

演習:map

- C++作業フォルダ内にPracMapフォルダを作成
`mkdir PracMap`
`cd PracMap`
- PracMapフォルダ内に main.cpp を作成する
`copy nul main.cpp`
- main.cpp をVisualStudioで開く

演習:map

PracMap(main.cpp)

```
#include<map>
#include<vector>
#include<iostream>
using namespace std;
typedef struct {    // 構造体Enemy
    string name;
    int hp, atk, def;
} Enemy;
int main() {

    return 0;
}
```

演習:map

•課題①

構造体Enemyを格納できるvectorの配列
vEneを宣言しなさい

vEneには初期値として以下のデータを格納する

Name:	Slime,	HP:	10,	Atk:	5,	Def:	8
Name:	Wolf,	HP:	20,	Atk:	30,	Def:	1
Name:	Spider,	HP:	30,	Atk:	30,	Def:	5

演習:map

PracMap(main.cpp)

```
#include<map>
#include<vector>
#include<iostream>
using namespace std;
typedef struct {    // 構造体Enemy
    string name;
    int hp, atk, def;
} Enemy;
int main() {
    // 構造体Enemyを管理するvector配列を宣言
    vector<Enemy> vEne{ { "Slime",10,5,8 },
        { "Wolf",20,30,1 }, { "Spider",30,15,5 } };
    return 0;
}
```

演習:map

•課題②

キーは文字列(`string`)、
値には構造体(`Enemy`)を格納するmapコンテナ
`mapEn`を宣言しなさい

ただし、`mapEn`は空のmapとする。

演習:map

PracMap(main.cpp)

```
typedef struct {    // 構造体Enemy
    string name;
    int hp, atk, def;
} Enemy;

int main() {
    // 構造体Enemyを管理するvector配列を宣言
    vector<Enemy> vEne{ { "Slime",10,5,8 },
                       { "Wolf",20,30,1 }, { "Spider",30,15,5 } };
    map<string, Enemy> mapEne{};
    return 0;
}
```

mapは基本データ型以外も格納可能

演習:map

- 課題③

mapEneのキーとして、エネミーの名前を格納し、
値にはパラメータ(Name, HP, Atk, Def)を格納する。

vEneに格納されているすべてのエネミーのデータを
mapEneに格納しなさい

演習:map

PracMap(main.cpp)


```
typedef struct {    // 構造体Enemy
    string name;
    int hp, atk, def;
} Enemy;

int main() {
    // 構造体Enemyを管理するvector配列を宣言
    vector<Enemy> vEne{ { "Slime", 10, 5, 8 },
                        { "Wolf", 20, 30, 1 }, { "Spider", 30, 15, 5 } };
    map<string, Enemy> mapEne{};
    for (const auto& d : vEne) { // 構造体を参照でdに格納
        mapEne.emplace(d.name, d);
    }
    return 0;
}
```


演習:map

•課題④

キーボードからエネミーの名前を入力し、**mapEne**から名前をキーとして値(Name,HP,Atk,Def)を画面表示するようにしなさい
(実行例)

エネミー名> Spider 

Name: Spider

HP: 30

Atk: 15

Def: 5

演習:map

PracMap(main.cpp)

```
vector<Enemy> vEne{ { "Slime",10,5,8 },
{ "Wolf",20,30,1 }, { "Spider",30,15,5 } };
map<string, Enemy> mapEne{};
for (const auto& d : vEne) { // 構造体を参照でdに格納
    mapEne.emplace(d.name, d);
}
cout << "エネミーの名前を入力>";
string input;
cin >> input;
if (mapEne.count(input)) {
    cout << "Name: " << mapEne[input].name << endl
        << "   HP: " << mapEne[input].hp << endl
        << "  Atk: " << mapEne[input].atk << endl
        << "  Def: " << mapEne[input].def << endl;
}
return 0;
}
```