

Základy programování (IZP)

9. počítačové cvičenie

Brno University of Technology, Faculty of Information Technology
Božetěchova 1/2. 602 00 Brno - Královo Pole
ilazur@fit.vut.cz



Stiahnite si kostru programu

<https://github.com/Jorgen98/IZP/blob/main/cv9/sort.c>

Postupne opravte kód podľa zadania

```
typedef struct {  
    unsigned year;  
    char *name;  
} Person;
```

```
typedef struct {  
    unsigned len;  
    Person *data;  
} PersonArray;
```

Issue 1

```
// Destruktor. Vycisti zaznam o osobe  
void person_dtor(Person *p)  
{  
    p->year = 0;  
    p->name = NULL;  
    // chyba! neni uvolnovana pamet pro jmeno  
}
```

Issue 2

```
// Vytvori novy objekt a zkopiruje data o osobe ze src na dst  
// Vraci NULL, pokud se nepovedlo  
void *person_copy(Person *src, Person *dst)  
{  
    *dst = *src;  
    return dst;  
    // chyba! Neni provadena kopie jmena a roku  
}
```

Issue 1

```
void person_dtor(Person *p)
{
    p->year = 0;
    free(p->name);
    p->name = NULL;
}
```

Issue 2

```
void *person_copy(Person *src, Person *dst)
{
    person_ctor(dst, src->year, src->name);
    return dst;
}
```

Issue 3

```
void array_dtor(PersonArray *a)
{
    if (a->len)
        free(a->data);
    // chyba! Je potreba zlikvidovat postupne vsechny zaznamy.
}
```

Issue 3

```
void array_dtor(PersonArray *a)
{
    if (a->len)
    {
        for (unsigned int i = 0; i < a->len; i++)
        {
            person_dtor(&a->data(i));
        }
        free(a->data);
    }
}
```

Implementovanie radiacich funkcií

Selection sort

Funkcia pre nájdenie minimálneho prvku

```
// Najde v poli osobu, ktora ma najmensi year (a pripadne prvni v  
// abecede)  
// Minimum hleda od indexu idx. Pro hledani v celem poli treba  
// zadat index 0.  
// Vyuzijte funkci person_cmp  
unsigned int array_find_min(PersonArray *a, int idx)  
{  
    // TODO  
}
```

Implementovanie radiacich funkcií

Selection sort

Funkcia pre nájdenie minimálneho prvku

```
unsigned int array_find_min(PersonArray *a, int idx)
{
    unsigned int min = idx;
    for (int i = idx + 1; i < a->len; i++)
    {
        if (person_cmp(&a->data(i), &a->data(min)) < 0)
        {
            min = i;
        }
    }

    return min;
}
```


Implementovanie radiacich funkcií Selection sort

Funkcia pre zoradenie prvku

```
// Seradi osoby v poli podle jejich roku (a pripadne jmena)
void array_sort(PersonArray *a)
{
    // TODO
}
```

Implementovanie radiacich funkcií

Selection sort

Funkcia pre zoradenie prvku

```
void array_sort(PersonArray *a)
{
    unsigned int min = 0;
    for (int i = 0; i < a->len - 1; i++) {
        min = array_find_min(a, i);
        person_swap(&a->data(i), &a->data(min));
    }
}
```

Stiahnite si kostru programu

<https://github.com/Jorgen98/IZP/blob/main/cv9/fib.c>

Implementujte funkciu pre výpočet postupnosti

```
ull fib(ull n)
{
    ull result;
    // TODO
    return result;
}
```

Implementujte funkciu pre výpočet postupnosti

```
ull fib(ull n)
{
    nf++;
    if (n == 0) {
        return 0;
    } else if (n == 1) {
        return 1;
    }
    return fib(n - 1) + fib(n - 2);
}
```

Stačí vypracovať jednu variantu

- Alokujte pole čísiel o veľkosti 10 a inicializujte toto pole hodnotami zo STDIN. Následne pole zoraďte pomocou selection sortu. Nezabudnite na korektnú prácu s pamäťou.
- Načítajte jeden argument programu, jedno celé číslo. Spočítajte pomocou rekurzcie súčet všetkých čísiel od 0 po dané číslo.
- Načítajte jeden argument programu, jedno celé číslo. Pre dané číslo vypočítajte s využitím rekurzcie hodnotu jeho faktoriálu.