# Personal tips and notes for programming!

(programing basics)

1. **Separating Prime from Non-Prime numbers:**

* **Code example:**

int num1, num2, num3, num4;

num1 = int.Parse(Console.ReadLine());//starting number

num2 = int.Parse(Console.ReadLine());//ending number

for ( num3 = num1; num3 <= num2; num3++)

{

num4 = 0;//filter for non-Prime numbers(moje i bool izdraz)

for (int i = 2; i <=num3/2 ; i++)// where the magic happens :D

{

if (num3%i==0)//i zapochva ot 2 znachi ako chetno vliza, vkliuchva filtyra i izliza.

{

num4++;

break;

}

}

if (num4 ==0 && num3!=1)// ako ima filtyr ne vliza

{

Console.WriteLine(num3);//prime numebr!

}

}

* + - * **Method Example:**

public static class PrimeTool

{

public static bool IsPrime(int candidate)

{

// Test whether the parameter is a prime number.

if ((candidate & 1) == 0)

{

if (candidate == 2)

{

return true;

}

else

{

return false;

}

}

// Note:

// ... This version was changed to test the square.

// ... Original version tested against the square root.

// ... Also we exclude 1 at the end.

for (int i = 3; (i \* i) <= candidate; i += 2)

{

if ((candidate % i) == 0)

{

return false;

}

}

return candidate != 1;

}

}

class Program

{

static void Main(string[] args)

{

int num1 = int.Parse(Console.ReadLine());

for (int i = 0; i < num1; i++)

{

bool isPrime = PrimeTool.IsPrime(i);

if (isPrime)

{

Console.WriteLine($"obicham te {i}");

}

}

}

}

}

1. **Getting 2,3,4… digits from a number:**

* **Explanation:**

Searching through 2,3,4 and many digit numbers with nested FOR loops works easier when you have a FOR loop for each digit from that number, instead of converting is to string.Lenght and then getting each char out of it and convert it to number once again.

If we have 1234 as a number, simplest way is to make a FOR loop for each digit (for first digit will be from 1 to 9, for second from 1 to 9 and so on)

* + - **Code example:**

static void Main(string[] args)

{

int num = int.Parse(Console.ReadLine());//Just a case IGNORE

for (int first = 1; first <= 9; first++)//first digit

{

for (int second = 1; second <= 9; second++)//second digit

{

for (int third = 1; third <= 9 ; third++)//third digit

{

for (int fourth = 1; fourth <= 9; fourth++)//fourth digit

{

if ((first+second)==(third+fourth))//just a case IGNORE

{

string number = first.ToString() + second + third + fourth;//make them as one string

int luckyNum = int.Parse(number);//convert the string to number

if (num % (first + second) == 0)//just a case IGNORE

{

Console.Write($"{luckyNum} ");

}

}

}

}

}

}

}

}

1. **Converting:**

* **Explanation and code:**

There are many ways to convert one type to another but this is what I find the best:

Anything to string - example.ToString();

int number = 12345;//the number

string numberToString = number.ToString();//the number converted to string

String to number - if the string is made of digits representing a number then is - int.Parce/double.Parce(example);

string name = "1234";

int nameToNum = int.Parse(name);

Else if the string does not represent a number then you can only convert each character into Char and then take its ASKII value:

string example = "Teo";//the string we want to convert

char first = example[0];//the first character of it "T"

char second = example[1];// the second character of it

char third = example[2];//the third character of it

int fisrtToNum = (int)first;//the first character ASKII number value

int secondToNum = (int)second;// the second character ASKII number value

int thirdToNum = (int)third;// the third character ASKII number value

number to char - converting number to char will get the number and find its character replacement from ASKII table - (char)example;

int a = 200;//random number

char b = (char)a;//ASKII char value of the number

int c = b;//ASKII num value of the char

1. **Finding Percentage**

* Explanation and code:

Finding percent of number can simply be done by subtract the number by 100 and multiply it by the percent you want.

double percent = 10;//percent we want to take off or put on

double number = 100;//number we want to work with

double percentFromNumber = number / 100 \* percent;//the number value of percent from number

ticketCount/freeSpots \*100//finding how many percents of freespots are the tickets

You can do a quicker calculations by multiplying the number by (100 - percent) /100

double percent = 10;//percent we want to take off or put on

double number = 100;//number we want to work with

double numberAfter = number \* 0.90;//number after we took percent out

double NumberAfter2 = number \* 1.10;//number after we add extra percent

1. **Finding Odd and Even number**

* Code example:

int number = 10;//number in question, usually input number

if (number%2==0)//if number / 2 gives to extra (6/2 = 3 and 0 extra)

{

Console.WriteLine("even");//the number is even

}

else if (number%2!=0)//or number%2==1. If the number / 2 gives extra (5/2 = 2 and 1 extra)

{

Console.WriteLine("odd");//the number is odd

}