1.

Write an **if**-statement that takes two integer variables and **exchanges**

their values if the first one is greater than the second one.

Console.WriteLine("Enter your number1");

int number1 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter your number2");

int number2 = int.Parse(Console.ReadLine());

int exchanges = number1;

if (number2>number1)

{

exchanges = number2;

}

Console.WriteLine($"{exchanges} is the greater number");

2.

Write a program that shows the sign (**+** or **-**) of the product of three real

numbers, without calculating it. Use a sequence of **if** operators.

using System;

namespace Question\_2

{

class Program

{

static void Main(string[] args)

{ Console.WriteLine(" Enter your first sign (+ or -) in number ");

int sign1 = int.Parse(Console.ReadLine());

Console.WriteLine(" Enter your second sign (+ or -) in number ");

int sign2 = int.Parse(Console.ReadLine());

Console.WriteLine(" Enter your third sign (+ or -) in number ");

int sign3 = int.Parse(Console.ReadLine());

if (sign1 < 0 && sign2 < 0 && sign3 > 0)

{

Console.WriteLine("you have (+)");

}

else if (sign1 < 0 && sign2 > 0 && sign3 < 0)

{

Console.WriteLine(" you have (+)");

}

else if (sign1 > 0 && sign2 < 0 && sign3 < 0)

{

Console.WriteLine(" you have (+)");

}

else if (sign1 > 0 && sign2 < 0 && sign3 < 0)

{

Console.WriteLine(" you have (-)");

}

else if (sign1 > 0 && sign2 < 0 && sign3 < 0)

{

Console.WriteLine(" you have (-)");

}

else if (sign1 < 0 && sign2 > 0 && sign3 < 0)

{

Console.WriteLine(" you have (-)");

}

else if (sign1 <= 0 && sign2 <= 0 && sign3 <= 0)

{

Console.WriteLine(" you have (-)");

}

else if (sign1 > 0 && sign2 > 0 && sign3 > 0)

{

Console.WriteLine(" you have (+)");

}

3.

Write a program that finds the **biggest of three integers**, using nested

**if** statements.

Console.WriteLine("Enter your number");

int number1 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter your number");

int number2 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter your number");

int number3 = int.Parse(Console.ReadLine());

if (number1 > number2 && number1 > number3 )

{

Console.WriteLine($"{number1} is the biggest between {number2} and {number3} ");

}

else if (number2 > number1 && number2 > number3 )

{

Console.WriteLine($"{number2} is the biggest between {number1} and {number3}");

}

else if (number3 > number2 && number3 > number1)

{

Console.WriteLine($"{number3} is the biggest between {number1} and {number2}");

}

4.

**Sort 3 real numbers** in descending order. Use nested **if** statements.

Console.WriteLine("Enter your number");

int number1 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter your number");

int number2 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter your number");

int number3 = int.Parse(Console.ReadLine());

if (number1 < number3)

{

number1 = number1 + number3;

number3 = number1 - number3;

number1 = number1 - number3;

Console.WriteLine($" The number in ascending is {number1} , {number2} , {number3} ");

}

if (number2 < number3)

{

number2 = number2 + number3;

number3 = number2 - number3;

number2 = number2 - number3;

Console.WriteLine($" The number in ascending is {number1} , {number2} , {number3} ");

}

if (number1 < number2)

{

number1 = number1 + number2;

number2 = number1 - number2;

number1 = number1 - number2;

Console.WriteLine($" The number in ascending is {number1} , {number2} , {number3} ");

}

else

{

Console.WriteLine($" The number {number1} , {number2} , {number3} is invalid ");

}

5.

Write a program that asks for a digit (0-9), and depending on the input,

**shows the digit as a word** (in English). Use a **switch** statement

Console.WriteLine("Enter your Digit between 0 - 9 and see it word");

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

switch (digit)

{

case 0:

Console.WriteLine("Zero");

break;

case 1:

Console.WriteLine("One");

break;

case 2:

Console.WriteLine("Two");

break;

case 3:

Console.WriteLine("Three");

break;

case 4:

Console.WriteLine("Four");

break;

case 5:

Console.WriteLine("Five");

break;

case 6:

Console.WriteLine("Six");

break;

case 7:

Console.WriteLine("Seven");

break;

case 8:

Console.WriteLine("Eight");

break;

case 9:

Console.WriteLine("Nine");

break;

default:

Console.WriteLine("You enter wrong input the number is between 0 and 9,");

break;

6.

Write a program that gets the coefficients ***a***, ***b*** and ***c*** of a quadratic

equation: ***a*x2 + *b*x + *c***, calculates and prints its real roots (if they exist).

Quadratic equations may have 0, 1 or 2 real roots.

7.

Write a program that finds the **greatest of given 5 numbers**.

int maximum = 0;

for (int i = 0; i < 5; i++)

{

Console.WriteLine(" Enter your number ");

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

if (number>maximum)

{

maximum = number;

}

}

Console.WriteLine($" The maximum number is {maximum}");

Write a program that, depending on the user’s choice, inputs **int**, **double**

or **string** variable. If the variable is **int** or **double**, the program

increases it by 1. If the variable is a **string**, the program appends "**\***" at

the end. Print the result at the console. Use **switch** statement.

Console.WriteLine(" Enter 0=int, 1=double, 2=string,");

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

switch (input)

{

case 0 :

Console.WriteLine("Enter int type");

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

int o = 1;

Console.WriteLine($" your int is {type} + {o} ");

break;

case 1:

Console.WriteLine("int variable +1 = int type");

double type2 = int.Parse(Console.ReadLine());

double d =1;

Console.WriteLine($"your double is {type2 + d} ");

break;

case 2 :

Console.WriteLine("Enter string type");

string type3 = Console.ReadLine();

string star = "\*";

Console.WriteLine($" your string is {type3 + star} ");

break;

default:

Console.WriteLine("Invalid input ");

break;

9.

We are given 5 integer numbers. Write a program that finds those

**subsets whose sum is 0**. Examples:

-

If we are given the numbers {3, -2, 1, 1, 8}, the sum of -2, 1 and 1

is 0.

-

If we are given the numbers {3, 1, -7, 35, 22}, there are no subsets

with sum 0.

Console.WriteLine("Enter first number");

int number1 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter second number");

int number2 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter third number");

int number3 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter fourth number");

int number4 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter fifth number");

int number5 = int.Parse(Console.ReadLine());

if (number1+number2==0)

{

Console.WriteLine($" = {number1} , {number2}");

}

if (number1 + number2 + number3 == 0)

{

Console.WriteLine($" {number1} , {number2} , {number3}");

}

if (number1 + number2 + number3 + number4 == 0)

{

Console.WriteLine($" = {number1} , {number2} , {number3} , {number4}");

}

if (number1 + number2 + number3 + number4 + number5 == 0)

{

Console.WriteLine($" {number1} , {number2} , {number3} , {number4} , {number5}");

}

if (number2 + number3 == 0)

{

Console.WriteLine($"{number2} , {number3}");

}

if (number2 + number3 + number4 == 0)

{

Console.WriteLine($" {number2} , {number3} , {number4}");

}

if (number2 + number3 + number4 + number5 == 0)

{

Console.WriteLine($" {number2} , {number3} , {number4} , {number5}");

}

if (number3 + number4 == 0)

{

Console.WriteLine($"{number3} , {number4}");

}

if (number3 + number4 + number5 == 0)

{

Console.WriteLine($" {number3} , {number4} , {number5}");

}

if (number4 + number5 == 0)

{

Console.WriteLine($" {number4} , {number5}");

}

else

{

Console.WriteLine($" number is not in subset");

}

10.

Write a program that applies **bonus points** to given scores in the range

[1…9] by the following rules:

-

If the score is between 1 and 3, the program multiplies it by 10.

-

If the score is between 4 and 6, the program multiplies it by 100.

-

If the score is between 7 and 9, the program multiplies it by 1000.

-

If the score is 0 or more than 9, the program prints an error

message.

Console.WriteLine("Enter number between 1 and 9");

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

if (input > 1 && input <= 3)

{

Console.WriteLine($" your point is {input\*10}");

}

if (input >= 4 && input <= 6)

{

Console.WriteLine($" your point is {input \* 100}");

}

if (input >= 7 && input <= 9)

{

Console.WriteLine($" your point is {input \* 1000}");

}

else

{

Console.WriteLine($" Error message");

}

**Chapter 6. Loops**

1.

Write a program that prints on the console **the numbers from 1 to N**.

The number **N** should be read from the standard input.

Console.WriteLine("Enter your number ");

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

for (int i = 0; i < number; i++)

{

Console.WriteLine(i);

}

2.

Use a **for**-loop and the operator **%** for finding **the remainder** in integer

division. A number **num** is not divisible by 3 and 7 simultaneously exactly

when (**num % (3\*7) == 0**).

Console.WriteLine("Enter number ");

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

for (int i = 0; i < number; i++)

{

if (i % 3 != 0)

{

if (i % 7 !=0)

{

Console.WriteLine(i);

}

}

}

3.

Write a program that reads from the console a series of integers and

prints the **smallest** and **largest** of them.

int lowest = 0;

int largest = 0;

Console.WriteLine("Enter number");

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

for (int i =0;i<number; i++)

{

if(i == 0)

{

if (lowest > number )

{

lowest = number;

}

if (largest < number)

{

largest=number;

}

}

}

Console.WriteLine($" The largest is {largest} and the lowest is {lowest}");

4.

Write a program that prints **all possible cards from a standard deck**

of cards, without jokers (there are 52 cards: 4 suits of 13 cards).

for (int i = 0; i < 4; i++)

{

if (i != 0)

{

}

switch (i)

{

case 1:

Console.WriteLine("Club");

break;

case 2:

Console.WriteLine("Diamond");

break;

case 3:

Console.WriteLine("Heart");

break;

case 4:

Console.WriteLine("Spades");

break;

default:

break;

}

for (int j = 0; j < 14; j++)

{

switch (j)

{

case 1:

Console.WriteLine("1");

break;

case 2:

Console.WriteLine("2");

break;

case 3:

Console.WriteLine("3");

break;

case 4:

Console.WriteLine("4");

break;

case 5:

Console.WriteLine("5");

break;

case 6:

Console.WriteLine("6");

break;

case 7:

Console.WriteLine("7");

break;

case 8:

Console.WriteLine("8");

break;

case 9:

Console.WriteLine("9");

break;

case 10:

Console.WriteLine("10");

break;

case 11:

Console.WriteLine("J");

break;

case 12:

Console.WriteLine("Q");

break;

case 13:

Console.WriteLine("K");

break;

case 14:

Console.WriteLine("A");

break;

default:

Console.WriteLine("Invalid input");

break;

}

Console.WriteLine(i);

5.

Write a program that reads from the console number N and print the sum

of the first N members of the **Fibonacci sequence**: 0, 1, 1, 2, 3, 5, 8,

13, 21, 34, 55, 89, 144, 233, 377,

int value1 = 0;

int value2 = 1;

int value3 = 0;

Console.WriteLine("r1");

Console.WriteLine("r2");

Console.WriteLine("Enter your number");

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

for (int i = 0; i < n; i++)

{

value3 = value2 + value1;

value1 = value2;

value2 = value3;

Console.WriteLine(value3);

}

6.

Write a program that calculates **N!/K!** for given N and K (1<K<N)

Console.WriteLine("Enter first number for factorial N ");

int number1 = int.Parse(Console.ReadLine());

int factorialn = 1;

for (int i = 1; i <= number1 ; i++)

{

factorialn = factorialn \* i;

}

Console.WriteLine(factorialn);

Console.WriteLine("Enter first number for factorial K ");

int number2 = int.Parse(Console.ReadLine());

int factorialk = 1;

for (int i = 1; i <= number2; i++)

{

factorialk = factorialk \* i;

}

Console.WriteLine(factorialk);

int value = factorialn / factorialk;

Console.WriteLine(value);

7.

Write a program that calculates **N!\*K!/(N-K)!** for given N and K

(1<K<N).

Console.WriteLine("Enter first number for factorial N ");

int number1 = int.Parse(Console.ReadLine());

int factorialn = 1;

for (int i = 1; i <= number1; i++)

{

factorialn = factorialn \* i;

}

Console.WriteLine(factorialn);

Console.WriteLine("Enter first number for factorial K ");

int number2 = int.Parse(Console.ReadLine());

int factorialk = 1;

for (int i = 1; i <= number2; i++)

{

factorialk = factorialk \* i;

}

Console.WriteLine(factorialk);

int value = factorialn \* factorialk / (factorialn - factorialk);

Console.WriteLine(value);

}

8.

Use the same concept of **canceling the faction of simple factors**, like

you probably did in the **previous problem**.

You may also read more about the **Catalan numbers** in Wikipedia

(http://en.wikipedia.org/wiki/Catalan\_number) and use the **recurrent**

**formula** for calculating them.

9.

10.

Write a program that reads from the console a **positive integer number**

**N** (N < 20) and prints a **matrix** of numbers as on the figures below:

**N = 3**

**N = 4**

1

2

3

2

3

4

3

4

5

1

2

3

4

2

3

4

5

3

4

5

6

4

5

6

7

Console.WriteLine("Enter number");

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

for (int i = 0; i < number; i++)

{

for (int j = 0; j <= i; j++)

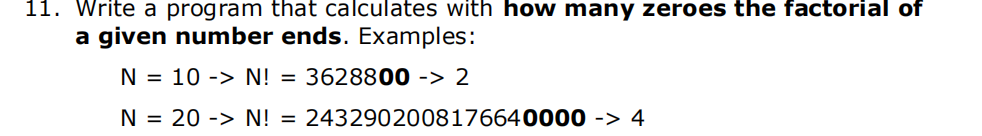
{

Console.WriteLine(j);

}

Console.WriteLine(i);

}



ulong value = 0;

Console.WriteLine("Enter a number");

ulong number = ulong.Parse(Console.ReadLine());

for (ulong i= number-1; i > 0; i--)

{

number \*= i;

Console.WriteLine($"N! is {number} ");

if (number % 10 ==10)

{

number /= 10;

value++;

Console.WriteLine(value);

}

}

16.

17.

Console.Write("Enter first number: ");  
    int a = Int32.Parse(Console.ReadLine());  
    Console.Write("Enter second number: ");  
    int b = Int32.Parse(Console.ReadLine());  
      
    while (a != 0 && b != 0)  
    {  
        if (a > b) a %= b;  
        else b %= a;  
    }if (a == 0) Console.WriteLine(b);  
    else Console.WriteLine(a);