1. Program to Check Odd or Even Using the Ternary Operator

```
#include <iostream>
using namespace std;
int main()
{
    int num;
    cout << "Enter an integer: ";
    cin >> num;
    (num % 2 == 0) ? cout << num << " is even." : cout << num <<" is odd.";

    return 0;
}
```

2. Program to Check Vowel or Consonant or any other

```
#include <iostream>
using namespace std;
int main(void)
{
    char c;
    cout << "Enter an alphabet: ";
    cin >> c;

    int lowercase_vowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
    int uppercase_vowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');

    // Show error message if c is not an alphabet
    if (!isalpha(c))
        cout << "Error! Non-alphabetic character.";
    else if (lowercase_vowel || uppercase_vowel)
        cout << c << " is a vowel.";
    else
        cout << c << " is a consonant.";
}
```

3. Program to Check Leap Year

```
#include <iostream>
using namespace std;
int main(void) {
   int year;
   cout << "Enter a year: ";
   cin>> year;
   if (year % 400 == 0) {
      cout << year << " is a leap year.";
   }
   else if (year % 100 == 0) {
      cout << year << " is not a leap year.";
   }
   else if (year % 4 == 0) {
      cout << year << " is a leap year.";
   }
   else {
      cout << year << " is not a leap year.";
   }
}
```

4. Notes Demonstration

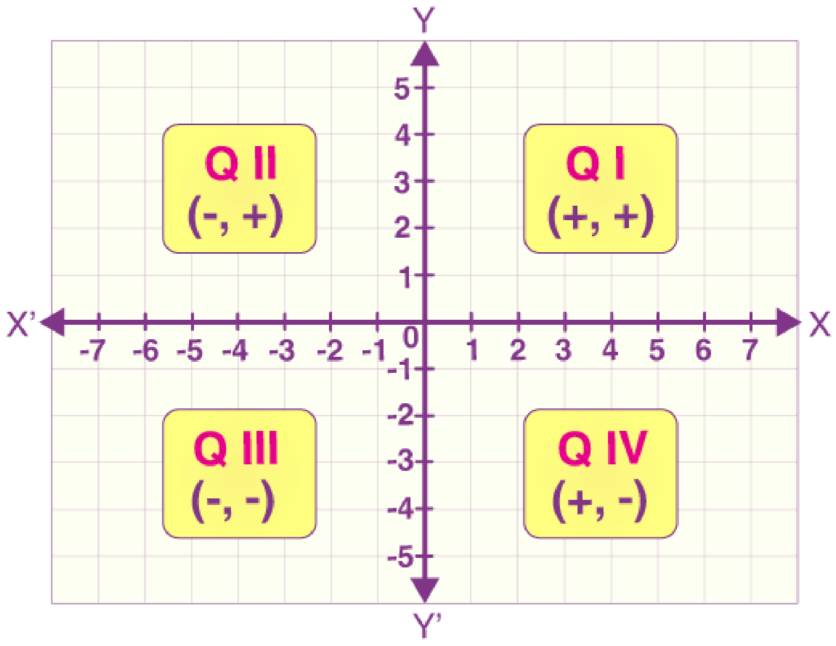
```
#include <iostream>
using namespace std;
int main(void)
{
    int amount;
    int note500=0, note100=0, note50=0, note20=0, note10=0, note5=0, note2=0,
note1=0;
    cin >> amount;

    if(amount >= 500)
    {
        note500 = amount/500;
        amount -= note500 * 500;
    }
    if(amount >= 100)
    {
        note100 = amount/100;
        amount -= note100 * 100;
    }
    if(amount >= 50)
    {
        note50 = amount/50;
        amount -= note50 * 50;
    }
    if(amount >= 20)
    {
        note20 = amount/20;
        amount -= note20 * 20;
    }
    if(amount >= 10)
    {
        note10 = amount/10;
        amount -= note10 * 10;
    }
    if(amount >= 5)
    {
        note5 = amount/5;
        amount -= note5 * 5;
    }
    if(amount >= 2)
    {
        note2 = amount /2;
        amount -= note2 * 2;
    }
    if(amount >= 1)
    {
        note1 = amount;
    }

    cout << "Notes of 500 - " << note500;
    cout << "\nNotes of 100 - " << note100;
    cout << "\nNotes of 50 - " << note50;
    cout << "\nNotes of 20 - " << note20;
    cout << "\nNotes of 10 - " << note10;
    cout << "\nNotes of 5 - " << note5;
    cout << "\nNotes of 2 - " << note2;
    cout << "\nNotes of 1 - " << note1;
}
```

5. Problem statement
Check Quadrant

Check the quadrant graph below for a better understanding.



```
#include <stdio.h>
void main()
{
        int x,y;
        printf("Enter X corresponds - ");
        scanf("%d",&x);
        printf("Enter Y corresponds - ");
        scanf("%d",&y);
        if(x > 0 && y > 0)
                printf("Ist Quadrant");
        else if(x < 0 && y > 0)
                printf("IInd Quadrant");
        else if(x < 0 && y < 0)
                printf("IIIrd Quadrant");
        else if(x > 0 && y < 0)
                printf("IVth Quadrant");
        else
                printf("Origin");
}
```

6. Problem statement
"VideoShare" is an online video-sharing platform. The company has decided to rate its user's channels based on the sum total of the number of views received online and the subscribers. This sum total is referred to as User Points. The rating will be given according to the below charts:

User Points	Rating
30 - 50	Average
51 - 60	Good
61 - 80	Excellent
81 - 100	Outstanding

The whole process is automated and is carried out by the company's system.
Write an algorithm to find the rating of the given user's channel based on the user points.

Example
Input
77
Output
Excellent
Explanation
77 lies in the range 61-80, so the rating is 'Excellent'.

Solution1: with if else

```
#include<iostream>
using namespace std;
void fun(int n)
{
        if(n >= 30 && n <= 50)
                cout << "Average";
        else if(n >= 51 && n <= 60)
                cout << "Good";
        else if(n >= 61 && n <= 80)
                cout << "Excellent";
        else if(n >= 81 && n <= 100)
                cout << "Outstanding";
}
int main()
{
        int n;
        cin >> n;
        fun(n);
        return 0;
}
```

Solution2: with switch case

```
#include <iostream>
using namespace std;
int main(void)
{
    int n;
    cin >> n;

    switch(n)
    {
        case 30 ... 50 :
            cout << "Average";
            break;
        case 51 ... 60 :
            cout << "Good";
            break;
        case 61 ... 80 :
            cout << "Excellent";
            break;
        case 81 ... 100 :
            cout << "Outstanding";
            break;
    }
}
```

7. Problem statement

The online math course provided 'MathAtTip' has designed a course for children called Learning Number Recognition and Counting. The assessment part of the course has a question where the
student is given a number and a digit. The student needs to find out the total count of the digits present in the number excluding the given digit.

Write an algorithm to help the student find out the count of the total number of digits present in the number excluding the given digit.

Example

Input

5644456 5

Output

5

Explanation

Excluding 5: the digits in the numbers are 4 and 6 and their total count is 5. Hence the output is 5.

Solution:

```
#include<iostream>
using namespace std;
int excludingDigit(int,int);
int main()
{
        int num,n;
        cin >> num;
        cin >> n;
        cout << excludingDigit(num,n);
        return 0;
}
int excludingDigit(int num,int n)
{
        int digit,count=0;
        while(num)
        {
                digit=num%10;
                num=num/10;
                if(digit!=n)
                {
                        count++;
                }
        }
        return count;
}
```

8. Problem statement

The new bank, "YoursPay", has a list of N customers' bank account balances. The list consists of both positive and negative balances. The positive balance signifies the current year's customers and the negative balance signifies last year's customers. The bank has decided to offer shortlisted customers credit scores to their credit cards. The credit score will be the sum of the two balances from the list with the smallest product when multiplied. If the credit score is positive then the credit will be provided to the current year's customer, otherwise, it will go to the last year's customer.

Write an algorithm to find the credit score.

Input Format

The first line of input consists of an integer - numCustomers, representing the number of banking customers (N).

The second line of input consists of N space-separated integers - balance0, balance1, ... balanceN-1 representing the customers' bank balances.

Output Format

Print an integer representing the credit score.

Sample Input	**Sample Output**
5	3
1 8 -5 7 5	

Solution:

```
#include<iostream>
using namespace std;
int fun(int arr[], int n)
{
        int i, min = 99999, max = -999999;
        for(i = 0; i < n; i++)
        {
                if(arr[i] > max)
                max = arr[i];
                if(arr[i] < min)
                min = arr[i];
        }
        return min+max;
}
int main()
{
        int n, i;
        cin >> n;
        int arr[n];
        for(i = 0; i < n; i++)
                cin >> arr[i];
        cout << fun(arr, n);
        return 0;
}
```

9. Problem statement

The games development company "FunGames" has developed a balloon shooter game. The balloons are arranged in a linear sequence and each balloon has a number associated with it. The numbers on the balloons are in the fibonacci series.
In the game, the player shoots 'k' balloons. The player's score is the sum of numbers on the 'k' balloons.
Write an algorithm to generate the player's score.

Example
Input
7
Output
20
Explanation
The fibonacci sum is 0+1+1+2+3+5+8=20

Solution:

```
#include<iostream>
using namespace std;
int fibo(int n)
{
        int fibs[n];
        fibs[0] = 0;
        fibs[1] = 1;
        int sum=0;
        for(int i=2;i<n;i++)
        {
                fibs[i] = fibs[i-1]+fibs[i-2];
        }
        for(int i=0;i<n;i++)
        {
                sum += fibs[i];
        }
        return sum;
}
int main()
{
        int numBalloons;
        cin>>numBalloons;
        int result = fibo(numBalloons);
        cout<<result;
        return 0;
}
```

10. Problem statement

The IT giant "SoftCompInfo" has decided to transfer its message through the network using a new encryption technique. The company has decided to encrypt the data using the non-prime number concept. The message is in the form of a number and the sum of non-prime digits present in the message is used as the encryption key.

Write an algorithm to determine the encryption key.

Example

Input

45673

Output

10

Explanation

The non-prime digits are 4 and 6. Hence the output is 4+6 = 10.

Solution:

```
#include <iostream>
using namespace std;
int nonprime(int);
int nonprime(int n)
{
        int digit,i,sum=0,m;
        while(n!=0)
        {
                digit=n%10;
                n=n/10;
                for(i=2;i<digit;i++)
                {
                        if(digit%i==0)
                        {
                                sum=sum+digit;
                                break;
                        }
                }
        }
        return sum;
}
int main()
{
        int n;
        cin>>n;
        cout<<nonprime(n);
        return 0;
}
```

11. Problem statement

The e-commerce company "TodaysApparel" has a list of sale values of N days. Some days the company made a profit, represented as a positive sales value. Other days the company incurred a loss, represented as a negative sales value. The company wishes to know the number of profitable days on the list.

Write an algorithm to help the company know the number of profitable days in the list.

Example

Input

```
7
23 -7 13 -34 56 43 -12
```

Output

```
4
```

Explanation

The number of positive sales values in the list is 4. Hence the output is 4.

Solution:

```
#include<iostream>
using namespace std;
int fun(int arr[], int n)
{
        int i, count = 0;
        for(i = 0; i < n; i++)
        {
                if(arr[i] > 0)
                count++;
        }
        return count;
}
int main()
{
        int n, i;
        cin >> n;
        int arr[n];
        for(i = 0; i < n; i++)
        {
                cin >> arr[i];
        }
        cout << fun(arr, n);
        return 0;
}
```

12. Problem Statement

The cosmetic company "BeautifyMe" wishes to know the alphabetic product code from the product barcode. The barcode of the product is a numeric value and the alphabetic product is a string value tagged 'a-j'. The alphabetic range 'a-j' represents the numeric range '0-9'. To produce the alphabetic product code, each digit in the numeric barcode is replaced by the corresponding matching letters.

Write an algorithm to display the alphabetic product code from the numeric barcode.

Input Format

The input consists of an integer - barcode, that represents the barcode of the product.

Output Format

Print a string representing the alphabetic product code.

Sample Input	**Sample Output**
12403	bcead

Solution:

```
#include<iostream>
#include <string>
using namespace std;
string myMethod(int barcode,char alphabets[])
{
        string result = "";
        while(barcode>0)
        {
                int rem = (int) (barcode%10);
                result = alphabets[rem] + result;
                barcode = barcode/10;
        }
        return result;
}
int main()
{
        int barcode;
        cin>>barcode;
        char alphabets[] = {'a','b','c','d','e','f','g','h','i','j'};
        string res=myMethod(barcode,alphabets);
        cout<<res;
        return 0;
}
```

13. Problem statement

The children's toy-making company "ToysFun" is building cubic-shaped learning toys. The company has a list of N dimensions suggested by its designers but they wish to choose only those dimensions for the toys that are perfect cube numbers. To do this, they need to know the total count of perfect cube numbers present in the list of dimensions.

Write an algorithm to help the toy manufacturers find the total count of perfect cube numbers present in the list of dimensions.

Example

Input

9

23 1 8 56 27 67 64 125 232

Output

5

Explanation

The cube numbers are 1, 8, 27, 64, 125. Hence the output is 5.

Solution:

```
#include<iostream>
using namespace std;
bool dimens_is_cube(int n) {
        int i=1;
        bool result = false;
        while(i<=n) {
                if(i*i*i == n) {
                        result = true;
                        break;
                }
                i++;
        }
        return result;
}
int main()
{
        int numDimensions;
        cin>>numDimensions;
        int dimens[numDimensions];
        for(int i=0;i<numDimensions;i++) {
                cin>>dimens[i];
        }
        int perfect_cube = 0;
        for(int i=0;i<numDimensions;i++) {
                if(dimens_is_cube(dimens[i])) // if (dimens[i] %3 ==0) {
                        perfect_cube++;
                }
        }
        cout<<perfect_cube;
}
```

14. Problem statement

"Perfect Math" is an online math program. In one of the assignments the system displays a list of N numbers and a value K, and students need to calculate the sum of remainders after dividing all the numbers from the list of N numbers by K. The system needs to develop a program to calculate the correct answer for the assignment.
Write an algorithm to calculate the correct answer for the assignment.

Example
Input

```
5
4
25 26 54 81 48
```

Output

```
6
```

Explanation

The calculation is as follows:
25 % 4 = 1
26 % 4 = 2
54 % 4 = 2
81 % 4 = 1
48 % 4 = 0
Sum = 1 + 2 + 2 + 1 = 6.
Hence the output is 6.

Solution:

```
#include<iostream>
using namespace std;
int fun(int arr[], int n, int value)
{
        int i, sum = 0;
        for(i = 0; i < n; i++)
        {
                sum = sum + (arr[i] % value);
        }
        return sum;
}
int main()
{
        int n, i, value;
        cin >> n >> value;
        int arr[n];
        for(i = 0; i < n; i++)
        {
                cin >> arr[i];
        }
        cout << fun(arr, n, value);
        return 0;
}
```

15. Problem statement

In an online exam, the test paper set is categorized by the letters A-Z. The students enrolled in the exam have been assigned a numeric value called application ID. To assign the test set to the student, firstly the sum of all digits in the application ID is calculated. If the sum is within the numeric range 1-26 the corresponding alphabetic set code is assigned to the student, else the sum of the digits are calculated again and so on until the sum falls within the 1-26 range.

Write an algorithm to display the examination set code according to the student application ID.

Input

6442

Output

P

Explanation

The sum of the digits of the application ID is 6 + 4 + 4 + 2 = 16.

The letter that corresponds to 16 is 'P'. Hence the output is P.

Solution:

```
#include<iostream>
using namespace std;
char fun(int n)
{
        int rem, s = 0;
        while(n != 0 || s > 26)
        {
                if(s > 26)
                {
                        n = s;
                        s = 0;
                }
                rem = n % 10;
                s = s + rem;
                n = n / 10;
        }
        return (char)(s+64);
}
int main()
{
        int n;
        cin >> n;
        cout << fun(n);
        return 0;
}
```

16. Problem statement

Developers at the mobile company 'TalkFree' have designed a game for the launch of their new product. In the game, the player is given a number. The player has to find out the difference between the number and the reverse of the number. The difference between the two numbers is the player's score. The number given to the player and the player's score can either be a negative or positive
number.

Write an algorithm to find the player score.

Example

Input

1234

Output

-3087

Explanation

The reverse of 1234 is 4321.

Difference is: 1234 - 4321 = -3087

Hence the output is -3087.

Solution:

```
#include<iostream>
using namespace std;
long int reverse(long int n)
{
        long int rev=0,rem,temp;
        temp=n;
        while(n)
        {
                rem=n%10;
                rev=rev*10+rem;
                n/=10;
        }
        return temp-rev;
}
int main()
{
        long int n;
        cin>>n;
        cout<<reverse(n);
        return 0;
}
```

17. Problem statement
The warehouse of an e-commerce company has limited stock available for each item due to a promotional sale. The company needs to distribute this stock to its distribution center in a particular manner. If the value of the available stock for an item is an even number, then the total stock is divided by 2. If it is an odd number, 1 is subtracted from it. The company wishes to know how many distributions can take place before the stock reaches 0.
Write an algorithm to calculate the number of distributions that can take place before the stock reaches 0.

Example
Input
12
Output
5
Explanation
12 is even, 12/2 is 6
6 is even, 6/2 is 3
3 is odd, 3-1 is 2
2 is even, 2/2 is 1
1 is odd, 1-1 is 0
The total number of distributions that take place is 5. Hence the output is 5.

Solution:

```
#include<iostream>
using namespace std;
int fun(int n)
{
        int count = 0;
        while(n != 0)
        {
                if(n % 2 == 0)
                {
                        n = n / 2;
                        count++;
                }
                else
                {
                        n = n - 1;
                        count++;
                }
        }
        return count;
}
int main()
{
        int n, count = 0;
        cin >> n;
        cout << fun(n);
        return 0;
}
```