**Please write code which takes a whole number and spells it out in words.**

private static string oneDigit(string Numbers)

{

int number = Convert.ToInt32(Numbers);

string name = string.Empty;

switch (number)

{

case 1:

name = "One";

break;

case 2:

name = "Two";

break;

case 3:

name = "Three";

break;

case 4:

name = "Four";

break;

case 5:

name = "Five";

break;

case 6:

name = "Six";

break;

case 7:

name = "Seven";

break;

case 8:

name = "Eight";

break;

case 9:

name = "Nine";

break;

}

return name;

}

private static string twoDigit(string Numbers)

{

int number = Convert.ToInt32(Numbers);

string name = string.Empty;

switch (number)

{

case 10:

name = "Ten";

break;

case 11:

name = "Eleven";

break;

case 12:

name = "Twelve";

break;

case 13:

name = "Thirteen";

break;

case 14:

name = "Fourteen";

break;

case 15:

name = "Fifteen";

break;

case 16:

name = "Sixteen";

break;

case 17:

name = "Seventeen";

break;

case 18:

name = "Eighteen";

break;

case 19:

name = "Nineteen";

break;

case 20:

name = "Twenty";

break;

case 30:

name = "Thirty";

break;

case 40:

name = "Fourty";

break;

case 50:

name = "Fifty";

break;

case 60:

name = "Sixty";

break;

case 70:

name = "Seventy";

break;

case 80:

name = "Eighty";

break;

case 90:

name = "Ninety";

break;

default:

if (number > 0)

{

name = twoDigit(Numbers.Substring(0, 1) + "0") + " " + oneDigit(Numbers.Substring(1));

}

break;

}

return name;

}

public static string ConvertWholeNumber(string number)

{

string word = string.Empty;

bool beginsZero = false;

bool isDone = false;

double dblNumber = Convert.ToDouble(number);

beginsZero = number.StartsWith("0");

int numDigits = number.Length;

int pos = 0;

String place = string.Empty;

switch (numDigits)

{

case 1://ones' range

word = oneDigit(number);

isDone = true;

break;

case 2://tens' range

word = twoDigit(number);

isDone = true;

break;

case 3://hundreds' range

pos = (numDigits % 3) + 1;

place = " Hundred ";

break;

case 4:

case 5:

case 6:

pos = (numDigits % 4) + 1;

place = " Thousand ";

break;

case 7:

case 8:

case 9:

pos = (numDigits % 7) + 1;

place = " Million ";

break;

case 10:

case 11:

case 12:

pos = (numDigits % 10) + 1;

place = " Billion ";

break;

default:

isDone = true;

break;

}

if (!isDone)

{

if (number.Substring(0, pos) != "0" && number.Substring(pos) != "0")

{

word = ConvertWholeNumber(number.Substring(0, pos)) + place + ConvertWholeNumber(number.Substring(pos));

}

else {

word = ConvertWholeNumber(number.Substring(0, pos)) + ConvertWholeNumber(number.Substring(pos));

}

}

return word;

}

}

public static void Main(string[] args)

{

string isNegative = "";

Console.WriteLine("Enter a Number to convert");

string number = Console.ReadLine();

number = Convert.ToDouble(number).ToString();

if (number.Contains("-"))

{

isNegative = "Minus ";

number = number.Substring(1, number.Length - 1);

}

if (number == "0")

{

Console.WriteLine("The number is \nZero Only");

}

else

{

Console.WriteLine("The number is \n{0}", isNegative + ConvertToWords(number));

}

Console.ReadKey();

}

}