CODEFORCE

Hello friends, welcome to the tutorial of **"CodeForce"**. "CodeForce" is a platform to present your programming skills on the university level. It's time for everyone should get aware of the talent growing inside you. This tutorial is to help you get accustomed with the programming style and platform used for solving the problem by pouring you logic into any programming language and giving you some tips to help achieve more in the contest.

Before starting these are the key points to be understood properly so that at the contest time your maximum time will be utilized in correct way.

1. Choose correct programming language
2. Use of standard libraries
3. Optimize the algorithm
4. Trade-off space for time

For competing in any programming contest it is important to choose right programming language for your idea to be implemented correctly. Choosing incorrect language may cause the damage of your time and effort. Choice of language is affected by these factors:

1. **Ease of programming:** Choose the language by which you are familiar. Programming in new language or the language with less practice will waste lots of time getting used to it.
2. **Availability of libraries:** Choosing the language having adequate library will help in decreasing the time used for coding as most of the tasks can be done by using appropriate library functions. For example, sorting can be done by using sort command from the library instead of coding for the sorting.

CODEFORCE

There are 3 programming languages supported in this competition viz. C,C++,Java. The compiler for C and C++ will be gcc/g++ (for linux users) and Dev C++ (for windows users). The compiler for Java will be j2sdk1.4 (or higher). Any IDE from can be used for programming for example Eclipse, NetBeans, gedit, kate, They can be downloaded from internet by the following links:

- Dev C++ : http://www.bloodshed.net/devcpp.html
- Java : http://www.oracle.com/index.html
- gcc/g++ : Available in linux distros, if not then install by RPM/yum/apt-get etc.
- gedit, kate : Available in linux distros, if not then install by RPM/yum/apt-get etc.
- Eclipse : http://www.eclipse.org/downloads/
- NetBeans : http://netbeans.org/downloads/

Note:

- If you are using windows and C/C++ as a programming language for the competition then don't use Turbo C/C++ instead use Dev C++(http://sourceforge.net/projects/dev-cpp/files/Binaries/Dev-C%2B%2B%204.9.9.2/devcpp-4.9.9.2_setup.exe/download) otherwise you will not able to successfully submit program in the online contest. Turbo C/C++ is outdated and therefore does not support standards of C/C++.
- Practice on www.codechef.com to get familiar with the online platform used in CodeForce to conduct online programming contest.

CODEFORCE

Coding Guidelines

While coding few things are always required. In this tutorial we will give focus on the following points:

1. Input/output
2. Conditional statements
3. Looping constructs

Input/Output: Input/output is the most important aspect of any programming contest. It is important for the program to take input in correct form and way. It is required that your program take input from Standard input (stdin in linux) and print output in Standard Output (stdout in linux).

Conditional statements: Condition checking is one of the most important points. While coding it should be in your mind that all the conditions are being checked properly or not, especially boundary conditions.

Looping constructs: Loops are very important, they perform the automation task of the program. It is important for a loop to be finish and to run with correct iterations.

Optimization is the most important aspect of coding in any programming contest. The program should be optimized enough to encompass all the features asked in the question. The faster the program will finish, the more the points you will get.

CODEFORCE

Some examples of program

A program to check whether the number is prime or not:

C++ syntax:

```
#include <iostream>
using namespace std;
inline bool isprime(int);

int main()
{
        int i;
        cin>>i;                                         /*Take input from standard input*/
        if(isprime(i))
        {
                cout<<"Yes"<<endl;                      /*Print output on standard output*/
        }
        else
        {
                cout<<"NO"<<endl;                       /*Print output on standard output*/
        }
        return 0;
}

inline bool isprime(int a)
```

```
{
	for(int i=2;i<a;i++)
	{
		if(a%i==0)
		return false;
	}
	return true;
}
```

To compile the program in linux execute the following command in terminal:

```
# g++ -o <name of object file> <name of program file>
```

example (in our case object file is prime and also program file is Prime.cpp)

```
# g++ -o prime Prime.cpp
```

To compile the program in windows using Dev C++ press Crtl+F9

To run the program in linux execute the folowing command in terminal

```
# ./prime
```

To run the program in windows executes the .exe file created in the directory of the program file or press F9 in Dev C++.

C syntax:

```c
#include <stdio.h>
inline int isprime(int);

int main()
{
        int i;
        scanf("%d",&i);                             /*Take input from standard input*/
        if(isprime(i)==1)
        {
                printf("Yes\n");                    /*Print output on standard output*/
        }
        else
        {
                printf("NO\n");                     /*Print output on standard output*/
        }
        return 0;
}

inline int isprime(int a)
{
        int i;
        for(i=2;i<a;i++)
        {
                if(a%i==0)
                return 0;
        }
        return 1;
}
```

To compile c program in linux replace g++ with gcc in the command rest all the steps are same also for running. Compilation and running in windows is same for C and C++ program.

Java syntax:

```
import java.util.*;
public class Prime
{
        public static void main(String args[])
        {
                int i;
                Scanner in = new Scanner(System.in);
                i = in.nextInt();                               /*Take input from standard input*/
                if(isprime(i))
                {
                        System.out.println("YES");      /*Print output on standard output*/
                }
                else
                {
                        System.out.println("NO");       /*Print output on standard output*/
                }

        }
```

```
    public static boolean isprime(int a)
    {
        for(int i=2;i<a;i++)
        {
            if(a%i==0)
            return false;
        }
        return true;
    }
}
```

Java program can be compiled and executed by executing following command in terminal (for linux and windows both).

```
javac <name of program file>
java <name of the class>
```

example

```
javac Prime.java
java Prime
```

Optimization can be brought in the above program by modifying it a little bit. Making change in the isprime() method will increase the optimization level.

Change the for loop condition statement:

for(i=0;i<=sqrt(a);i++)

for this you will need to include specific header file in C/C++ which is math.h and us Math.sqrt instead of simple sqrt in java.

In this program what happened is that we have limit the loop for checking the divisibility upto the square root of the number, by making the use of knowledge that roots of any number can never be greater than its square root, so to check for the number to be prime we check whether the number is divisible by any number between 2 and its square root. This will decrease the check conditions upto 1/sqrt(of number). For example it we are to check for 101 for prime than the loop will not go from 2 to 100, it will go upto 10 only.
Now the program is changed to:

C++ syntax:

```
#include <iostream>
#include <math.h>
using namespace std;
inline bool isprime(int);

int main()
{
        int i;
        cin>>i;                                         /*Take input from standard input*/
        if(isprime(i))
        {
                cout<<"Yes"<<endl;                        /*Print output on standard output*/
        }
        else
        {
                cout<<"NO"<<endl;                         /*Print output on standard output*/
        }
        return 0;
}

inline bool isprime(int a)
{
        for(int i=2;i<=sqrt(a);i++)
        {
                if(a%i==0)
                return false;
        }
        return true;
}
```

C syntax:

```
#include <stdio.h>
inline int isprime(int);

int main()
{
        int i;
        scanf("%d",&i);                         /*Take input from standard input*/
        if(isprime(i)==1)
        {
                printf("Yes\n");                /*Print output on standard output*/
        }
        else
        {
                printf("NO\n");                 /*Print output on standard output*/
        }
        return 0;
}

inline int isprime(int a)
{
        int i;

        for(i=2;i*i<=a;i++)
        {
                if(a%i==0)
                return 0;
        }
        return 1;
}
```

Java syntax:

```
import java.util.*;
public class Prime
{
        public static void main(String args[])
        {
                int i;
                Scanner in = new Scanner(System.in);
                i = in.nextInt();                               /*Take input from standard input*/
                if(isprime(i))
                {
                        System.out.println("YES");      /*Print output on standard output*/
                }
                else
                {
                        System.out.println("NO");       /*Print output on standard output*/
                }
        }
        public static boolean isprime(int a)
        {
                for(int i=2;i<=Math.sqrt(a);i++)
                {
                        if(a%i==0)
                        return false;
                }
                return true;
        }
}
```

A program to find factorial of a number:

C++ syntax (itirative version)

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

inline int factorial(int );

int main()
{
        int i;
        cin>>i;
        cout<<factorial(i)<<endl;
        return 0;
}

inline int factorial(int i)
{
        int j=1;
        for(int a=1;a<=i;a++)
        {
                j *= a;
        }
        return j;
}
```

C++ syntax (recursive version)

```
#include <iostream>
using namespace std;
inline int factorial(int );

int main()
{
        int i;
        cin>>i;
        cout<<factorial(i)<<endl;
        return 0;
}

inline int factorial(int i)
{
        if(i==1)
        {
                return 1;
        }
        else
        {
                return i*factorial(i-1);
        }
}
```

Java syntax (itirative version)

```
import java.util.*;
public class Factorial
{
        public static void main(String args[])
        {
                int i;
                Scanner in = new Scanner(System.in);
                i = in.nextInt();
                System.out.println(factorial(i));
        }
        public static int factorial(int i)
        {
                int j=1;
                for(int a=1;a<=i;a++)
                {
                        j *= a;
                }
                return j;
        }
}
```

Java syntax (recursive version)

```
import java.util.*;
public class Factorial
{
	public static void main(String args[])
	{
		int i;
		Scanner in = new Scanner(System.in);
		i = in.nextInt();
		System.out.println(factorial(i));
	}
	public static int factorial(int i)
	{
		if(i==1)
		{
			return 1;
		}
		else
		{
			return i*factorial(i-1);
		}
	}
}
```

A program to find factorial of larger numbers

C++ syntax

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>

using namespace std;

void mult(int);

int val[2000],i,j;

int main()
{
        int a=0,b,c=0,t;
        string str;
        a=0;
        c=0;
        b=0;
        cin>>str;
        i = str.length();
        c = i-1;
        val[0] = 1;
        for(b=i-1;b>=0;b--)
        {
                a = a + (str[i-b-1]-48)*(pow(10,c));
                c--;
        }
        for(b=2;b<=a;b++)
        {
                mult(b);
        }

        for(b=i-1;b>=0;b--)
        {
                cout<<val[b];
                val[b] = 0;
        }
        cout<<endl;
        return 0;
}
void mult(int a)
{
        int b,c,temp=0;
        for(b=0;b<i;b++)
        {
                temp = val[b]*a + temp;
                val[b] = temp%10;
                temp = temp/10;
        }
        while(temp>0)
```

```
          {
                    val[i] = temp%10;
                    i++;
                    temp = temp/10;
          }
}
```

Java syntax

```
import java.math.*;
import java.util.*;
public class BigFactorial
{
          public static void main(String args[])
          {
                    BigInteger i;
                    Scanner in = new Scanner(System.in);
                    i = in.nextBigInteger();
                    System.out.println(factorial(i).toString());
          }
          public static BigInteger factorial(BigInteger i)
          {
                    BigInteger j = new BigInteger("1");
                    BigInteger k = new BigInteger("1");
                    BigInteger a=new BigInteger("1");
                    for(;a.compareTo(i)<=0;a = a.add(j))
                    {
```

```
                k=k.multiply(a);
            }
            return k;
        }
}
```

A program to find GCD, LCM and Prime Factors

C++ syntax

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

int gcd(int a, int b)
{
        int r=1;
        if(a>b)
        {
                while ( r!=0)
                {
                        r=a%b;
                        a=b;
                        b=r;
                }
                return a;
        }
        else if ( b>a)
        {
                while (r!=0)
                {
                        r=b%a;
                b=a;

                a=r;

            }

            return b;

        }

        else if ( a==b)

            return a;

}


int lcm(int a, int b)

{

        return a*b/gcd(a,b);

}


bool isprime(int a)

{

        if(a==2) return true;

        else

        {

            for(int i=2;i<a/2+1;++i)

            {

                if(a%i==0) return false;

                if(i==a/2) return true;

            }

        }

}
```

```
vector<int> primefactors(int a)
{
	vector<int> primefactors;
	for(int i=1;i<=a;++i)
		if(a%i==0&&isprime(i))
		{
			primefactors.push_back(i);
			a=a/i;i=1;
		}
	return primefactors;
}

int main()
{
	int a,b;
	cin>>a>>b;
	cout<<"GCD = "<<gcd(a,b)<<endl;
	cout<<"LCM = "<<lcm(a,b)<<endl;
	if(!isprime(b)) cout<<"notprime"<<endl;
	vector <int> p =
	primefactors(b);
	for(int i=0;i<p.size();++i)
		cout<<p[i]<<endl;
	return 0;
}
```

CODEFORCE

Other useful Links for practice:

- www.codechef.com
- www.spoj.pl
- www.topcoder.com