שיעורי בית יסודות 16.11 – פעולות, אופיר הופמן י3

תרגיל 1

```
public static int DigitNum (int num)
{
    int count = 0; // Counter - number of digits

    while (num > 0) // Disassemble the number
    {
        count++;
        num /= 10;
    }
    return count;
}
```

תרגיל 2

```
public static int Digits_Sum (int num)
{
    int sum = 0;

    while (num > 0)
    {
        sum += num % 10;
        num /= 10;
    }
    return sum;
}
```

תרגיל 3.א.

```
public static int Digit_num_count (int num, int digit)
{
    int count = 0;

    while (num > 0)
    {
        if ((num % 10) == digit)
        {
            count++;
        }
        num /= 10;
    }
    return count;
}
```

תרגיל 3.ב.

```
static void Main(string[] args)
{
    Console.WriteLine("Enter number (digit)");
    int digit = int.Parse(Console.ReadLine());

    while (digit != 0)
    {
        Console.WriteLine("Enter a number");
        int num = int.Parse(Console.ReadLine());

        int pelet = Digit_num_count(num, digit);
        Console.WriteLine($"This number has the digit {digit} -
{pelet} times in it");
    }
}
```

תרגיל 4

```
public static void Dividers (int num)
{
    for (int i = 1; i < num; i++) // Repeat from 1 to number-1
    {
        if (num % i == 0) // check if number divides by it
        {
            Console.WriteLine(i); // print the divider
        }
    }
}
```

המשך למטה

תרגיל 5

```
public static bool Perfect (int num)
{
    int sum = 0; // sum od dividers
    for (int i = 1; i < num; i++) //go over all numbers from 1
to num-1
    {
        if (num % i == 0)// check if is divisible
        {
            sum += i; // add to sum of dividers
        }
    }
     // check if sum of dividers equals to number
    if (sum == num)
    {
        return true;
    }
    return false;
}
```

תרגיל 6

```
public static bool Prime (int num)
{
    bool is_prime = true;
    for (int i = 2; i < num; i++) // repeat on every number
from 2 to num-1
    {
        if (num % i == 0)// check if divisible
        {
            is_prime = false;
        }
    }

    if (is_prime) // final check if prime before returning
        return true;

    return false;
}
```

המשך למטה

תרגיל 7

```
public static bool Palindrome (int num)
{
    //save the num as string (gibui)
    string saveNum = "";
    saveNum += num;

    string RevNum = ""; //reversed number variable

    // reverse the number
    while (num > 0)
    {
        RevNum += (num % 10); //the units and add to reversed number
        num /= 10; // get rid of the units
    }

    // check if reversed number is the same as the original number
    if (saveNum == RevNum)
    {
        return true;
    }
    return false;
}
```

תרגיל 8

```
public static int CommonFactor (int num1, int num2)
{
    int MaxCommonFactor = 0; // biggest common factor

    bool found = false; // condition for keep looking for a factor

    // go over all numbers from the smallest of them down to 1
    for (int i = Math.Max(num1, num2); i > 0 && found == false; i--)
    {
        if (num1 % i == 0 && num2 % i == 0)
        {
            MaxCommonFactor = i;
            found = true;
        }
    }
    return MaxCommonFactor;
}
```

תרגיל 9

```
public static int Factorial (int num)
{
    int kefel = num; // multiplication of all numbers

    // go over all numbers from num-1 down to 2
    for (int i = num - 1; i > 1; i--)
    {
        kefel *= i; // multiply by i
    }
    return kefel;
}
```

תרגיל 10.א.

```
public static double Power3 (double num)
{
    return Math.Pow(num, 3);
}

static void Main(string[] args)
{
    int num = int.Parse(Console.ReadLine());
    Console.WriteLine(Power3(num));
}
```

תרגיל 10.ב.

```
public static bool Shilush (int num)
{
    double sum = 0; // sum of units cubed
    int saveNum = num; // gibui

    // peiruk number
    while (num != 0)
    {
        double units = num % 10; // get units
        sum += Math.Pow(units, 3); //cube units
        num = num / 10; //get rid of units
    }

    // check if sum of units cubed == number
    if (saveNum == sum)
    {
        return true;
    }
    return false;
```

```
}
static void Main(string[] args)
{
    int num = int.Parse(Console.ReadLine());
    Console.WriteLine(Shilush(num));
}
```

תרגיל 10.ג.

```
static void Main(string[] args)
{
    // go over all positive integers
    for (int i = 1; i < int.MaxValue; i++)
    {
        bool shilush = Shilush (i); //shilush of i
        if (shilush == true) // check if true
        {
            Console.WriteLine(i);
        }
    }
}
```

המשך למטה

תרגיל 11.א. + ב.

```
public static bool Zariz (int num)
{
    bool is_zariz = true;

    // Peiruk of the digits
    while (num > 10 && is_zariz)
    {
        int dig = num % 10; //current units
        int nextDig = (num / 10) % 10; // next digit (dozens)

        //check if units if bigger than dozens
        If (dig > nextDig)
        {
            is_zariz = false;
        }
        num /= 10; //get rid of units
    }
    return is_zariz;
}

static void Main(string[] args)
{
    int num = int.Parse(Console.ReadLine());
    // Zakif
    while(num != 999)
    {

        // check if number is Zariz using the method above
        bool zariz = Zariz(num);
        if(zariz == true)
        {
            Console.WriteLine("This number is zariz");
        }
        else
        {
            Console.WriteLine("This number is NOT zariz");
        }

        num = int.Parse(Console.ReadLine());// get a new number
    }
}
```