1. **Nested loops**

You are given a **n** number of rows, print a pyramid made of stars like the one below:

Output –1 when user inputs an integer bigger than 100 and less than 1.

Input example #1 5	Output example #1 * *** ***** ******* *********
Input example #2 0	Output example #2 -1
Input example #3 101	Output example #3 -1
Input example #4 -5	Output example #4 -1
Input example #5 1	Output example #5 *

Code:

```cpp
#include <iostream>

using namespace std;

int main() {
  int n;

  cin >> n;

  if(n <= 0){
    cout << "-1" << endl;
  } else if(n > 100){
          cout << "-1" << endl;
  }
  else {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++) {
            cout << " ";
        }

    for (int j = 1; j <= 2 * i - 1; j++) {
        cout << "*";
    }
        cout << endl;
    }
  }

  return 0;
}
```

---

2. **Strings, use of char functions.**

You are given a string as input. Find the most frequent character (excluding spaces and special characters) in the string. If multiple characters have the same frequency, print any of them.

Input:

A string from the user.

Output:

Print the most frequent character and its frequency.

Input example #1 HELLO WORLD	Output example #1 l 3
Input example #2 !()*$^#%@	Output example #2 -1
Input example #3 "" (empty string)	Output example #3 -1
Input example #4 sal@m !as$f	Output example #4 a 2
Input example #5 asdfghjkl	Output example #5 a 1

Code:

```cpp
#include <bits/stdc++.h>

using namespace std;

int main() {
  string str;
  int charCount[26] = {0};
  char character = ' ';
  int maxCount = 0;

  getline(cin, str);

  for (int i = 0; i < str.length(); i++) {

    if (isalpha(tolower(str[i]))) {
      int charIndex = tolower(str[i]) - 'a';
      charCount[charIndex]++;
    }
  }

  for (int i = 0; i < 26; i++) {
    if (charCount[i] > maxCount) {
      character = 'a' + i;
      maxCount = charCount[i];
    }
  }

  if (character == ' ') {
    cout << "-1" << endl;
  } else {
    cout << maxCount << " " << character << endl;
  }

  return 0;
}
```

---

3. **Strings, use of string functions.**

You are given a string as input. Check if it's a palindrome. A palindrome is a word or phrase that reads the same backward as forward ("madam", "level" etc.).

Input:

Any string

Output:

Print “YES” if the string is palindrome, otherwise print “NO”.

Input example #1 mom	**Output example #1** YES
Input example #2 Hello World	**Output example #2** NO
Input example #3 a	**Output example #3** YES
Input example #4 leveL	**Output example #4** YES
Input example #5 123.321	**Output example #5** YES

Code:

```cpp
#include <bits/stdc++.h>

using namespace std;

bool isPalindrome(const string& str) {
  int len = str.length();

  if (len <= 1) {
    return true;
  }

  string lowerStr;
  for (int i = 0; i < len; i++) {
    lowerStr.push_back(tolower(str[i]));
  }

  string reversedStr(lowerStr);

  reverse(reversedStr.begin(), reversedStr.end());

  return (lowerStr == reversedStr);
}

int main() {
  string str;

  getline(cin, str);

  if (isPalindrome(str)) {
    cout << "YES" << endl;
  } else {
    cout << "NO" << endl;
  }

  return 0;
}
```

4. **Arrays**

You are given a two dimensional array. Calculate the sum of the elements present in the four corners of a two-dimensional array.

Input:

n number of columns.

Output:

Sum of the elements in the corners.

Input example #1 **3** 1 2 3 4 5 6	**Output example #1** 14
Input example #2 1 1 2	**Output example #2** 3
Input example #3 0	**Output example #3** -1
Input example #4 4 -1 2 3 4 5 6 -7 8	**Output example #4** 16
Input example #5 5 1 1 1 1 1 1 1 1 1 1	**Output example #5** 4

5. **Vectors**

You are given two vectors of integers. You need to find the intersection of these two vectors, the elements that are common to both vectors. Output the elements in the intersection. Output "NO INTERSECTION" if there is no intersection element.

Input:

The number of elements in the first vector. Elements of the first vector.

The number of elements in the second vector. Elements of the second vector.

Output:

Elements in the intersection of two vectors.

Input example #1 5	**Output example #1** 3 4 5

1 2 3 4 5 4 3 4 5 6	
Input example #2 3 1 2 3 3 4 5 6	**Output example #2** NO INTERSECTION
Input example #3 4 1 2 3 4 4 1 2 3 4	**Output example #3** 1 2 3 4
Input example #4 0 0	**Output example #4** NO INTERSECTION
Input example #5 5 1 2 3 4 5 6 4 5 6 7 8 9	**Output example #5** 4 5

Code:

```cpp
#include <bits/stdc++.h>

using namespace std;

vector<int> intersection(vector<int>& nums1, vector<int>& nums2) {
    vector<int> result;

    for (int i = 0; i < nums1.size(); ++i) {
        int num = nums1[i];
        if (find(nums2.begin(), nums2.end(), num) != nums2.end()) {
            result.push_back(num);
        }
    }

    return result;
}

int main() {
    int n, m;
    cin >> n;
    vector<int> nums1(n);
    for (int i = 0; i < n; ++i) {
        cin >> nums1[i];
    }

    cin >> m;
    vector<int> nums2(m);
    for (int i = 0; i < m; ++i) {
        cin >> nums2[i];
    }

    vector<int> result = intersection(nums1, nums2);

    if (result.empty()) {
        cout << "NO INTERSECTION" << endl;
    } else{
        for (int i = 0; i < result.size(); i++) {
            int num = result[i];
            cout << num << " ";
        }
    }

    return 0;
}
```