1) LCS

```
#include <bits/stdc++.h>
using namespace std;
int findLCSOf3(string& s1, string& s2, string& s3, int n1, int n2, int n3) {

    // Base case: If any of the strings is empty
    if (n1 == 0 || n2 == 0 || n3 == 0)
        return 0;

    // If last characters of s1, s2, and s3 are the same
    if (s1[n1 - 1] == s2[n2 - 1] && s2[n2 - 1] == s3[n3 - 1])
        return 1 + findLCSOf3(s1, s2, s3, n1 - 1, n2 - 1, n3 - 1);

    // If last characters are not the same, calculate
    // LCS by excluding one string at a time
    return max({
                findLCSOf3(s1, s2, s3, n1 - 1, n2, n3),
                findLCSOf3(s1, s2, s3, n1, n2 - 1, n3),
                findLCSOf3(s1, s2, s3, n1, n2, n3 - 1)});
}

int lcsOf3(string& s1, string& s2, string& s3) {
    int n1 = s1.size();
    int n2 = s2.size();
    int n3 = s3.size();
    int res = findLCSOf3(s1, s2, s3, n1, n2, n3);
    return res;
}

int main() {
    string s1 = "AGGT12";
    string s2 = "12TXAYB";
    string s3 = "12XBA";
    int res = lcsOf3(s1, s2, s3);
    cout << res ;
    return 0;
}
```

2)Space Separted Strings

```
#include <iostream>

#include <sstream>

#include <map>
```

```
#include <vector>
#include <algorithm>
using namespace std;

void findRepeatedWords(const string& text) {
  map<string, int> wordCount;
  istringstream iss(text);
  string word;

  while (iss >> word) {
    wordCount[word]++;
  }

  vector<string> repeatedWords;
  for (const auto& entry : wordCount) {
    if (entry.second > 1) {
      repeatedWords.push_back(entry.first);
    }
  }

  if (repeatedWords.empty()) {
    cout << "NA";
  } else {
    sort(repeatedWords.begin(), repeatedWords.end());
    for (size_t i = 0; i < repeatedWords.size(); i++) {
      if (i > 0) cout << " ";
      cout << repeatedWords[i];
    }
  }
  cout << endl;
}
```

```
int main() {
    string text;
    getline(cin, text);
    findRepeatedWords(text);
    return 0;
}
```

3) Count of specific string

```
// Online C++ compiler to run C++ program online
#include <iostream>
#include<string>
using namespace std;

int main() {
    string str;
    int c=0;
    std::cout << "Try programiz.pro";
    getline(cin,str);
    for(char ch:str){
        if(ch=='a'){
            c++;
        }
    }
    std::cout << c;
    return 0;
}
```

4) Product IDs

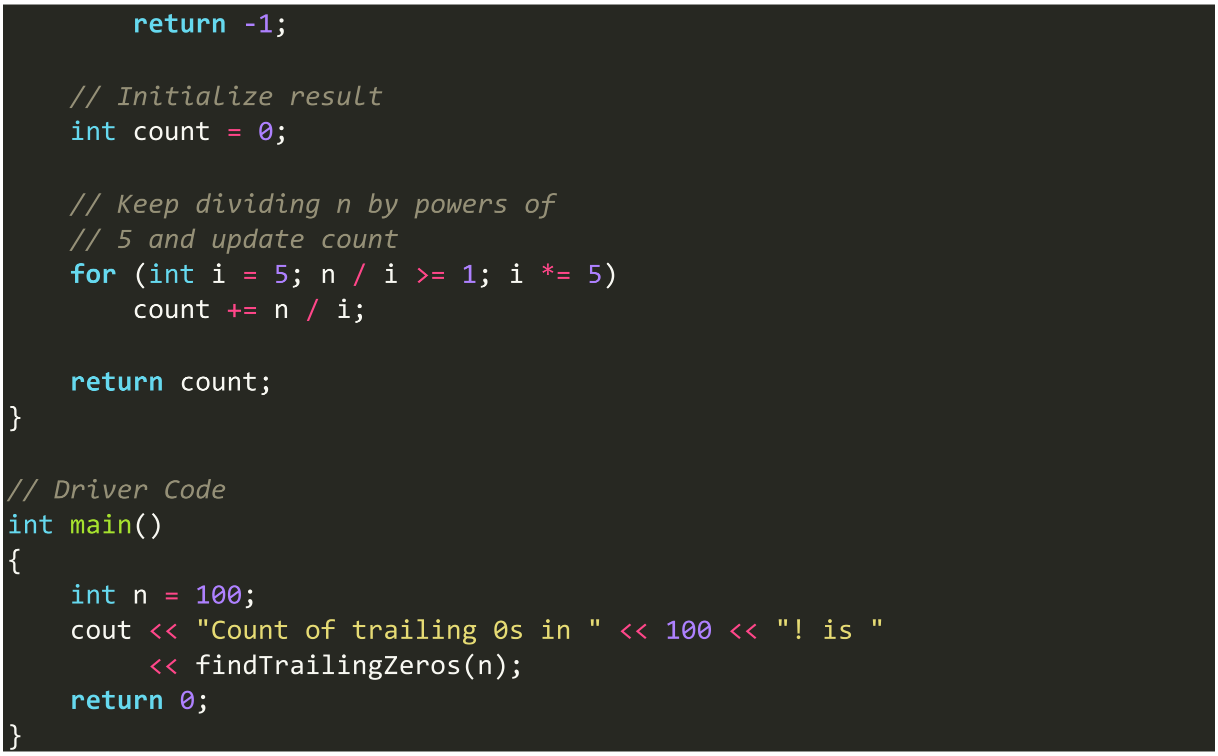
```
#include <iostream>
#include <string>
using namespace std;

int main(){
    string str;
    int v,k,l = 0;
    getline(cin,str);
    for(char c: str){
        if(c=='v'){
            v=1;
        }
        if(c=='k'){
            k=1;
        }
        if(c=='l'){
            l=1;
        }
    }
    cout<<k+v+l<<endl;
    return 0;
}
```

5) Count of trailing zeros in fact value

```
#include <iostream>
using namespace std;

// Function to return trailing 0s in factorial of n
int findTrailingZeros(int n)
{
      // Negative Number Edge Case
    if (n < 0)
```



```
        return -1;

    // Initialize result
    int count = 0;

    // Keep dividing n by powers of
    // 5 and update count
    for (int i = 5; n / i >= 1; i *= 5)
        count += n / i;

    return count;
}

// Driver Code
int main()
{
    int n = 100;
    cout << "Count of trailing 0s in " << 100 << "! is "
         << findTrailingZeros(n);
    return 0;
}
```

6) Decimal to Binary

```
#include <iostream>

#include <bits/stdc++.h>

using namespace std;


int main(){

   vector<int> arr;

   int num,i=0,n=0;

   cin>>num;

   while(num>0){

      arr.push_back(num%2);

      num=num/2;

      n=n+1;

   }
```

```
    reverse(arr.begin(),arr.end());
    for(i=0;i<n;i++){
        cout<<arr[i];
    }

    return 0;
}
```

7) Unsold Products

```
#include <iostream>
#include <vector>
using namespace std;


int main() {
    vector<int> A = {5, 2, 0, 8, 0, 2, 1};
    vector<int> sold, unsold;
    for (int num : A) {
        if (num == 0) {
            unsold.push_back(num);
        } else {
            sold.push_back(num);
        }
    }
    sold.insert(sold.end(), unsold.begin(), unsold.end());
    for (int num : sold) {
        cout << num << " ";
    }
    cout << endl;
    return 0;
}
```

8) Alice and String

```
#include <iostream>
#include <string>
using namespace std;

int longestUninterruptedSubstring(string S)
{
    int maxl = 0;
    int current = 0;
    for (char c : S)
    {
        if (c == '.')
        {
            current = 0;
        }
        else
        {
            current++;
            maxl = max(maxl, current);
        }
    }
    return maxl;
}

int main()
{
    string S;
    cout << "Enter the string: ";
    cin >> S;
    int result = longestUninterruptedSubstring(S);
    cout << "Length of the longest uninterrupted substring: " << result << endl;
    return 0;
}
```

9) File version

```
#include <iostream>
#include <vector>
#include <string>
#include <regex>
using namespace std;

int findLatestVersion(vector<string>& S) {
    int latestVersion = -1;
    regex pattern("File_\\d+");

    for (const string& file : S) {
        if (regex_match(file, pattern)) {
            int version = stoi(file.substr(5));
            latestVersion = max(latestVersion, version);
        }
    }
    return latestVersion;
}

int main() {
    int n;
    cout << "Enter the number of files: ";
    cin >> n;
    cin.ignore(); // To clear newline character from input buffer

    vector<string> S(n);
    cout << "Enter file names: \n";
    for (int i = 0; i < n; i++) {
        getline(cin, S[i]);
    }

    cout << findLatestVersion(S) << endl;
    return 0;
}
```

10) Even Odd

```
#include <iostream>
using namespace std;
int main()
{
    string s="";
    int n;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    for(int i=0;i<n;i++){
        if(a[i]%2==0){
            s=s+"Even";
        }
        else {
            s=s+"Odd";
        }
    }
    cout<<s;
    return 0;
}
```

11) Magical Number

```
#include <iostream>
#include <string>
using namespace std;
int bsum(int n) {
    string b = "";
    while (n > 0) {
        b=(n%2==0?"0":"1")+b;
        n/=2;
    }
    int sum=0;
    for(char c : b) {
        if(c=='0')
            sum+=1;
        else if(c=='1')
            sum+=2;
    }
    return sum;
}
int count(int N) {
    int m = 0;
    for(int i=1;i<=N;i++) {
        int sum=bsum(i);
        if(sum%2 != 0)
            m++;
    }
    return m;
}
int main() {
    int N;
    cin >> N;
    cout <<count(N);
    return 0;
}
```

12) Rearrangement of Bits

```
#include <iostream>
#include <string>
#include <math.h>
using namespace std;

int main(){
    int n,i=0,m=0,ans=0, j=0;
    int arr[32],ar[32];
    cin>>n;

    if (n == 0) {
        cout << "Binary: 0" << endl;
        return 0;
    }
    while (n > 0) {
        arr[i] = n % 2;
        n = n / 2;
        i++;
        m++;
    }


    for(i=0;i<m;i++){
        if(arr[i]==1){
        ar[j]=arr[i];
        j++;
        }
    }


    for(i=0;i<j;i++){
        ans = ans+ pow(2,i);
    }
    cout<<ans<<endl;

    return 0;
}
```

13) Minimum Badness

```
#include <iostream>
#include <string>
#include <math.h>
using namespace std;
```

```
int main(){
    string str;
    int count=0;
    cin>>str;
    int l;
    l=str.length();

    for(int i=0;i<l;i++){
        if(str[i]=='w'){
            if(str[i-1]=='r' || str[i+1] == 'r'){
                str[i]='r';
            }
            else{
                str[i]='b';
            }
        }
    }
    for(int i=0;i<l-1;i++){
        if(str[i]!=str[i+1]){
            count++;
        }
    }
    cout<<str<<endl;
    cout<<count;
    return 0;
}
```

14) Number Of bids

```
#include <iostream>
```

```
using namespace std;
int main()
{
    int n,c=0;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    int t;
    cin>>t;
    for(int i=0;i<n;i++){
        for(int j=i;j<n;j++){
            if(abs(a[i]-a[j])==t){
                cout<<"["<<a[i]<<","<<a[j]<<"]"<<"\n";
                c++;
            }
        }
    }
    cout<<c;
    return 0;
}
```