

5. labor kis ZH, 2014. december 3., szerda 10-12

NÉV (olvasható):

NEPTUN kód:

- Definiálj** struktúrát, amely alkalmas egy állat nevének (max. 40 karakter) és hosszának tárolására!
- Írj **függvényt**, amely rendezi a paraméterként átvett, az a) feladatban definiált struktúrákból álló tömböt az állatok hossza szerint!
- Egészítsd ki **teljes programmá**! A program hozzon létre egy, a felhasználó által megadott méretű dinamikus tömböt az a)-ban definiált struktúrákból! Tölts fel tetszőleges névvel/nevekkel, a méreteket pedig 50 és 500 közötti véletlen egész értékekkel! Írja ki a tárolt név-hossz párokat hossz szerinti növekvő sorrendben! Szabadítsa fel a tömböt!

max. 10p, megoldási idő: 20 perc

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

typedef struct Adat{
    char nev[41];
    int hossz;
} Adat;

void rendez(Adat t[], int n){
    int i, j;
    for (i = n; i > 0; i--){
        for (j = 0; j < i - 1; j++){
            if (t[j].hossz > t[j + 1].hossz){
                Adat temp = t[j];
                t[j] = t[j + 1];
                t[j + 1] = temp;
            }
        }
    }
}

int main(void){
    int i, n;
    Adat *tomb;

    scanf("%d", &n);
    tomb = (Adat*)malloc(n*sizeof(Adat));
    if (tomb == NULL)
        return 1;

    srand(time(0));
    for (i = 0; i < n; i++){
        strcpy(tomb[i].nev, "kutya");
        tomb[i].hossz = rand() % 451 + 50;
    }

    rendez(tomb, n);

    for (i = 0; i < n; i++){
        printf("%s: %d\n", tomb[i].nev, tomb[i].hossz);
    }

    free(tomb);

    return 0;
}
```

5. labor kis ZH, 2014. december 5., péntek 8-10

NÉV (olvasható):

NEPTUN kód:

- d) **Definiálj** struktúrát, amely alkalmas egy állat fajának (max. 50 karakter) és lábszámának tárolására!
- e) Írj **függvényt**, amely rendezi a paraméterként átvett, az a) feladatban definiált struktúrákból álló tömböt lábszám szerint!
- f) Egészítsd ki **teljes programmá**! A program hozzon létre egy, a felhasználó által megadott méretű dinamikus tömböt az a)-ban definiált struktúrákból! Tölts fel tetszőleges fajjal/fajokkal, a lábszámot pedig 0 és 1000 közötti véletlen páros értékekkel! Írja ki a tárolt faj-lábszám párokat lábszám szerinti növekvő sorrendben! Szabadítsa fel a tömböt!

max. 10p, megoldási idő: 20 perc

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

typedef struct Adat{
    char faj[51];
    int labaszam;
} Adat;

void rendez(Adat t[], int n){
    int i, j;
    for (i = n; i > 0; i--){
        for (j = 0; j < i - 1; j++){
            if (t[j].labaszam > t[j + 1].labaszam){
                Adat temp = t[j];
                t[j] = t[j + 1];
                t[j + 1] = temp;
            }
        }
    }
}

int main(void){
    int i, n;
    Adat *tomb;

    scanf("%d", &n);
    tomb = (Adat*)malloc(n*sizeof(Adat));
    if (tomb == NULL)
        return 1;

    srand(time(0));
    for (i = 0; i < n; i++){
        strcpy(tomb[i].faj, "kutya");
        tomb[i].labaszam = (rand() % 501) * 2;
    }

    rendez(tomb, n);

    for (i = 0; i < n; i++){
        printf("%s: %d\n", tomb[i].faj, tomb[i].labaszam);
    }

    free(tomb);

    return 0;
}
```