

Sestavte šablonu třídy pro hašovací tabulku:

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

Parametry šablony:

- **T** - datový typ ukládaného prvku
- **n** - rozsah tabulky (počet řádků v tabulce)

Hašovací funkce bude mít jméno **h**, bude čistě virtuální a bude definována až v dědící třídě.

Operátory a funkce šablony **Hash**:

**virtual unsigned h(const T &) const =0;** Hašovací funkce.

**Hash & operator << (const T &);** Vloží prvek do tabulky. Pokud prvek se stejnou hodnotu je již v tabulce, vložení je ignorováno.

**unsigned operator + () const;** Vrací počet prvků, které jsou uloženy v tabulce.

**unsigned operator - () const;** Vrací počet prvků, které nebyly uloženy do tabulky, protože v tabulce už pro ně nebylo nalezeno volné místo.

**const T \* najit(const T &);** Hledá v tabulce prvek, který má stejnou hodnotu jako prvek uvedený v parametru. Pokud uložené prvky jsou objekty nějaké třídy, tato má pro tento účel přetížený operátor srovnání **==**, který srovná obsah objektů třídy prvků. Byl-li prvek v tabulce nalezen, operace vrací ukazatel na tento prvek, pokud nebyl nalezen, vrací nulový ukazatel (**nullptr**).

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 tabulce, a vrátí hodnotu *true*. Je-li tabulka prázdná, vrátí *false*.

**const T & aktual();** Vrací 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*.

---

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

Definujte šablonu pro hašovací tabulku pro zlomky děděním šablony **Hash** a v ní definujte hašovací funkci **h**.

```
template<unsigned n>
class HashZlomek;
```

Hašovací funkce pro zlomek:  $(31 \cdot \text{čitatel} + \text{jmenovatel}) \bmod \text{velikost\_tabulky}$

Začleňte do programu funkci:

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

Do hlavní funkce začleňte deklarace a příkazy:

```
HashZlomek<11> hz;
```

```

typedef Zlomek Z;

hz << Z(1,3) << Z(3,4) << Z(4,3) << Z(3,5) << Z(3,7)
    << Z(9,5) << Z(1,2) << Z(2,5) << Z(1,2);

cout << "V tabulce zlomku: " << +hz << endl;

hz.prvni();
do hz.aktual()(); while (hz.dalsi());

najit(hz,Z(3,7));
najit(hz,Z(2,5));
najit(hz,Z(9,5));
najit(hz,Z(2,3));

```

---

Volitelné rozšíření šablony **Hash**:

**bool operator -= (const T &);** Odstraní z tabulky prvek s hodnotou stejnou, jakou má prvek v parametru. Pro hledání prvku je použit operátor == pro srovnání hodnot prvků. Pokud uložené prvky jsou objekty nějaké třídy, tato má pro tento účel přetížený operátor srovnání ==, který srovná obsahy objektů třídy. Byl-li prvek odstraněn, výsledná hodnota je *true*, pokud nebyl prvek v tabulce nalezen, výsledná hodnota je *false*.

---

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: UP3CP1MalýPavel.cpp , UP3CP1MalýPavel.rar .