1) Jolly Jumper

```
#include <bits/stdc++.h>
using namespace std;

bool is_jolly_jumper(vector<int> &arr, int n)
{
    int s1 = 0, s2 = 0;

    s1 = ((n + 1) * n) / 2;

    for (int i = 1; i <= n; i++)
    {
        int t = abs(arr[i] - arr[i - 1]);
        // cout << "\n i: " << t;
        s2 += t;
    }

    // cout << s2 << s1 << "\n";

    if (s1 == s2)
        return true;
    else
        return false;
}

int main()
{
    vector<int> arr;
    int n; // n -> number of elements
    cin >> n;

    for (int i = 0; i < n; i++)
    {
        int temp;
        cin >> temp;
        arr.push_back(temp);
    }

    if (is_jolly_jumper(arr, n - 1))
    {
        cout << "Sequence is Jolly Jumper!";
    }
```

```
    else
    {
        cout << "Sequence is not a Jolly Jumper!";
    }

    return 0;
}
```

2) Merge Array

```
#include <bits/stdc++.h>
using namespace std;

void merge_array(int arr1[], int arr2[], int res[], int s1, int s2)
{
    int i = 0, j = s2 - 1, k = 0;

    while (i < s1 && j >= 0)
    {
        if (arr1[i] < arr2[j])
        {
            res[k] = arr1[i];
            k++;
            i++;
        }

        else
        {
            res[k] = arr2[j];
            k++;
            j--;
        }
    }

    while (i < s1)
    {
        res[k] = arr1[i];
        k++;
        i++;
    }

    while (j >= 0)
```

```
    {
        res[k] = arr2[j];
        k++;
        j--;
    }
}

int main()
{
    int arr1[] = {10, 15, 20, 35};
    int arr2[] = {40, 34, 25, 5, 2};

    // output--> 2 5 10 15 25 20 34 35 40
    int n1 = sizeof(arr1) / sizeof(arr1[0]);
    int n2 = sizeof(arr2) / sizeof(arr2[0]);

    int res[n1 + n2];

    merge_array(arr1, arr2, res, n1, n2);

    cout << "Array after merging: " << endl;
    for (int i = 0; i < n1 + n2; i++)
        cout << res[i] << " ";

    return 0;
}
```

3) Write a program to accept a number and print unique pairs of numbers such that multiplication of the pair is given number

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;
    cin >> n;

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

```
            if (n % i == 0)
            {
                cout << i << " * " << n / i << " = " << n << endl;
            }
        }

        return 0;
    }
```

4) Number of Occurrences

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    vector<int> v{10, 10, 20, 30, 10};
    // 10, 10, 20, 30, 10
    // element to find occurrence: 10
    int n = 10;
    int count = 0;

    for (auto x : v)
    {
        if (x == n)
            count++;
    }

    cout << " Number of occurrences: " << count << endl;

    return 0;
}
```

5) Count Value

```
#include <bits/stdc++.h>
using namespace std;

// Input
```

```
// array elements: 10, 10, 20, 30, 10, 20
// Output:
// 10: 3
// 20: 2
// 30: 1

int main()
{
    vector<int> v{10, 10, 20, 30, 10, 20};
    map<int, int> mp;

    for (int i = 0; i < v.size(); i++)
    {
        mp[v[i]]++;
    }

    cout << " \nResult: \n";

    for (auto i : mp)
    {
        cout << i.first << ": " << i.second << endl;
    }

    return 0;
}
```

6) Remove Duplicates

```
#include <bits/stdc++.h>
using namespace std;

int result(int *v, int n)
{
    int i = 0;
    int x = n;
    if (x == 0)
        return 0;

    for (int j = 1; j < x; j++)
    {
```

```
            if (v[j] != v[i])
            {
                i++;
                v[i] = v[j];
            }
        }

        return i + 1;
    }

    int main()
    {
        int v[] = {10, 10, 20, 30, 10, 20, 40, 50, 40, 100, 100};

        int len = sizeof(v) / sizeof(v[0]);
        sort(v, v + len);
        int n = result(v, len);

        for (int i = 0; i < n; i++)
        {
            cout << v[i] << " ";
        }
        return 0;
    }
```

7) Palindrome

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    string s;
    getline(cin, s);
    // hananah

    int len = s.length(), j = len - 1, flag = 1;
    for (int i = 0; i < len / 2; i++)
    {
```

```
        if (s[i] != s[j])
        {
            cout << "Not a palindrome!";
            flag = 0;
            break;
        }
        else
        {
            j--;
        }
    }

    if (flag)
        cout << "Is a palindrome!";

    return 0;
}
```

8) Reverse Array

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int arr[] = {10, 20, 30, 40, 50, 60, 70, 80};
    int n = sizeof(arr) / sizeof(arr[0]);
    int j = n - 1;

    for (int i = 0; i < n / 2; i++)
    {
        int t = arr[i];
        arr[i] = arr[j];
        arr[j] = t;
        j--;
    }

    cout << "Printing reverse array: \n";

    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
```

```
    return 0;
}
```

9) String Operations

```
#include <bits/stdc++.h>
using namespace std;

void snake_case(string s)
{
    for (int i = 0; i < s.length(); i++)
    {
        if (s[i] == ' ')
            s[i] = '_';
        else
            s[i] = tolower(s[i]);
    }

    cout << "\nsnake_case: " << s;
}

void camelCase(string s)
{
    string res;
    for (int i = 0; i < s.length(); i++)
    {
        if (s[i] == ' ')
        {
            res += toupper(s[i + 1]);
            i++;
        }
        else
        {
            res += s[i];
        }
    }

    cout << "\ncamelCase: " << res;
}

void PascalCase(string s)
{
    string res;
```

```
    res += toupper(s[0]);
    for (int i = 1; i < s.length(); i++)
    {
        if (s[i] == ' ')
        {
            res += toupper(s[i + 1]);
            i++;
        }
        else
        {
            res += s[i];
        }
    }

    cout << "\nPascalCase: " << res;
}

int main()
{
    string s = "user login count";

    snake_case(s);
    camelCase(s);
    PascalCase(s);
    return 0;
}
```

10)Write a program to check whether a given string is palindrome or not ignoring character case, white spaces and punctuations.

For example: following are valid palindrome

A man a plan a canal panama

Was it a cat I saw?

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    string str;
```

```
    getline(cin, str);
    // Was it a cat I saw?

    int len = str.length(), flag = 1;

    string s;
    for (int i = 0; i < len; i++)
    {
        char ch = tolower(str[i]);
        if (ch == ' ' || isdigit(ch))
            continue;

        if (isalpha(ch))
            s += ch;
    }

    // cout << s << "\n";

    int j = s.length() - 1;

    for (int i = 0; i < (s.length()) / 2; i++)
    {

        if (s[i] != s[j])
        {
            cout << "Not a palindrome!";
            flag = 0;
            break;
        }
        else
        {
            j--;
        }
    }

    if (flag)
        cout << "Is a palindrome!";

    return 0;
}
```