1.Kreirati matricu od 5 redova i 5 kolona, sa vrednostima redom od 1 do 25.

```
int[,] matrica = new int[5, 5];
for (int i = 0; i < 5; i++)
{
    for (int j = 0; j < 5; j++) { matrica[i, j] = (i * 5) + j + 1; }
}
```

2. Napisati funkciju koja ispisuje elemente prethodne matrice, u zasebnim redovima za svaki red matrice.

```
for (int i = 0; i < matrica.GetLength(0); i++)
{
    for (int j = 0; j < matrica.GetLength(1); j++)
    {
        Console.Write($"{matrica[i, j]} ");
    }
    Console.WriteLine();
}
```

3. Ispisati šahovsku tablu gde 0 predstavlja belo polje, a 1 crno.

```
int[,] SahovskaTabla = new int[8, 8];
bool crno = true;
for (int i = 0; i < SahovskaTabla.GetLength(0); i++)
{
    for (int j = 0; j < SahovskaTabla.GetLength(1); j++)
    {
        if (crno) Console.BackgroundColor = ConsoleColor.Black;
        else Console.BackgroundColor = ConsoleColor.White;

        Console.Write("  ");
        crno = !crno;
    }
    crno = !crno;
    Console.WriteLine();
}
```

4. Napisati algoritam koji računa n-ti stepen zadatog broja.

```
static double Stepen(int broj, int stepen)
{
    if (stepen == 0) return 1;
    else if (stepen > 0)
        return broj * Stepen(broj, stepen - 1);
    else
        return 1.0 / broj * Stepen(broj, stepen + 1);

}
```

5. Napisati algoritam koji računa faktorijel broja (4! = 1*2*3*4 = 24, 3! = 1*2*3 = 6).

```
static int Faktorijel(int broj)
{
    if (broj <= 0) return 1;

    return broj * Faktorijel(broj - 1);

}
```

6. Napisati algoritam koji računa najveći zajednički delilac dva prirodna broja.

```
static int NZD(int a, int b)
{
    if (a > b)
    {
        if (a % b == 0)
            return b;
        return NZD(a - b, b);
    }
    else
    {
        if (b % a == 0)
            return a;
        return NZD(b - a, a);
    }
}
```

Optimalnije rešenje:

```
static int NZD(int a, int b)
{
    if (b == 0)
        return a;

    return NZD(b, a % b);

}
```

7. Napisati algoritam koji računa n-ti član Fibonačijevog niza (0,1,1,2,3,5,8,13,21,34,55,89,144…).

```
static int Fibonaci(int n)
{
    if (n <= 0) return 0;

    if (n == 1) return 1;

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

}
```