

[Dashboard](#) / [My courses](#) / [ITB_IF2210_2_1819](#) / [Ujian Tengah Semester](#) / [Ujian Tengah Semester Praktikum](#)Question **3**Tries remaining:
1Marked out of
100

Time limit	1 s
Memory limit	64 MB

Diberikan *template* kelas **entry** dalam file [entry.h](#).

Buatlah *template* kelas **dictionary** yang dapat berisi sekumpulan entry. Simpan dalam file **dictionary.h**. Berikut contoh penggunaan *template* kelas dictionary. Implementasi *template* harus mencakup semua [fungsi anggota](#) (publik) yang pada contoh di bawah ditandai dengan cetak tebal dan garis bawah.

Definisikan *exception* jika diminta.

```
int main() {
    dictionary<int,string> d; // dapat berisi sekumpulan entry<int,string>

    try {
        d.put(1,"one"); // isi d saat ini: [1:one] (menyimpan entry(1,"one") ke d)
        cout<<1<<": "<<d.get(1)<<endl; // mencetak "1: one" (mengambil value entry yang memiliki key 1)

        d.put(2,"two"); // isi d saat ini: [1:one,2:two]
        cout<<1<<": "<<d.get(1)<<endl; // mencetak "1: one"
        cout<<2<<": "<<d.get(2)<<endl; // mencetak "2: two"

        d.put(1,"satu"); // isi d saat ini: [1:satu,2:two], (menimpa value dari key 1 dengan "satu")
        // BUKAN [1:one,2:two,1:satu]
        d.put(3,"three"); // isi d saat ini: [1:satu,2:two,3:three]
        cout<<1<<": "<<d.get(1)<<endl; // mencetak "1: satu"
        cout<<2<<": "<<d.get(2)<<endl; // mencetak "2: two"
        cout<<3<<": "<<d.get(3)<<endl; // mencetak "3: three"

        d.remove(2); // isi d saat ini: [1:satu,3:three] (menghapus entry yang memiliki key 2)
        cout<<1<<": "<<d.get(1)<<endl; // mencetak "1: satu"
        if (d.containsKey(2)) { // (mengembalikan true jika d memiliki entry dengan key 2)
            cout<<2<<": "<<d.get(2)<<endl; // tidak ada output
        }
        cout<<2<<": "<<d.get(2)<<endl; // mencetak "exception: key not found!" (key 2 tidak ada di d)
        cout<<3<<": "<<d.get(3)<<endl; // tidak ada output karena sudah exception
    } catch (const char* c) {
        cout<<"exception: "<<c<<endl;
        return 1;
    }
    return 0;
}
```

Submit file **dictionary.h**.

Petunjuk: Secara internal, Anda boleh menentukan sendiri bagaimana kumpulan entry disimpan (*array*, *std::vector*, dsb.). jika menggunakan *vector* Anda tidak perlu memikirkan penggeseran elemen *array* bilamana terdapat elemen yang dihapus di tengah *array* (gunakan fungsi *erase()*).

C++11

Maximum size for new files: 512MB, maximum attachments: 1

[Files](#)

