

Зад.1 Да се напише функцията, която приема като параметър списък с числа. За всеки елемент от списъка да се направи проверка дали е неотрицателно число. В случай че даден елемент е неотрицателен, да се коренува и резултата да се отпечата на екрана. В противен случай, да се отпечата на екрана подходящо съобщение, например "The result is not a real number".

```
In[37]:= myFunc[list_] :=  
  For[i = 1, i ≤ Length[list], i++,  
    If[list[[i]] ≥ 0,  
      Print[Sqrt[list[[i]]],  
      Print["The result is not a real number..."];  
    ]  
  ]
```

```
In[38]:= myFunc[{0.1, -17, 12, 44, -148}]
```

0.316228

The result is not a real number...

$2\sqrt{3}$

$2\sqrt{11}$

The result is not a real number...

Зад.2 Да се пресметне НОД на 144 и 15, като се използва алгоритъма на Евклид. Алгоритъмът да се имплементира **итеративно**.

*Упътване: За пресмятане на остатъка при деление на две числа можете да използвате функцията **Mod**.*

Алгоритъм на Евклид:

Взимайки двете подадени като вход числа A и B ($A > B$), провери дали B е равно на 0.

Ако да, числото A е търсеният НОД на A и B.

В противен случай, повтори процеса, като използваш за входни данни B и остатъка, получен при делението на A на B.

Пример: 1) A = 8, B = 6;

2) B > 0 => A = 6, B = 2;

3) B > 0 => A = 2, B = 0;

4) B = 0 => НОД (A, B) = A = 2.

```
In[39]:= a = 144; b = 15;  
While[b > 0, mod = Mod[a, b]; a = b; b = mod]  
a
```

```
Out[41]= 3
```

Зад. 2.1 (Допълнителна задача към Зад. 2) Да се напише функция, която приема като параметри две естествени числа. Функцията да намира НОД на двете числа по алгоритъма на Евклид, като алгоритъма се имплементира **рекурсивно**. За простота можем да считаме, че първия подаден параметър е винаги по-голям от втория.

```
In[42]:= myGCD[a_, b_] := If[b == 0, a, myGCD[b, Mod[a, b]]]
```

Зад. 3 Дефинирайте функция, която по подадени координати на точка в двумерното пространство, да определя в кой квадрант се намира тя. Да се обработят също случаите, в

които дадена точка лежи на някоя от координатните оси. За целта, използвайте структурата **Piecewise**.

```
In[43]:= whereAmI[x_, y_] := Piecewise[{{"Origin", x == 0.0 && y == 0.0},
{"Abscissa", y == 0}, {"Ordinate", x == 0}, {"1st", x > 0 && y > 0},
{"2nd", x < 0 && y > 0}, {"3rd", x < 0 && y < 0}, {"4th", x > 0 && y < 0}}]
```

Зад. 4 Да се напише функция, която приема като параметри две матрици и намира произведението им. В случай че размерностите на матриците не позволяват да се извърши умножението, да се изведе подходящо съобщение на екрана. В противен случай, функцията да връща резултатната матрица. Функцията да се напише итеративно (като се използват подходящи структури за цикъл) и да се сравнят резултатите от нея с резултатите от вградената функция “.” (**Dot**).

Упътване: За определяне на размерностите на входните матрици, можете да използвате функцията **Dimensions**.

```
In[44]:= matrixMultiplicator[m1_, m2_] := (
  If[Length[Dimensions[m1]] == 1 || Length[Dimensions[m2]] == 1,
    Return["Invalid input!"]];
  If[Dimensions[m1][[2]] != Dimensions[m2][[1]],
    Return["Matrices have incompatible dimensions!"],

    product = Table[0, {i, Dimensions[m1][[1]]}, {j, Dimensions[m2][[2]]}];
    For[row = 1, row <= Dimensions[m1][[1]], row++,
      For[col = 1, col <= Dimensions[m2][[2]], col++,
        product[[row, col]] = Sum[m1[[row, k]] * m2[[k, col]], {k, Dimensions[m1][[2]]}];
      ];
    ];
  product // MatrixForm
)
```

Зад. 5 Да се пресметне $10!$ с помощта на конструкцията **Do**. Сравнете получения резултат с този, върнат от системата *Mathematica*.

```
In[45]:= init = 1;
Do[init *= n, {n, 1, 10}]
init
```

```
Out[47]= 3 628 800
```

```
In[48]:= 10!
```

```
Out[48]= 3 628 800
```