

2. Proseminář

Kolekce

Proudový vstup a výstup v C++ (opakování)

- Knihovny pro standardní vstupní a výstupní proudy jsou v hlavičkovém souboru **<iostream>**.
- Pro zápis slouží operátor **<<**.
- Pro čtení slouží operátor **>>**.
- Standardní proudy jsou:
 - **cin**,
 - **cout**,
 - **cerr**.

Řetězce v C++

- Nachází se v hlavičkovém souboru **<string>**

```
#include <string>
```

```
using std::string;
```

```
// nebo
```

```
using namespace std;
```

```
string s = "Ahoj";
```

Vector

- Datový typ reprezentující rostoucí kolekci se sekvenčním přístupem
- Nachází se v hlavičkovém souboru **<vector>**

```
#include <vector>  
using std::vector;
```

Vector (2)

- Využívá šablonování pro specifikaci typu ukládaných prvků
- Příklad:
`vector<int> vectorInt;`
- Inicializace:
 - `vector<T> v2(v1);` //inicializuje v2 obsahem v1
 - `vector<T> v3(n, i);` //v3 bude obsahovat n prvků s hodnotou i

Vector (3)

Operace

<code>v.empty()</code>	Returns <code>true</code> if <code>v</code> is empty; otherwise returns <code>false</code>
<code>v.size()</code>	Returns number of elements in <code>v</code>
<code>v.push_back(t)</code>	Adds element with value <code>t</code> to end of <code>v</code>
<code>v[n]</code>	Returns element at position <code>n</code> in <code>v</code>
<code>v1 = v2</code>	Replaces elements in <code>v1</code> by a copy of elements in <code>v2</code>
<code>v1 == v2</code>	Returns <code>True</code> if <code>v1</code> and <code>v2</code> are equal
<code>!=, <, <=, >, and >=</code>	Have their normal meanings

Procházení vektoru const_iterator

```
for(vector<int>::iterator i1 = v.begin(); i1 != v.end(); i1++)  
{  
    *i1 = 5; // ok  
    cout << *i1 << endl;  
}
```

```
for(vector<int>::const_iterator i1 = v.begin(); i1 != v.end();  
i1++) {  
    *i1 = 5; // chyba  
    cout << *i1 << endl;  
}
```

Set – uspořádaná množina

- Hlavičkový soubor **<set>**
- Prochází se stejně jako vektor
- Operace s :
 - `insert(item)` – vloží prvek do s
 - `count(item)` – vrátí 1 pokud $item \in s$, jinak 0

<http://cppreference.com>

Zadání prosemináře

- Program pro hledání prvočísel od 1 do N
- Vstup:
N
- Výstup:
 - všechna prvočísla v intervalu $\langle 1, N \rangle$ oddělená mezerou
- Příklad:
 - Vstup: 10
 - Výstup: 2 3 5 7

Zadání domácího úkolu

- Program pro hledání prvku metodou binárního půlení
- Vstup:
 - Sekvence celých čísel, oddělovač je mezera
 - Na druhém řádku číslo které se hledá
- Výstup:
 - 1 nebo 0, 1 – nalezeno, 0 – nenalezeno

Zadání domácího úkolu (2)

- Příklad:

- Vstup:

8 2 8 3 4 5 28 34 87 98

10

- Výstup:

0

Požadavky

Pro implementaci použijte kolekci *vector*.

Součástí odevzdání musí být funkční Makefile.

Vše naprogramujte do jednoho souboru.

Makefile musí správně reagovat na:

make

make run vstupni_soubor

make clean

Úlohu pojmenujte *XPJC_01_vas_fel_name.tar.gz*

Vyřešenou úlohu zašlete na:

cernyvi2@fel.cvut.cz

a odměna Vás nemine 😊

Do předmětu uveďte *XPJC 01 vas_fel_name*