

Pro uložení prvků do uspořádaného (setříděného) pole sestavte šablonu:

```
template<class T,unsigned n>
class UsporadanePole { };
```

Parametry šablony:

- **T** - datový typ ukládaného prvku
- **n** - rozsah pole (počet prvků v poli)

Operátory a funkce šablony **Pole**:

**UsporadanePole & operator << (const T &);** Vloží další prvek do pole. Pokud prvek se stejnou hodnotu je již v poli nebo pole je plné, vložení je ignorováno. Prvek je vložen na příslušné místo tak, aby bylo zachováno uspořádání (setřídění) prvků. Pokud uložené prvky jsou objekty nějaké třídy, tato má pro tento účel přetížené operátor srovnání, které srovnávají obsahy objektů třídy.

**const T \*najit(const T &);** Binárním vyhledáváním vyhledá v poli prvek zadaný v parametru funkce. Pokud uložené prvky jsou objekty nějaké třídy, tato má pro tento účel přetížené operátor srovnání **<** a **>**, které srovnávají obsahy objektů třídy. Byl-li prvek v tabulce nalezen, operace vrátí ukazatel na tento prvek, pokud nebyl nalezen, vrátí nulový ukazatel.

**bool operator -= (const T &p);** Odebere prvek z pole. Operace vrátí hodnotu *true*, jestliže odebrání bylo úspěšné. Pokud prvek nebyl nalezen, vrátí hodnotu *false*.

**operator unsigned () const;** Vrátí počet prvků uložených v poli.

Dále šablona bude obsahovat iterátor pro průchod všemi prvky uloženými v tabulce. Funkce iterátoru:

**bool prvni();** Nastaví iterátor na první prvek, který je uložen v poli, a vrátí hodnotu *true*. Není-li uložen žádný prvek, vrátí *false*.

**const T & aktual();** Vrátí referenci na aktuální prvek.

**bool dalsi();** Pokud další prvek existuje, posune iterátor na další prvek a vrátí *true*. Pokud už další prvek neexistuje, vrátí *false*.

Začněte do programu funkci:

```
template<class T,unsigned n>
void najit(UsporadanePole<T,n> &pz &pz,const Zlomek &z)
{
    auto p=pz.najit(z);
    if (p==nullptr) { cout << "nenalezen: "; z(); }
    else { cout << "nalezen: "; (*p)(); }
}
```

Prvky ukládané do pole budou objekty třídy **Zlomek** (úloha 6c).

Do hlavní funkce začněte deklarace a příkazy:

```
UsporadanePole<Zlomek,20> pz;
typedef Zlomek Z;
pz << Z(1,3) << Z(3,5) << Z(7,4) << Z(3,4) << Z(2,3) << Z(7,2)
    << Z(5,4) << Z(1,4) << Z(6,7) << Z(4,3) << Z(2,3);
pz-=Z(6,7); pz-=Z(2,7);
cout << "pocet " << pz << endl;
```

```
pz.prvni();  
do pz.aktual(); while (pz.dalsi());  
najit(pz,Z(1,4));  
najit(pz,Z(7,2));  
najit(pz,Z(1,2));
```

---

Soubor se zdrojovým kódem nebo celý projekt pošlete nejpozději v neděli 15. prosince. Při zasílání do předmětu dopisu uveďte zkratku semináře UP3CP nebo ZP3CP. Soubor s projektem nebo zdrojovým kódem bude mít jméno sestavené ze zkratky semináře, čísla úlohy, příjmení a křestního jména (příjmení bude před křestním jménem).

Příklad: UP3CP2MalýPavel.cpp , UP3CP2MalýPavel.rar .