JUTGE PRO2 FIB

3. Diccionaris i Conjunts

GitHub: https://github.com/MUX-enjoyer/PRO2-FIB-2025

Índex de Fitxers

Exercici: 3.1 Pairs

X16157 Divisió i Mòdul.cc

```cpp
#include <iostream>
#include <utility>
using namespace std;

pair<int, int> divmod(int a, int b) {
pair<int, int> parell_enters(a / b, a % b);
return parell_enters;
}

int main() {
int a, b;
cin >> a >> b;
pair<int, int> p = divmod(a, b);
cout << p.first << ' ' << p.second << endl;
}
```

Exercici: 3.1 Pairs

X18027 Llista de Parelles.cc

```cpp
#include <iostream>
#include <list>
#include <map>
using namespace std;

list<pair<string, int>> all_pairs(map<string, int>& M) {
list<pair<string, int>> llista_string_int;

map<string, int>::iterator it;
for (it = M.begin(); it != M.end(); ++it) {
pair<string, int> p = *it;
llista_string_int.push_back(p);
}
return llista_string_int;
}

int main() {
string s;
int n;
map<string, int> M;
while (cin >> s >> n) {
M[s] = n;
}
list<pair<string, int>> L = all_pairs(M);
list<pair<string, int>>::iterator it = L.begin();
for (; it != L.end(); it++) {
cout << it->first << ' ' << it->second << endl;
}
}
```

Exercici: 3.2 Maps

X34352 Freqüència de paraules amb diccionaris.cc

```cpp
#include <iostream>
using namespace std;

#include <string>
#include <map>

int main() {
map<string, int> partits;
string codi, paraula;
while(cin >> codi >> paraula) {
if (codi == "a") ++partits[paraula];
else if (codi == "f") cout << partits[paraula] << endl;
}
}
```

Exercici: 3.2 Maps

X47779 Morse.cc

```cpp
#include <iostream>
#include <map>
using namespace std;

string to_morse(string s, const map<char, string>& M) {
string morse;
for (int i = 0; i < s.size(); ++i) {
char lletra = toupper(s[i]);
if (lletra != ' ') {
 if (i != 0) morse.push_back(' ');
 morse.append(M.at(lletra));
 }
 }
 return morse;
 }

 int main() {
 map<char, string> M;
 M['A'] = ".-";
 M['B'] = "-...";
 M['C'] = "-.-.";
 M['D'] = "-..";
 M['E'] = ".";
 M['F'] = "..-.";
 M['G'] = "--.";
 M['H'] = "....";
 M['I'] = "..";
 M['J'] = ".---";
 M['K'] = "-.-";
 M['L'] = ".-..";
 M['M'] = "--";
 M['N'] = "-.";
 M['O'] = "---";
 M['P'] = ".--.";
 M['Q'] = "--.-";
 M['R'] = ".-.";
 M['S'] = "...";
 M['T'] = "-";
 M['U'] = "..-";
 M['V'] = "...-";
 M['W'] = ".--";
 M['X'] = "-..-";
 M['Y'] = "-.--";
 M['Z'] = "--..";
 M['0'] = "------";
 M['1'] = ".-----";
 M['2'] = "..----";
 M['3'] = "...--";
 M['4'] = "....-";
 M['5'] = ".....";
 M['6'] = "-....";
 M['7'] = "--...";
 M['8'] = "---..";
 M['9'] = "----.";

```

```
56 string s;
57 getline(cin, s);
58 cout << to_morse(s, M) << endl;
59 }
```

Exercici: 3.2 Maps

X79905 Freqüència de paraules amb diccionaris (amb esborrat).cc

```cpp
#include <iostream>
using namespace std;

#include <string>
#include <map>

int main() {
map<string, int> partits;
string codi, paraula;
while(cin >> codi >> paraula) {
if (codi == "a") ++partits[paraula];
else if (codi == "e" && partits[paraula] > 0) --partits[paraula];
else if (codi == "f") cout << partits[paraula] << endl;
}
}
```

Exercici: 3.3 Conjunts

X83904 Activitats esportives (sets).cc

```cpp
#include <iostream>
using namespace std;

#include <string>
#include <set>

int main() {
set<string> activitat, totes_activitats, cap_activitat;
set<string>::iterator it_act, it_tot, it_cap;

string cognom;
cin >> cognom;
while (cognom != ".") {
cap_activitat.insert(cognom);
cin >> cognom;
}

int n;
cin >> n;
totes_activitats = cap_activitat;
for (int i = 0; i < n; ++i) {
cin >> cognom;
while (cognom != ".") {
activitat.insert(cognom);
cin >> cognom;
}
for (it_act = activitat.begin(); it_act != activitat.end(); ++it_act) {
it_cap = cap_activitat.find(*it_act);
if (it_cap != cap_activitat.end()) cap_activitat.erase(it_cap);
}
for (it_tot = totes_activitats.begin(); it_tot != totes_activitats.end();;) {
it_act = activitat.find(*it_tot);
if (it_act == activitat.end()) it_tot = totes_activitats.erase(it_tot);
else ++it_tot;
}
activitat.clear();
}

cout << "Totes les activitats:";
for (it_tot = totes_activitats.begin(); it_tot != totes_activitats.end();
++it_tot) {
cout << ' ' << *it_tot;
}
cout << endl << "Cap activitat:";
for (it_cap = cap_activitat.begin(); it_cap != cap_activitat.end(); ++it_cap)
{
cout << ' ' << *it_cap;
}
cout << endl;
}
```