Függvények

```
#include <stdio.h>

int main()
{
   utasítás;
   utasítás;
   ...
}
```

Saját függvények

```
#include <stdio.h>

void függvény_név()
{
   utasítás;
   utasítás;
   ...
}

int main()
{
   utasítás;
   függvény_név();
   utasítás;
   ...
}
```

1. Példa - argumentum (független változó) és visszatérítési érték nélkül

In [1]:
```
#include <stdio.h>

void koszones()
{
    char nev[20];
    printf("\nKérlek, add meg a neved.\n");
    scanf("%s", &nev);
    printf("\nHello %s!",nev);
}

int main()
{
    printf("\nEz egy egyszerű mintaprogram, ami köszönni tud.");
    koszones();
    printf("\nProgram vége.");
}
```

```
Ez egy egyszerű mintaprogram, ami köszönni tud.
Kérlek, add meg a neved.

Hello Béla!
Program vége.
```

2. Példa - argumentum (független változó) nélkül, visszatérítési értékkel

In [2]:
```
#include <stdio.h>

int kor();

int main()
{
    int ev;

    printf("\nEz egy egyszerű mintaprogram, ami megkérdezi az életkorod.");
    ev = kor();
    printf("\nTe %d éves vagy. ", ev);
    printf("\nProgram vége.");
}

int kor()
{
    int evek_szama;

    printf("\nKérlek, add meg az életkorod.\n");
    scanf("%d", &evek_szama);

    return evek_szama;
}
```

```
Ez egy egyszerű mintaprogram, ami megkérdezi az életkorod.
Kérlek, add meg az életkorod.

Te 18 éves vagy.
Program vége.
```

3. Példa - argumentummal, visszatérítési érték nélkül

In [3]:
```
#include <stdio.h>

void paros(int szam);

int main()
{
    int n;

    printf("\nEz egy egyszerű mintaprogram, ami eldönti a bevitt számról, hogy páros-e.");
    printf("\nAdj meg egy egész számot: ");
    scanf("%d", &n);
    paros(n);
}

void paros(int szam)
{
    if (szam%2==0){
        printf("A szám páros!");
    }
    else {
        printf("A szám páratlan!");
    }
}
```

```
Ez egy egyszerű mintaprogram, ami eldönti a bevitt számról, hogy páros-e.
Adj meg egy egész számot:

A szám páros!
```

4. Példa - argumentummal és visszatérítési értékkel

In [4]:
```
#include <stdio.h>

int prim_ellenorzes(int n);

int main()
{
    int szam, valto;

    printf("Adj meg egy pozitív egész számot: ");
    scanf("%d", &szam);

    valto = prim_ellenorzes(szam);

    if(valto == 1)
        printf("A %d nem prímszám!",szam);
    else
        printf("A %d prímszám!",szam);

    return 0;
}


int prim_ellenorzes(int n)
{
    for(int i=2; i <= n/2; ++i)
    {
        if(n%i == 0)
            return 1;
    }

    return 0;
}
```

```
Adj meg egy pozitív egész számot:

A 12 nem prímszám!
```

1. Feladat

Készítsen egyszerű programot a tömb(ök) manipulálásra. Az egyes műveleteket deklarálja függvényekként. A program rendelkezzen az alábbi funkciókkal:

- tömb feltöltése véletlen számokkal
- tömb törlése (kinullázása)
- tömb kiíratása
- tömb két elemének felcserélése
- tömb legnagyobb és legkisebb elemének megkeresése

In [14]:
```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

void print_array(int tomb[], int hossz){
     for (int i = 0; i < hossz; i++){
         printf("%d ",tomb[i]);
    }
    printf("\n");
}

void generate_array(int tomb[], int hossz, int min, int max){
    for (int i = 0; i < hossz; i++){
        tomb[i] = rand() % (max+ 1 - min) + min;
    }
}

int main(){
    srand(time(NULL));
    int array[] = {1,2,3,4,5};
    int array_2[] = {1,2,3,4,5,6,7,8,9,10};

    print_array(array, 5);
    print_array(array_2, 10);
    generate_array(array,5,100,1000);
    generate_array(array_2,10,0,1);
    print_array(array, 5);
    print_array(array_2, 10);
}
```

```
1 2 3 4 5
1 2 3 4 5 6 7 8 9 10
951 675 198 906 833
0 0 1 1 1 1 1 1 0 1
```

2. Feladat

Készítsen egyszerű számológépet, ahol az egyes műveletek saját függvények formájában vannak deklarálva.

3. Feladat

Készítsen átalautomatát, az egyes funkciókat hozza létre saját függvények formájában. A rendelés felvételére és számla kiállításához használjon tömböt.

4. Feladat

Készítsen szöveges játékot, ahol a felhasználó minden körben 2-4 lehetőségből választ. A helyes opció véletlen generált, a játék addig folytatódik amíg a felhasználó jól tippel. A játék során adott válaszok tömbben vannak eltárolva és a végén kiíratásra kerülnek.

5. Feladat

Készítsen szókitaláló játékot. A keresett szó minden betűje kezdeti állapotban "_" karakterrel van helyettesítve, majd helyes tipp esetén az adott betű behelyettesítésre kerül. A program megvalósításához használjon függvényeket indokolt esetben. Ha egy adott betű többször szerepel a megoldásban, akkor csak az első előfordulást fedi fel.

- legyen lehetőség rekérdezni a megfejtésre
- megoldásban többször előforduló karakter esetén az összes előfordulást fedje fel (pl. tipp: a -> a__a)

Mintakimenet:

```
A keresett szó: ____
Mit tippelt? a
A keresett szó tartalmaz "a" betűt.
A keresett szó a___
```

6. Feladat

Készítsen programot személyi azonosító ellenőrzésére! Indokolt esetben deklaráljon függvényt bizonyos funkciók megvalósítására!

A személyi szám úgy nevezett „beszélő szám", azaz struktúrája van. 11 decimális számjegyből áll és **M ÉÉHHNN SSSK** alakú. Bővebb információ: https://hu.wikipedia.org/wiki/Szem%C3%A9lyi_azonos%C3%ADt%C3%B3

- az M számjegy alapvetően a nemre és a születési év első két jegyére utal. Az 1997. január 1. előtt születtetek esetében tartalmazta az állampolgárságot is.
- az ÉÉHHNN számjegyek a születési év utolsó két jegyét, a hónapot és a napot kódolják.
- az SSS az azonos napon születettek megkülönböztetésére való.
- a K ellenőrzési célokat szolgál. A többi számjegyből kell képezni. Egyszerűbb hibák, elütések detektálhatók a segítségével. Képlete:

$$K_{11} = (1 * k_1 + 2 * k_2 + 3 * k_3 + \ldots + 10 * k_{10}) \mod 11$$

In [ ]: