

1. pelda

Osztályzat meghatározása pontszámból (if-else, switch, for, vektor):

```
#include <stdio.h>
float pontszam[30]; // a hallgatók elért pontszamai
int osztalyzat[30]; //index 0-29 között lehet
main()
{
    int i, letszam, szumma;
    printf("    Osztályzatmeghatározás az elért pontszám alapján");
    printf("\n    és osztályátlag számítása.\n\n");
    printf("Az osztálylétszám=");
    scanf("%d",&letszam);
    // Elért pontszámok bekeresése:
    for ( i = 0; i<letszam; i++ )
    {
        printf("\nAz %d. tanuló pontszáma=", i+1);
        scanf("%f",&pontszam[i]);
    }
    // Osztályzatok számítása:
    for ( i = 0 ; i<letszam; i++ )
    {
        // ez a kapcsos blokkzárójel-par elmaradhatna most
        if ( pontszam[i] >= 90 )
            osztalyzat[i] = 5;
        else if (pontszam[i] >= 77 )
            osztalyzat[i] = 4;
        else if (pontszam[i] >= 67 )
            osztalyzat[i] = 3;
        else if (pontszam[i] >= 51 )
            osztalyzat[i] = 2;
        else
            osztalyzat[i] = 1;
    }
    // Osztályzatok kiírása és az átlagszámítás előkészítése:
    for ( i = 0, szumma = 0; i<letszam; i++ )
    {
        printf("\nA %d. hallgató osztályzata= ", i+1);
        switch (osztalyzat[i])
        {
            case 1:
                printf("elegtelen");
                break;
            case 2:
                printf("elegendő");
                break;
            case 3:
                printf("közepes");
```

```
        break;
    case 4:
        printf("jo");
        break;
    case 5:
        printf("jeles");
        break;
    }
    szumma += osztalyzat[i];           // szumma = szumma +
osztalyzat[i];
}
// Az atlag kiszamitasa kiiratas kozben: (double) nelkul
egeszosztas lenne
printf("\n\nAz osztalyatlag= %5.2lf", szumma / (double)letszam );
}
```

2. példa

“Indul a gorog aludni”, szovegforditos jatek: karaktervektor, while.

```
#include <stdio.h>
main()
{
    char mondat[80];
    char forditott[80];
    int i, hossz;
    printf("Mondatfordito jatek\n\n");
    printf("Add meg a mondatot:\n");
    gets(mondat); // mondat maga egy cim, egy konstans mutatoertek
    // scanf-fel csak szokoz nelkuli szoveget tudnank bevenni
    // Biztos, ami biztos, feltoltjuk \0 karakterekkel a forditott
vektort:
    for ( i = 0; i<80 ; i++ )
    {
        forditott[i] = '\0';
    }
    // Megallapitjuk a mondat hosszat (mert meg nem ismerjuk az strlen())
fv-t,
    // es hogy gyakoroljuk a while utasitast is:
    hossz = 0;
    while ( mondat[hossz] != '\0' ) // vagy egyszeruen (mondat[hossz])
    {
        hossz++;
    }
    // Mondatforditas, ugyelve a mondatzaro \0 karakterre:
    for ( i = 0; i < hossz; i++)
    {
```

```
        forditott[i] = mondat[hossz-i-1]; // (hossz-1) az utolso betu
indexe
    }
    // Kiiratas:
    printf("\nA forditott mondat:\n%s",forditott);
}
```

3. pelda

Sztringek kozott nincs ertelmezve az ertekadas, nekunk kell beturol-beture atmasolni. A do-while eppen megfelel.

```
#include <stdio.h>
main()
{
    char eredeti[80];
    char masolat[80];
    int i;
    printf("Szovegvaltozo tartalmanak atadasa egy masik szovegvaltozo
reszere\n");
    printf("\nAz eredeti szoveg:\n");
    gets(eredeti); // itt nem kell az & operator, mert 'eredeti' maga
egy pointer is egyben
    printf("\nA masolat vektor tartalma masolas elott:\n %s", masolat);
    // a do-while ciklus idealis, mert legalabb a \0 karaktert at kell
masolni:
    i = -1;
    do
    {
        i++;
        masolat[i] = eredeti[i];
    }
    while ( eredeti[i] != '\0' ) ; // vagy egyszeruen (eredeti[i])
    printf("\nA masolat masolas utan:\n%s", masolat);
    getch();
}
```

4. pelda

Egy vektor elemeit a pointeraritmetika segitsegevel erjuk el, indexeles helyett.

```
#include <stdio.h>
#include <conio.h>
main()
{
    double meretek[25];
    double min;
```

```
int db,i,minindex;
printf("A meretek kozul az erdekel minket, amelyik a 100-tol
nagyobbak\n");
printf("kozul a legkisebb\n\n");
printf("Add meg a meresek szamat=");
scanf("%d",&db);
printf("\n\nAdd meg a mereteket:\n");
for ( i = 0; i < db; i++ )
{
    printf("\nAz %d. meret=",i);
    scanf("%lf",&meretek[i] );    // vagy (%lf",(meretek+i) )
}
for ( i = 0, min = 1.7E300; i < db; i++)
{
    if ( *(meretek+i) > 100  &&  *(meretek+i) < min )
    {
        minindex = i;
        min = *(meretek+i);
    }
}
printf("\nA 100-tol nagyobbak kozul a legkisebb a %d indexu,
erteke=%lf",
        minindex, *(meretek+minindex) );
}
```

From:
<https://edu.iit.uni-miskolc.hu/> - Institute of Information Science - University of Miskolc

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:tovabbi_algoritmusok_ii

Last update: 2022/09/05 21:53

