

## Vektorok kezelése

**1. feladat:** Kérjen be a felhasználótól egy számot (n) és egy 100 elemű char vektorban hozzon létre n darab véletlen kis-nagy betűkből álló jelszót.

```
#include <stdio.h>
#include <time.h>
int main()
{
    char password[100];
    int n, i;
    printf("Hany karakteres legyen a jelszo?");
    scanf("%d", &n);
    srand(time(0));
    for(i = 0; i < n; i++) {
        password[i] = rand() % ('z' - 'A') + 'A';
    }
    password[i] = '\0';
    printf("%s", password);
}
```

**2. feladat:** Keressük meg egy egészekből álló tömb második legnagyobb elemét.

```
#include <stdio.h>

int main()
{
    int tomb[] = {44, 45, 23, 43, 7};
    int n = sizeof(tomb) / sizeof(int);
    int i;
    // legnagyobb elem megkeresese
    int max = tomb[0];
    for(i = 0; i < n ; i++)
    {
        if(max < tomb[i])
        {
            max = tomb[i];
        }
    }
    // masodik legnagyobb elem megkeresese
    int max2 = tomb[0];
    for(i = 0; i < n ; i++)
    {
        if(max != tomb[i] && tomb[i] > max2)
        {
            max2 = tomb[i];
        }
    }
}
```

```
    }  
  }  
  printf("A legnagyobb = %d\n", max);  
  printf("A 2. legnagyobb = %d\n", max2);  
}
```

**3. feladat:** Írassuk ki egy tömb elemeit fordított sorrendben.

```
#include <stdio.h>  
  
int main()  
{  
    int tomb[] = {44, 45, 23 , 43 , 7};  
    int n = sizeof(tomb) / sizeof(int);  
    int i;  
    for(i = n-1; i >= 0; i--)  
    {  
        printf("%d ", tomb[i]);  
    }  
}
```

**4. feladat:** Adott egy 5 elemű egészekből álló tömb. Kérjünk be a felhasználótól egy új elemet és egy pozíciót. Szúrjuk be a tömbbe az új elemet és írassuk ki az eredményt.

```
#include <stdio.h>  
  
int main()  
{  
    int tomb[6] = {44, 45, 23 , 43 , 7};  
    int n = sizeof(tomb) / sizeof(int);  
    int i;  
    int b, ii;  
    printf("Mit szeretne beilleszteni?");  
    scanf("%d", &b);  
    printf("Hanyadik elemkent szeretne beilleszteni?");  
    scanf("%d", &ii);  
    for(i = n-1; i >= ii; i--)  
    {  
        tomb[i] = tomb[i - 1];  
    }  
    tomb[ii] = b;  
  
    for(i = 0; i < n; i++)  
    {  
        printf("%d ", tomb[i]);  
    }  
}
```

```
}  
}
```

**5. feladat:** "Forgassuk el" egy előre megadott vektor elemeit megadott számszor balra, a balról kieső elemet utolsó elemként használjuk fel.

```
#include <stdio.h>  
  
int main()  
{  
    int tomb[] = {44, 45, 23, 43, 7};  
    int n = sizeof(tomb) / sizeof(int);  
    int i, rot;  
    printf("Hanszor forgatunk balra?");  
    scanf("%d", &rot);  
  
    while(rot--)  
    {  
        int x = tomb[0];  
        for(i = 1; i < n; i++)  
        {  
            tomb[i - 1] = tomb[i];  
        }  
        tomb[n - 1] = x;  
    }  
  
    for(i = 0; i < n; i++)  
    {  
        printf("%d ", tomb[i]);  
    }  
}
```

**6. feladat:** Adott egy vektor ami azonos elemeket is tartalmazhat. Töröljük ki a többször szereplő elemeket és írassuk ki az új vektort.

```
#include <stdio.h>  
  
int main()  
{  
    int tomb[] = {10, 20, 10, 1, 100, 10, 2, 1, 5, 10, 32};  
    int n = sizeof(tomb) / sizeof(int);  
    int tomb2[n];  
    int i, j, k = 0;  
    int dupla = 0;  
    for(i = 0; i < n - 1; i++)  
    {  
        for(j = i + 1; j < n; j++)
```

```
{
    if(tomb[i] == tomb[j])
    {
        for(int m = j; m < n-1; m++)
        {
            tomb[m] = tomb[m+1];
        }
        n--;
    }
}

for(i = 0; i < n; i++)
{
    printf("%d ", tomb[i]);
}
}
```

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:vektorok\\_peldak\\_2?rev=1662413420](https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:vektorok_peldak_2?rev=1662413420)

Last update: 2022/09/05 21:30

