiler Message

Success

Submitted Code

Language: C

Open in editor

```
 * The function accepts 2D_STRING_ARRAY arr as parameter.
 */

void countSort(int arr_rows, int arr_columns, char*** arr) {
    // Find the maximum index to size the array
    int max_index = 0;
    for (int i = 0; i < arr_rows; i++) {
        int index = atoi(arr[i][0]);
        if (index > max_index) {
            max_index = index;
        }
    }

    // Create an array of string lists for each index
    char*** buckets = malloc((max_index + 1) * sizeof(char**));
```

Test case 0

Test case 1

Compiler Message

Success

```
//TEJA USN 15
long getWays(int n, int c_count, long* c) {
    long* dp = malloc((n + 1) * sizeof(long));
    dp[0] = 1;
    for (int i = 1; i <= n; i++) {
        dp[i] = 0;
    }


    for (int i = 0; i < c_count; i++) {
        long coin = c[i];
        for (int j = coin; j <= n; j++) {
            dp[j] += dp[j - coin];
        }
```

Submitted Code

Language: Java 15 Open in editor

```
12      BigInteger output = new BigInteger("0");
13
14      int n = input.nextInt();
15
16      if(n == 1){System.out.println(t1);}
17      if(n == 1){System.out.println(t2);}
18
19      for(int i = 2; i < n; i++)
20      {
21          output = t2.multiply(t2);
22          output = output.add(t1);
23
24          t1 = new BigInteger(t2.toString());
25
26          t2 = new BigInteger(output.toString());
27
```

Test case 0

Test case 1

Test case 2

Compiler Message

Success

Input (stdin) Download

Language: Java 15 Open in editor

```java
public class Solution {

    //declaring static variables which will be same for all the test cases
    static long M = 1000000007;
    static boolean[] f = new boolean[8200];
    static ArrayList<Integer> P= new ArrayList<Integer>();

    // Complete the primeXor function below.
    static int primeXor(int[] a, long[] cnt) {
        //to store the count of subsets
        long[][] dp = new long[1005][8200];

        dp[0][3500] = (cnt[0] + 1)/2;
        dp[0][0] = (cnt[0] + 2) / 2;

        for(int i=1; i<1005; i++){
```

Test case 0

Test case 1

Test case 2

Compiler Message

Success

Input (stdin) Download

1 1

Submitted Code

Language: C++20 Open in editor

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
const int MOD = 1000000007;
int n, ne;

int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    long long nv = 1;
    long long sd = 0;
    long long dtc = 0;
    long long diam = 0;
```

Test case 0

Test case 1

Compiler Message

Success

Submitted Code

Language: Python 3 Open in editor

```
import math
import os
import random
import re
import sys

# Complete the unboundedKnapsack function below.

T = int(input())  # Use input() in Python 3

def multisum(k, C):
    D = [False for _ in range(k+1)]   # Create a list of length k+1 which contain False
    D[0] = True
    for i in range(1, k+1):
        D[i] = any([D[i-c] for c in C if i - c >= 0])
```

Test case 0

Test case 1

Compiler Message

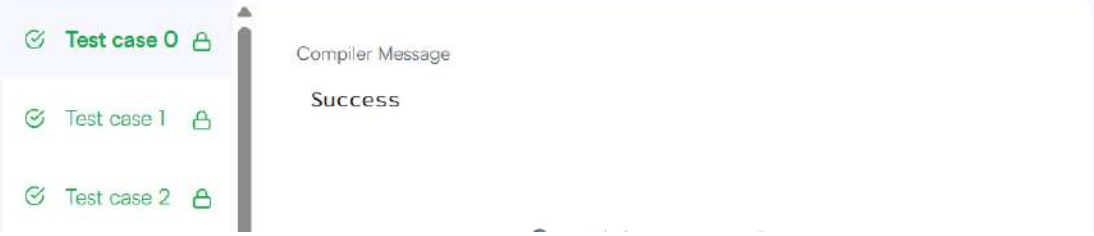
Success

Submitted Code

Language: Python 3 — Open in editor

```
def check(node):
    answer = True
    current = 0
    for i in range(N):  # use range()
        index = (node + i) % N
        current = min(C, current + fuel[index])
        current -= cost[index]
        if current < 0:
            answer = False
            break
    return answer

N, C = map(int, input().split())  # use input()
fuel = list(map(int, input().split()))  # list() needed in Python 3
cost = list(map(int, input().split()))
```



Submitted Code

Language: C

Open in editor

```c
#include <assert.h>
#include <limits.h>
#include <math.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define min(X,Y) ((X) > (Y) ? (Y) : (X))
#define f(i,a,b) for(int i = (a); i <= (b); i++)

typedef long long ll;

const ll MOD = 1000000007;
int N, K;
```

Test case 0

Test case 1

Compiler Message

Success

Input (stdin)

Download

Submitted Code

Language: C

Open in editor

```
16
17 int parse_int(char*);
18
19 /*
20  * Complete the 'twoArrays' function below.
21  *
22  * The function is expected to return a STRING.
23  * The function accepts following parameters:
24  *  1. INTEGER k
25  *  2. INTEGER_ARRAY A
26  *  3. INTEGER_ARRAY B
27  */
28
29 /*
30  * To return the string from the function, you should either do static allocation or
    dynamic allocation
```

Test case 0

Test case 1

Compiler Message

Success

Submitted Code

Language: C++20 Open in editor

```
    adj_mat adj(n);
    for (int i = 0, a,b; i < m; ++i) {
        std::cin >> a >> b;
        --a; --b;
        adj[a].push_back(b);
        adj[b].push_back(a);
    }

    for (auto& v : adj) {
        std::sort(v.begin(), v.end(), [&adj](const int& a, const int& b){
            return adj[a].size() < adj[b].size();
        });
    }

    std::vector<bool> visited(adj.size(), false);
    std::vector<int> order;
```

Submitted Code

Language: Python 3 — Open in editor

```
17    def __init__(self, regex_string):
18        RegexGraphNFA.node_count = 0
19        self.regex_string = regex_string
20        nfa_graph = self.translate_regex()
21        translate_graph = TranslateGraph(nfa_graph)
22        self.dfa_graph = translate_graph.translate()
23
24    def calculate(self, string_length):
25        return self.dfa_graph.count_paths(string_length)
26
27    def translate_regex(self, index=0):
28        result_set = ResultSet()
29        while index < len(self.regex_string):
30            if self.regex_string[index] == '(':
31                out_list, index = self.translate_regex(index + 1)
32                result  set.insert(out  list)
```

Test case 0

Test case 1

Compiler Message

Success

Submitted Code

Language: Python 3 Open in editor

```
            zip_longest(line, islice(line, k, None),
                        fillvalue=-1)])
        k <<= 1
    return line

def inverse_array(l):
    n = len(l)
    ans = [0] * n
    for i in range(n):
        ans[l[i]] = i
    return ans


def to_int_keys_best(l):
    seen = set()
```

Test case 0

Test case 1

Compiler Message

Success