**Finding Odd or Even**

Write a program to find whether a given integer is an odd or even number.

**Input and Output Format:**

Input consists of a single integer.

The output consists of a single line, which represents whether it is an even number or not.

**Sample Input 1:**

5

**Sample Output 1:**

5 is an odd number

**Sample Input 2:**

88

**Sample Output 2:**

88 is an even number

using System;

namespace EVEN\_\_OR\_ODD

{

internal class Program

{

private static object cosole;

static void Main(string[] args)

{

int num = int.Parse(Console.ReadLine());

if(num%2==0)

{

Console.WriteLine("Given num is even");

}

else

{

Console.WriteLine("Given nu is odd");

}

}

}

}

**Checking Divisibility By 7 and 3**

Write a program to find whether a given number is divisible by both 7 and 3.

**Input Format:**

Input consists of a single integer.

**Output Format:**

Output consists of a single line. Refer sample output for the format.

**Sample Input 1:**

21

**Sample Output 1 :**

21 is divisible by both 7 and 3

**Sample Input 2:**

18

**Sample Output 2:**

18 is not divisible by both 7 and 3

using System;

namespace Checking\_Divisibility\_By\_7\_and\_3

{

internal class Program

{

private static object cosole;

static void Main(string[] args)

{

int num = int.Parse(Console.ReadLine());

if (num % 7 == 0 && num % 3==0)

{

Console.WriteLine("Given number is Divisibility\_By\_7\_and\_3");

}

else

{

Console.WriteLine("Given number is not Divisibility\_By\_7\_and\_3");

}

}

}

}

**Maximum of Three Numbers**

Write a  program to find the maximum of three numbers.

**Input Format:**

Input consists of three lines. Each line consists of an integer.

**Output Format:**

The output consists of a single line, with the value of a maximum number.  
  
**Sample Input 1:**

3

5

1

**Sample Output 1:**

5 is the maximum number

**Sample Input 2:**

20

60

80

**Sample Output 2:**

80 is the maximum number

using System;

namespace max\_\_among\_3\_num

{

internal class Program

{

static void Main(string[] args)

{

int a = int.Parse(Console.ReadLine());

int b = int.Parse(Console.ReadLine());

int c = int.Parse(Console.ReadLine());

if(a>b)

{

if(a>c)

{

Console.WriteLine("a is big");

}

else

{

Console.WriteLine("c is big");

}

}

else

{

if(b>c)

{

Console.WriteLine("b is big");

}

else

{

Console.WriteLine("c is big");

}

}

}

}

}

**Finding Leap Year**

Write a program to check whether a given year is a leap year or not.

**Input Format:**

Input consists of a single integer.

**Output Format:**

The output consists of a single line. Refer to sample output for details.

**Sample Input 1:**

1988

**Sample Output 1:**

1988 is a leap year

**Sample Input 2:**

1994

**Sample Output 2:**

1994 is not a leap year

**Hint:**

In general terms, the algorithm for calculating a leap year is as follows...

A year will be a leap year if it is divisible by 4 but not by 100. If a year is divisible by 4 and by 100, it is not a leap year unless it is also divisible by 400.

Thus years such as 1996, 1992, 1988, and so on are leap years because they are divisible by 4 but not by 100.

For century years, the 400 rule is important. Thus, century years 1900, 1800 and 1700 while all still divisible by 4 are also exactly divisible by 100. As they are not further divisible by 400, they are not leap-years.

using System;

namespace Finding\_Leap\_Year

{

internal class Program

{

static void Main(string[] args)

{

int year = int.Parse(Console.ReadLine());

if(year % 400 == 0 )

{

Console.WriteLine("Given year is a leap year");

}

else if(year %100 ==0)

{

Console.WriteLine("Given year is not a leap year");

}

else if ( year % 4 == 0)

{

Console.WriteLine("Given year is a leap year");

}

else

{

Console.WriteLine("Given year is not a leap year");

}

}

}

}

**EarthQuake Intensity Decision**

Mr. Ram is a very rich businessman and he  lost his family in the Gujarat Earthquake. He lost interest in his business after the tragic incident and he decided to serve the society. He started an NGO named after his only son Santosh to help the earth-quake victims. Santosh Helpline will provide a compensation to all earth quake survivors.

The compensation amount given to the survivors is not fixed and it depends on the intensity of the earthquake.

The expression **Richter magnitude scale** refers to a number of ways to assign a single number to quantify the energy contained in an earthquake.

Earthquake Magnitude Scale

|  |  |
| --- | --- |
| **Magnitude** | **Type** |
| 2.4 or less | Micro |
| 2.5 to 5.4 | Light |
| 5.5 to 6.0 | Moderate |
| 6.1 to 6.9 | Strong |
| 7.0 to 7.9 | Major |
| 8.0 or greater | Great |

Please help Ram to decide the intensity of the earthquake.

**Input Format:**

Input consists of a single floating point number which corresponds to the earthquake's ritcher magnitude scale.

**Ouput Format:**

Output consists of the string “Micro” or “Light” or “Moderate” or “Strong” or “Major” or Great”

Refer sample input and output for further formatting specifications.

**[All text in bold corresponds to input and the rest corresponds to output]**

**Sample Input and Output 1:**

Richter Magnitude:

**5.7**

Moderate

**Sample Input and Output 2:**

Richter Magnitude:

**5**

Light

using System;

namespace EarthQuake\_Intensity\_Decision

{

internal class Program

{

static void Main(string[] args)

{

double num = double.Parse(Console.ReadLine());

if(num<2.4)

{

Console.WriteLine("Magnitude is micro");

}

else if (num >= 2.4 && num <= 5.4)

{

Console.WriteLine("Magnitude is light");

}

else if (num >= 5.5 && num <= 6.0)

{

Console.WriteLine("Magnitude is moderate");

}

else if (num >= 6.1 && num <= 6.9)

{

Console.WriteLine("Magnitude is strong");

}

else if (num >= 7.0 && num <= 7.9)

{

Console.WriteLine("Magnitude is major");

}

else if (num >8.0)

{

Console.WriteLine("Magnitude is great");

}

else

{

Console.WriteLine("Magnitude is not in range");

}

}

}

}

**Grade Calculation**

Write a program to determine the grade of the student in a particular subject. Refer to the table given below for grade details.

|  |  |
| --- | --- |
| **Marks scored** | **Grade** |
| 100 | S |
| [90,100) | A |
| [80,90) | B |
| [70,80) | C |
| [60,70) | D |
| [50,60) | E |
| <50 | F |

The interval **[a,b]** includes all numbers greater than or equal to **a** and less than **b**.

**Input and Output Format:**  
Input consists of a single integer that corresponds to the marks scored by the student.

Print **"Invalid Input"** if it is not in the range **0 to 100**.

Refer sample input and output for formatting specifications.

**[All text in bold corresponds to input and the rest corresponds to output.]**

**Sample Input and Output 1:**  
Enter the marks

**85**

The student obtained a B grade

**Sample Input and Output 2:**  
Enter the marks

**850**

Invalid Input

using System;

using System.Diagnostics;

namespace Grade\_Calculation

{

internal class Program

{

static void Main(string[] args)

{

int marks = int.Parse(Console.ReadLine());

if (marks == 100)

{

Console.WriteLine("Grade is S");

}

else if (marks < 100 && marks >= 90)

{

Console.WriteLine("Grade is A");

}

else if (marks < 90 && marks >= 80)

{

Console.WriteLine("Grade is B");

}

else if (marks < 80 && marks >= 70)

{

Console.WriteLine("Grade is C");

}

else if (marks < 70 && marks >= 60)

{

Console.WriteLine("Grade is D");

}

else if (marks < 60 && marks >= 50)

{

Console.WriteLine("Grade is E");

}

else if (marks < 50)

{

Console.WriteLine("Grade is F");

}

else

{

Console.WriteLine(" ");

}

}

}

}

**Palindromic Prize**

A customer in the Personalised Gift Store is awarded a prize when their bill number is a 3-digit palindrome.Write a program for identifying the prize winners.

**Input Format:**

Input consists of a number that corresponds to the bill number.

**Output Format:**

The output consists of a string that is either 'yes' or 'no'. The output is 'yes' when the customer receives the prize and is 'no' otherwise.

**Sample Input 1:**

565

**Sample Output 1:**

yes

**Sample Input 2:**

568

**Sample Output 2:**

no

**Sample Input 3:**

66

**Sample Output 3:**

No

using System;

namespace Palindrome\_prize

{

internal class Program

{

static void Main(string[] args)

{

int r;

int sum = 0;

int n = int.Parse(Console.ReadLine());

int temp = n;

while (n>0)

{

r = n % 10;

sum = (sum \* 10) + r;

n = n / 10;

}

if(temp == sum)

{

Console.WriteLine("yes");

}

else

{

Console.WriteLine("no");

}

}

}

}

**Sorted Prize**

A customer in the Personalised Gifts Store is awarded a prize when their bill number is a 3-digit number and all the 3 digits are in sorted order. (Examples ---> 379, 256, 973, 652, 225, 522 ...]

Icon

Description automatically generated                          Icon

Description automatically generated  
Help Gita in identifying the prize winners.

**Input Format:**

Input consists of a number which corresponds to the bill number.

**Output Format:**  
The output consists of a string that is either 'yes' or 'no'. The output is yes when the customer receives the prize and is no otherwise.

**Sample Input 1:**

565

**Sample Output 1:**

no

**Sample Input 2:**

620

**Sample Output 2:**

yes

**Sample Input 3:**

66

**Sample Output 3:**

No

using System;

namespace Celsius\_to\_Fahrenheit\_Converter

{

internal class Program

{

static void Main(string[] args)

{

string l = Console.ReadLine();

if (l.Length == 3)

if (l[0] > l[1] && l[1] > l[2])

Console.WriteLine("Yes");

else if (l[0] < l[1] && l[1] < l[2])

Console.WriteLine("Yes");

else

Console.WriteLine("No");

else

Console.WriteLine("No");

}

}

}