

GETTING STARTED

1.print z

```
#include<iostream>
using namespace std;
int main(int argc, char** agrc){

    //write your code here
    cout<<"*****"<<endl;
    cout<<"    *   "<<endl;
    cout<<"  *    "<<endl;
    cout<<" *     "<<endl;
    cout<<"*****"<<endl;
    return 0;
}
```

2.grading system

```
#include<iostream>
using namespace std;
void gradingSystem(int m)
{
    // write your code here
    if(m>90){
        cout<<"excellent";
    }
    else if ((m>80)&&(m<=90))
    {
        cout<<"good";
    }
    else if ((m>70)&&(m<=80))
```

```

    {
        cout<<"fair";
    }
    else if ((m>60)&&(m<=70))
    {
        cout<<"meets expectations";
    }
    else if (m<=60)
    {
        cout<<"below par";
    }
}

```

```

int main()
{
    int marks;
    cin>>marks;
    gradingSystem(marks);
    return 0;
}

```

3.Is A Number Prime Easy

1. You've to check whether a given number is prime or not.
2. Take a number "t" as input representing count of input numbers to be tested.
3. Take a number "n" as input "t" number of times.
4. For each input value of n, print "prime" if the number is prime and "not prime" otherwise.

```
#include <iostream>
#include <cmath>
using namespace std;
int main(int argc, char **argv){
    int t;
    cin >> t;

    //write your code here
    // int w{0};
    // if(t<=1){
    //     cout<<"not prime"<<endl;
    // }

    // for(int j{2};j<=sqrt(t);j++){
    //     if(t%j == 0){
    //         w++;
    //     }
    // }
    // if(w>0){
    //     cout<<"prime"<<endl;
    // }else{
    //     cout<<"not prime"<<endl;
    // }

    int n{0};
    for(int i{0};i<t;i++){
        cin >> n;
        int k{0};

        for(int j{2};j<=sqrt(n);j++){
            if(n%j == 0){
                k++;
                break;
            }
        }
    }
}
```

```

    }
}
if(k==1){
    cout<<"not prime"<<endl;
}else{
    cout<<"prime"<<endl;
}

```

Example

Sample Input

```

5
13
2
3
4
5

```

Sample Output

```

prime
prime
prime
not prime
prime

```

```

}

```

```

}

```

4. Print All Primes Till N

Easy

1. You've to print all prime numbers between a range.
2. Take as input "low", the lower limit of range.
3. Take as input "high", the higher limit of range.
4. For the range print all the primes numbers between low and high (both included).

Constraints

$2 \leq \text{low} < \text{high} < 10^6$

```
#include <iostream>
#include <cmath>

using namespace std;
int main(int argc, char **argv){
    int low, high;
    cin >> low >> high;

    //write your code here
    for(int i{low}; i<=high; i++){
        int count{0};
        for(int j{2}; j<=sqrt(i); j++){
            if(i%j==0){
                count++;
                break;
            }
        }
        if(count==0){
            cout<<i<<endl;
        }
    }
}
```

```
}    return 0;
```

Example

Sample Input

```
6
24
```

Sample Output

```
7
11
13
17
19
23
```

5. Print Fibonacci Numbers Till N
Easy

1. You've to print first n fibonacci numbers.
2. Take as input "n", the count of fibonacci numbers to print.
3. Print first n fibonacci numbers.

Constraints

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

```
int main(int argc, char **argv)
{
    int n;
    cin >> n;

    //write your code here
    // if(n==1){
    //     int a{0};
    //     cout<<a<<endl;
    // }else if(n==2){
    //     int a{0};
    //     int b{1};
    //     cout<<a<<endl<<b<<endl;
    // }else{
    //     int a{0};
    //     int b{1};
    //     int c{a+b};
    //     cout<<a<<endl<<b<<endl;
    //     for(int i{2};i<n;i++){
    //         cout<<c<<endl;
    //         a=b;
    //         b=c;
    //         c=a+b;
    //     }
}
```

```
// }
```

```
int a{0};  
int b{1};  
int c{0};  
cout<<a<<endl;  
for(int i{1};i<n;i++){  
    cout<<b<<endl;  
    c = a+b;  
    a=b;  
    b=c;  
}
```

```
return 0;
```

```
}
```

Example

Sample Input

```
10
```

Sample Output

```
0  
1  
1  
2  
3  
5  
8  
13  
21  
34
```


6.Count Digits In A Number

Easy

1. You've to count the number of digits in a number.
2. Take as input "n", the number for which the digits has to be counted.
3. Print the digits in that number.

Constraints

$1 \leq n < 10^9$

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

```
int main(int argc, char **argv){
    // int n;
    // cin >> n;

    //write your code here
```

```

int n{0};
int d{0};
int deno{10};
int ratio{0};

//  cout<<"Enter the number you wants to
count digits of: ";
cin>>n;

do{
d++;
ratio =(n/deno);
deno *=10;
}while(ratio>=1);

//  cout<<"Number of digits in "<<n<<"
is "<<d<<endl;
cout<<d<<endl;

return 0;
}

```

Example

Sample Input

65784383

Sample Output

8

7.Digits Of A Number

Easy

1. You've to display the digits of a number.
2. Take as input "n", the number for which digits have to be displayed.
3. Print the digits of the number line-wise.

Constraints

$1 \leq n < 10^9$

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
int main(int argc, char **argv){
```

```
    //write your code here
```

```
    int n{0};
```

```
    //cout<<"Enter the number whose digits you want's to display:"
```

```
    ";
```

```
    cin>>n;
```

```
    int nod{0};
```

```
    int num{n};
```

```
    while(num !=0){
```

```
        nod++;
```

```
        num /= 10;
```

```
    }
```

```
    int q{0};
```

```
    int div{pow(10,nod-1)};
```

```
    while(div!=0){
```

```
        q =n/div;
```

```
        n =n%div;
```

```
        cout<<q<<endl;
```

```
        div/=10;
```

```
    }
```

```
    return 0;
```

```
}
```

Example

Sample Input

```
65784383
```

Sample Output

```
6
5
7
8
4
3
8
3
```

8.Reverse A Number

Easy

1. You've to display the digits of a number in reverse.
2. Take as input "n", the number for which digits have to be display in reverse.
3. Print the digits of the number line-wise, but in reverse order.

Constraints

$1 \leq n < 10^9$

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

```
int main(int argc, char **argv){
    int n;
    cin >> n;

    //write your code here

    // cout<<"Enter the number whose number you want's to display:
"<<endl;

    int num {0};
    int new_n {n};

    do{
        num = new_n%10;
        new_n =new_n/10;
        cout<<num<<endl;
    }while(new_n>=1);

    return 0;

}
```

Example

Sample Input

```
65784383
```

Sample Output

```
3
8
3
4
8
7
5
6
```

9.Inverse Of A Number

Easy

1. You are given a number following certain constraints.
2. The key constraint is if the number is 5 digits long, it'll contain all the digits from 1 to 5 without missing any and without repeating any. e.g. 23415 is a 5 digit long number containing all digits from 1 to 5 without missing

and repeating any digit from 1 to 5. Take a look at few other valid numbers - 624135, 81456273 etc. Here are a few invalid numbers - 139, 7421357 etc.

3. The inverse of a number is defined as the number created by interchanging the face value and index of digits of number. e.g. for 426135 (reading from right to left, 5 is in place 1, 3 is in place 2, 1 is in place 3, 6 is in place 4, 2 is in place 5 and 4 is in place 6), the inverse will be 416253 (reading from right to left, 3 is in place 1, 5 is in place 2, 2 is in place 3, 6 is in place 4, 1 is in place 5 and 4 is in place 6). More examples - inverse of 2134 is 1243 and inverse of 24153 is 24153.

4. Take as input number "n", assume that the number will follow constraints.

5. Print its inverse.

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

```
int main(int argc, char **argv){
    int n;
    cin >> n;

    //write your code here

    int num {0};
    int new_n {n};

    int p{0};
    int inverse{0};
    int order {0};
    do{
        num = new_n%10;
        new_n = new_n/10;
        //cout<<num<<endl;
        p++;
        order = pow(10, num-1);
        inverse += order*p;

    }while(new_n>=1);

    cout<<inverse<<endl;
    return 0;
}
```

Example

Sample Input

28346751

Sample Output

73425681

10.Rotate A Number

Easy

1. You are given two numbers n and k . You are required to rotate n , k times to the right. If k is positive, rotate to the right i.e. remove rightmost digit and make it leftmost. Do the reverse for negative value of k . Also k can have an absolute value larger than number of digits in n .

2. Take as input n and k .

3. Print the rotated number.

4. Note - Assume that the number of rotations will not cause leading 0's in the result. e.g. such an input will not be given

$n = 12340056$

$k = 3$

$r = 05612340$

Constraints

$1 \leq n < 10^9$

$-10^9 < k < 10^9$

Example

Sample Input

```
562984
2
```

Sample Output

```
845629
```

```
#include <iostream>
#include<cmath>
```

```
using namespace std;
int main(){
    int n,k;
    cin>>n>>k;
```

```
//write your code here
```

```
int num{n};
int nod{0};
while (num>0){
    nod++;
    num/=10;
}
```

```
k = k%nod;
```

```
int q{n};
int r{0};
int p{0};
int i(0);
if(k>0){
```

```
    while(i<k){
        r = q%10;
        q = q/10;
```

```

        p = pow(10,nod-1);
        q += r*p;
        i++;
    }
} else if(k<0){
    int d{0};
    while(i>k){
        p = pow(10,nod-1);
        d = q/p;
        r = q%p;
        q =(r*10)+d;
        i--;
    }
}
cout<<q<<endl;
return 0;
}

```

11.Gcd And Lcm

Easy

1. You are required to print the Greatest Common Divisor (GCD) of two numbers.
2. You are also required to print the Lowest Common Multiple (LCM) of the same numbers.
3. Take input "num1" and "num2" as the two numbers.
4. Print their GCD and LCM.

Constraints

$2 \leq n \leq 10^9$

```

#include <iostream>
using namespace std;
int main(int argc, char **argv){
    int num1, num2;
    cin >> num1 >> num2;

    //write your code here

    //GCD
    int a{0};
    int smaller{0};
    smaller = (num1<num2) ? num1 : num2 ;
    for(int i{smaller};i>0;i--){
        if((num1%i==0)&&(num2%i==0)){

```

```

        a = i;
        break;
    }
}
cout<<a<<endl;

//LCM
int b{0};
int larger{0};
larger = (num1>num2) ? num1 : num2;
for(int i{larger};i>1;i++){
    if((i%num1==0)&&(i%num2==0)){
        b = i;
        break;
    }
}
cout<<b<<endl;

return 0;
}

```

12.Prime Factorisation Of A Number

Easy

1. You are required to display the prime factorisation of a number.
2. Take as input a number n.
3. Print all its prime factors from smallest to largest.

For example:

for n = 1440, the output should be: 2 2 2 2 2 3 3 5

Constraints

$2 \leq n < 10^9$

Format

Input

n, an integer

Output

p1 p2 p3 p4.. all prime factors of n

Example

Sample Input

1440

Sample Output

2 2 2 2 2 3 3 5

```
#include <iostream>
using namespace std;
int main(int argc, char **argv){
    int n;
    cin >> n;

    //write your code here
    int num{n};
    for(int i{2}; i<num; i++){
        if(n%i == 0){
            cout<<i<<" ";
            n = n/i;
            i--;
            if(n==1){
                break;
            }
        }
    }
    return 0;
}
```

13. Pythagorean Triplet

Easy

1. You are required to check if a given set of numbers is a valid pythagorean triplet.
2. Take as input three numbers a, b and c.
3. Print true if they can form a pythagorean triplet and false otherwise.

Constraints

1 <= a <= 10⁹

1 <= b <= 10⁹

1 <= c <= 10⁹

Format

Input

a, an integer

b, an integer

c, an integer

Output

true if the numbers form a pythagorean triplet and false otherwise

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

Example

Sample Input

5 3 4

Sample Output

true

```

int main(int argc, char **argv){
    int a,b,c;
    cin >> a >> b >> c;

    //write your code here
    if((a>b)&&(a>c)){
        if((a*a)==((b*b)+(c*c))){
            cout<<"true"<<endl;
        }else{
            cout<<"false"<<endl;
        }
    }else if(b>c){
        if((b*b)==((a*a)+(c*c))){
            cout<<"true"<<endl;
        }else{
            cout<<"false"<<endl;
        }
    }else{
        if((c*c)==((a*a)+(b*b))){
            cout<<"true"<<endl;
        }else{
            cout<<"false"<<endl;
        }
    }

    return 0;
}

```

14.The Curious Case Of Benjamin Bulbs

Easy

1. You are given n number of bulbs. They are all switched off. A weird fluctuation in voltage hits the circuit n times. In the 1st fluctuation all bulbs are toggled, in the 2nd fluctuation every 2nd bulb is toggled, in the 3rd fluctuation every 3rd bulb is toggled and so on. You've to find which bulbs will be switched on after n fluctuations.
2. Take as input a number n, representing the number of bulbs.
3. Print all the bulbs that will be on after the nth fluctuation in voltage.

Constraints

$2 \leq n < 10^9$

```

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

```

Example

Sample Input

6

Sample Output

1
4

Question Video

```
int main(int argc, char**argv){
    int n;
    cin>>n;

    //write your code here
    for(int i{1};i*i<=n;i++){
        cout<<i*i<<endl;
    }
    return 0;
}
```