**שיעורי בית ביסודות רקורסיה 4.17 + אתגר – אופיר הופמן י3**

**דף עבודה**

**תרגיל 1 – מספר הספרות**

public static int DigCount(int num)

{

if (num < 10)

return 1;

return DigCount(num / 10)+1;

}

**תרגיל 2 – סכום ספרות**

public static int DigSum(int num)

{

if (num < 10)

return num;

return DigSum(num / 10) + num%10;

}

**תרגיל 3 - חזקה**

public static int Power(int baseNum, int power)

{

if (power == 0)

return 1;

return Power(baseNum, power - 1) \* baseNum;

}

**תרגיל 4 - עצרת**

public static int Factorial(int num)

{

if (num == 1)

return 1;

return Factorial(num-1)\*num;

}

**תרגיל 5 – איבר בסדרת פיבונצ'י**

public static int Fibonacci(int n)

{

if (n <= 1)

return n;

return Fibonacci(n-1)+Fibonacci(n-2);

}

**תרגיל 6 – סכום שני מספרים**

public static int GetSum(int num1, int num2)

{

if (num2 == 0)

return num1;

return GetSum(num1+1, num2-1);

}

**תרגיל 7 – מכפלת שני מספרים**

public static int GetMul(int num1, int num2)

{

if (num2 == 1)

return num1;

return num1 + GetMul(num1, num2 - 1);

}

**תרגיל 8 – מציאת ערך בסדרה**

public static int FindInSeries(int n, int first, int difference)

{

if (n == 1)

return first;

return difference + FindInSeries(n - 1, first, difference);

}

**תרגיל 9 – מציאת סכום איברים ראשונים בסדרה**

public static int FindInSeries(int n, int first, int difference)

{

if (n == 1)

return first;

return difference + FindInSeries(n - 1, first, difference);

}

**תרגיל 10 – שעון חול**

public static void Hourglass(int n)

{

if (n == 1)

{

Console.WriteLine("1");

return;

}

Console.WriteLine(new String((char)(n+48), n));

Hourglass(n - 1);

Console.WriteLine(new String((char)(n+48), n));

}

**המשך למטה**

**מצגת רקורסיה**

**תרגיל 6 - האם ספרות המספר בסדר עולה**

public static bool SederOle(int num)

{

if (num == 0)

return true;

return SederOle(num / 10) && num % 10 > (num/10)%10;

}

**תרגיל 7 – האם הספרה נמצאת במספר**

public static bool IsDigInNum(int num, int digit)

{

if (num < 10)

return num == digit;

return (digit == num % 10) || IsDigInNum(num / 10, digit);

}

**תרגיל אתגר – העכבר והשוקולד**

public static int MouseAndChoclate(int[,] arr)

{

// Base Case (stop condition) - a 1x1 array

if (arr.GetLength(0) == 1)

return arr[0, 0];

// Make an array of N-1 x N-1 of the square around the point with more choclate

int [,] newArr = new int[arr.GetLength(0) - 1, arr.GetLength(0)-1];

// more choclate down

if (arr[1, 0] > arr[1, 1])

{

for (int i = 0; i < newArr.GetLength(0); i++)

{

for (int j = 0; j < newArr.GetLength(0); j++)

{

newArr[i, j] = arr[i+1, j];

}

}

}

// more choclate on down right diagonal

else

{

for (int i = 0; i < newArr.GetLength(0); i++)

{

for (int j = 0; j < newArr.GetLength(0); j++)

{

newArr[i, j] = arr[i + 1, j + 1];

}

}

}

return arr[0, 0] + MouseAndChoclate(newArr);

}