

[prog2] Programming C++ (C6) Exercise Guide (Ex04)

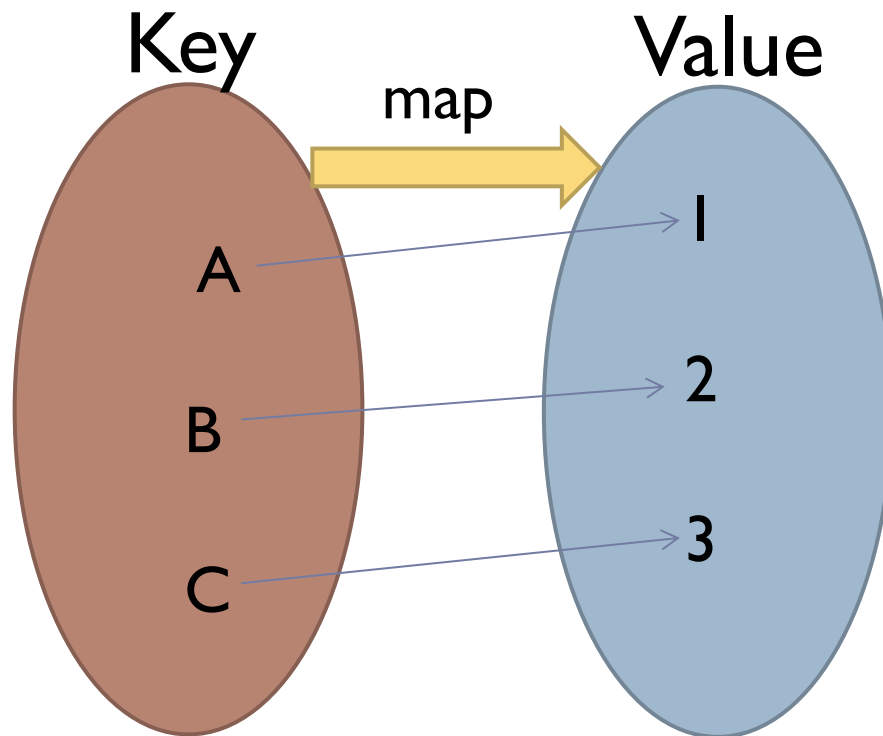
10/16, Monday 3rd period.

Ex04 について

▶ STLコンテナ

map キーと値の対になっている構造

- ▶ 数学的には「写像」
- ▶ アルゴリズム的には「ハッシュ」



連想コンテナ map (問題 1)

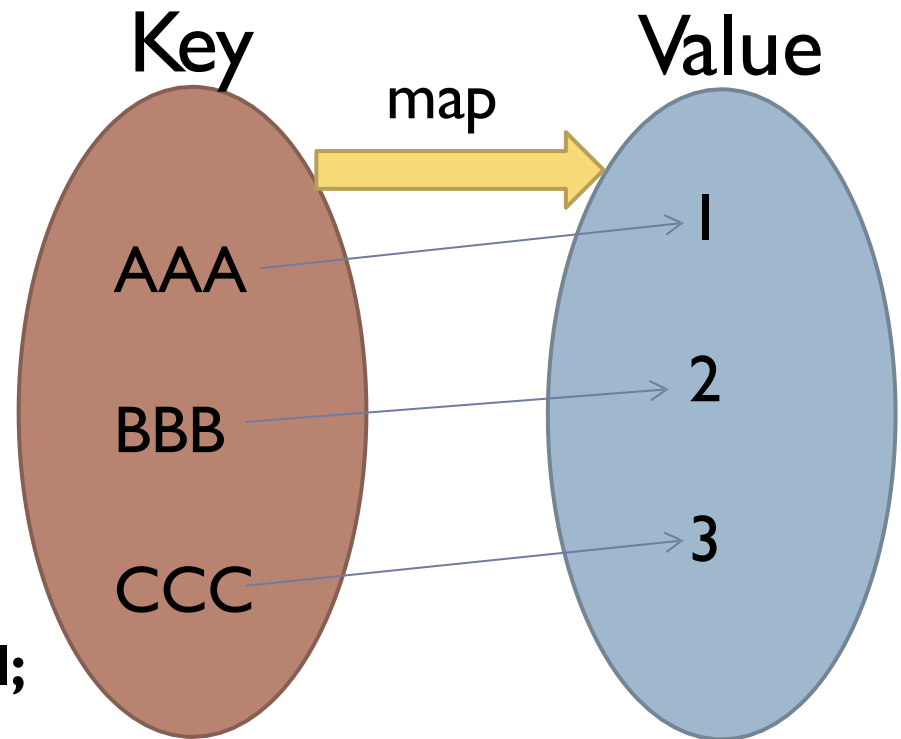
using namespace std;
を書かないなら、
std::map< ... , ... > となる

- ▶ (要インクルード) **#include <map>**
- ▶ **map<キーの型, 値の型> container;**

キーと値を対応付ける

map<string,int> ct;

- ▶ **ct["AAA"] = 1;**
- ▶ **ct["BBB"] = 2;**
- ▶ **ct["CCC"] = 3;**
- ▶ **cout << ct["AAA"] << endl;**
⇒ 1 が出力される



連想コンテナ map (問題 1)

map<string, int> ct;

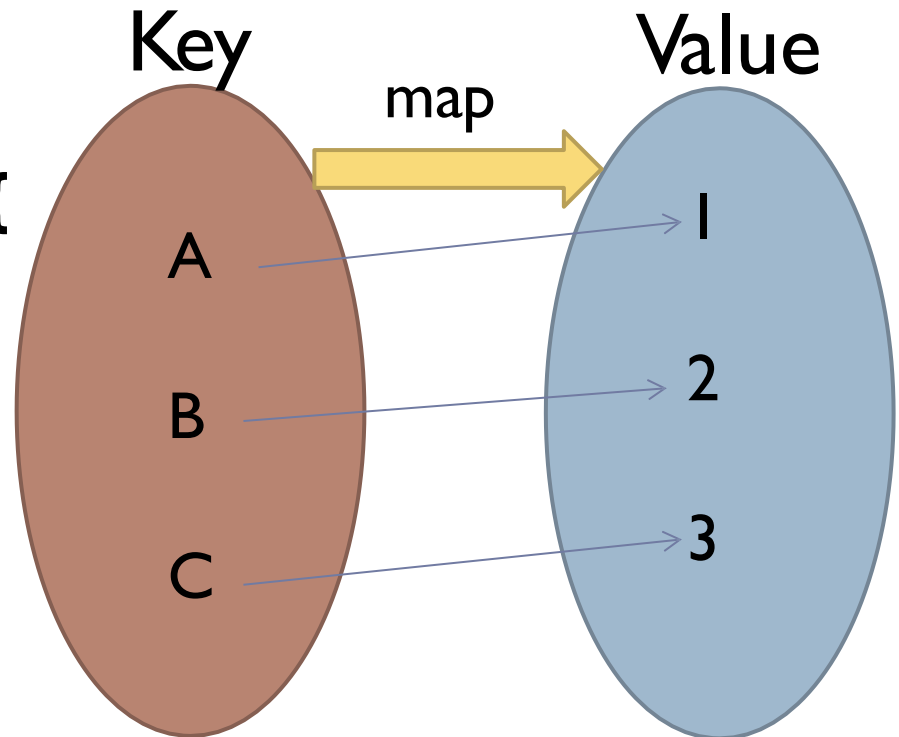
文字列をキーとして、整数の値に対応させる連想コンテナ

- ▶ **int val;**
- ▶ **string key;**
- ▶ **while(cin >> key >> val){**
- ▶ **ct[key] = val;**
- ▶ **}**

配列の形で

変数名[キー] = 値; とすると

その値を登録できる。登録後は、**変数名[キー]** で参照可能



map のイテレータ（問題 2）

- ▶ キーと値の対応関係は、配列の形で参照可能だが、mapには、キーと値の対応関係しかないので
`ct[0],ct[1],ct[2]` のようなランダムアクセスはできない

便宜上、イテレータを使えばmapの中身全体を参照可能
（使い方自体は、vectorやlistのイテレータと同じ）

```
map<string, int> ct;
```

```
map<string, int>::iterator itr;
```

```
for(itr = ct.begin(); itr!= ct.end();itr++){
```

```
    ...
```

```
}
```

map のイテレータ (問題 2)

- ▶ mapは、キーと値からなる構造である

⇒ イテレータが指す内容を取得する際には

`itr -> first;` `// first` ⇒ キー

`itr -> second;` `// second` ⇒ 値

のようにする

- ▶ `map<string, int> ct;`
- ▶ **`map<string, int>::iterator itr;`**
- ▶ **`for(itr = ct.begin(); itr!= ct.end();itr++){`**
 - ▶ `cout << itr-> first <<" " << itr->second << endl;`
 - ▶ `}`

map あれこれ（問題 2）

- ▶ キーを用いて値を参照する時、もしそのキーがmapに登録されていないならば、そのキーを新規に登録する。
⇒ 指定をしなければ、対応する値は0で初期化される
- ▶ map内のデータは、常に「キー」を基準にソートされている
⇒ イテレータで 先頭からデータ対を辿っていくと、キーがソートされた状態で順番に現れる
⇒ mapのキーに指定できる変数型は、順序関係が定義されているものに限る (int ⇒ 値が大きい、小さい, string ⇒ 文字のアルファベット順)
⇒ 例えば、構造体をキーにすることはダメ

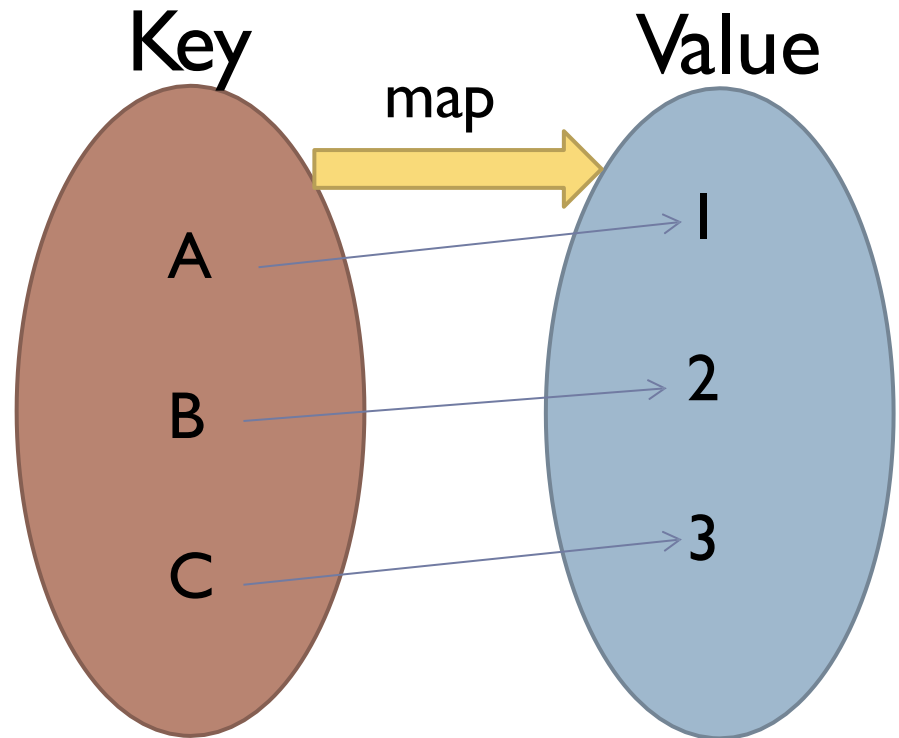
```
map< Student_info, int> data;
```

map あれこれ・2（問題3）

▶ mapの値に指定できるものは...？

⇒ map は キー1つに対して、値1つが対応する（全単射）

int や string といった型しか
値にできないか...？



map あれこれ・2（問題3）

構造体や、vector,list といったものは、

「複数のデータを保持する」1つの値として見なせる

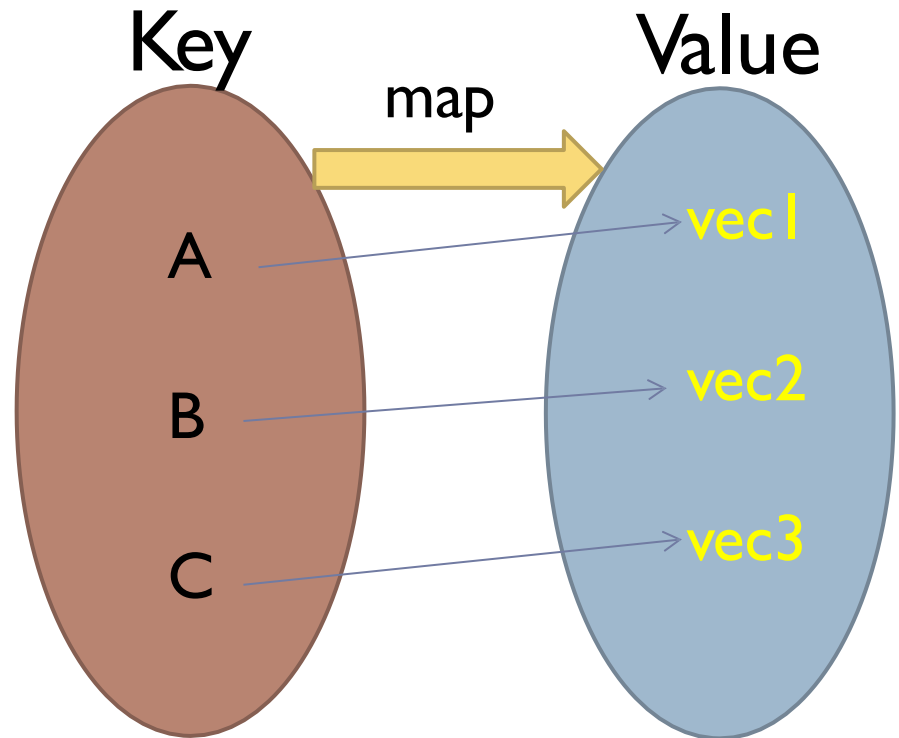
```
//キー Aの vectorコンテナに  
//値 x を挿入する。
```

```
ct["A"].push_back(x);
```

```
//キーに対応するvector参照は  
//vectorのイテレータで。
```

```
vector<int>::iterator itr;
```

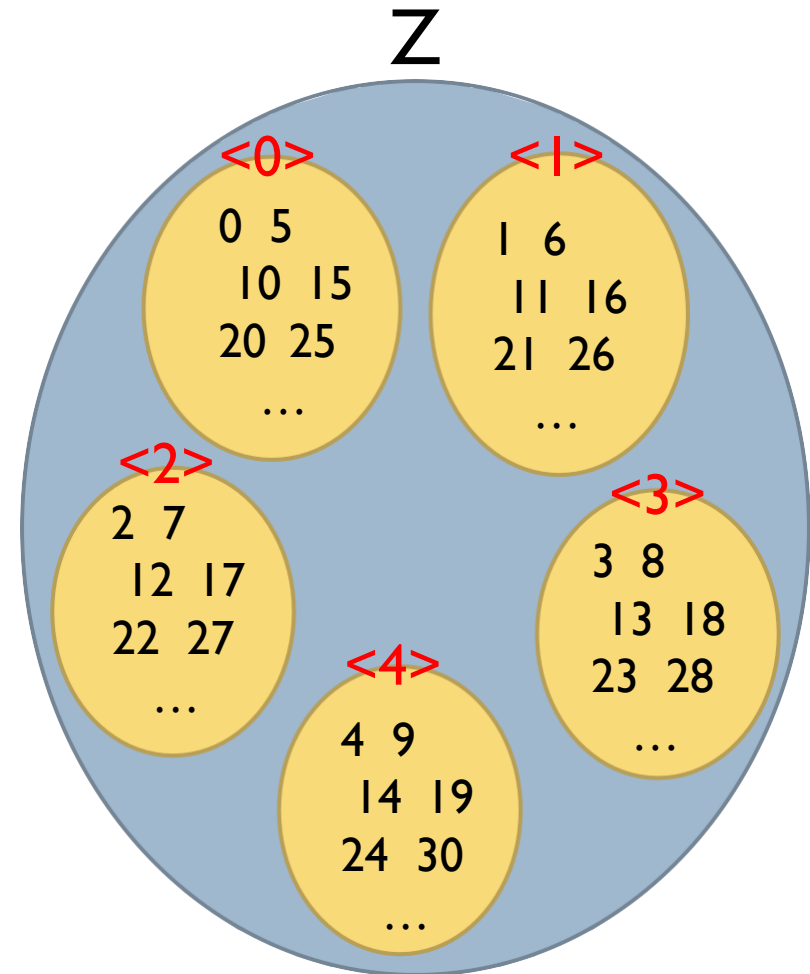
```
itr = ct["A"].begin();
```



問題 3 0 ~ 100 を n で割った余りで分類

- ▶ 剰余 n を入力
- ▶ 全ての整数 を n で割った余りの値に対応させる
- ▶ $n-1$ 個のグループ (キー)
- ▶ そのとき、余りが
入力値 m ($0 < m < n$) となるものを
全て出力する

1つのキーに対して、複数の値が対応
⇒ mapのキーを `int` 型、
値を `vector<int>` にする



例) $n = 5$ としたときの $\langle \text{キー} \rangle$ と 値たち の対応