

```

#include <iostream>
#include <string>
#include <vector>
using namespace std;
class Etudiant{
    protected :
        string nom;
        string prenom;
        int age;
        float moyenne;
    public :
        Etudiant(string n,string p,int a,float m):nom(n),prenom(p),age(a),moyenne(m){ }
        virtual void afficher(){
            cout<<"[nom]=>"<<nom<<endl;
            cout<<"[prenom]=>"<<prenom<<endl;
            cout<<"[age]=>"<<age<<endl;
            cout<<"[moyenne]=>"<<moyenne<<endl;
        }
        string getNom(){
            return nom;
        }
        virtual ~Etudiant(){ }
};

class EtudiantInfo : public Etudiant{
    private :
        string specialite;
    public :
        EtudiantInfo(string n,string p,int a,float m,string s):Etudiant(n,p,a,m),specialite(s){ }
        void afficher() override{
            Etudiant::afficher();
            cout<<"[specialite]=>"<<specialite<<endl;
        }
};

template <typename T>

```

```

class Conteneur{
    private :
        vector<T *>tabconteneur;
    public :
        void ajouter(T* e){
            tabconteneur.push_back(e);
        }
        int supprimer(string nom){
            int i=0;
            for(auto &etu : tabconteneur){

                i++;
                if (etu->getNom()==nom){
                    tabconteneur.erase(tabconteneur.begin()+i-1);
                    return 1;
                }
            }
            return 0;
        }

        void afficher(){
            for(auto etu : tabconteneur){
                etu->afficher();
            }
        }

};

int main(){

    Etudiant* E1 = new Etudiant("kassimi","abderrahmane",21,20);
    Etudiant* E2 = new Etudiant("sabi","houssame",22,20);
    Etudiant* E3 = new EtudiantInfo("nekt","hamza",24,20,"VALO");
    Etudiant* E4 = new EtudiantInfo("boussoga","ziyad",21,20,"VALO");
    Conteneur<Etudiant >classE;

```

```

classE.ajouter(E1);
classE.ajouter(E2);
classE.ajouter(E3);
classE.ajouter(E4);
cout<<"----[affichier]----"<<endl;
classE.afficher();
cout<<"----[suprsion]----"<<endl;
cout<<"----[hamza]----"<<endl;
try {
if (classE.supprime("nekt") == 0) {
    throw runtime_error("not found");
} else {
    cout<<"suppresion done"<<endl;
    }
} catch (const exception& e) {
    cout << "exception : " << e.what() << std::endl;
}
cout<<"----[adam]----"<<endl;
try {
    if (classE.supprime("adam") == 0) {
        throw runtime_error("not found");
    } else {
        cout<<"suppresion done"<<endl;
    }
} catch (const exception& e) {
    cout << "exception: " << e.what() << std::endl;
}

cout<<"----[apres la suppression]----"<<endl;
classE.afficher();
}

```