

Question 1 :

```
//Valeur.h
class Valeur
{
    int code;
    char* libelle;
    float prix;
    int type;

public:
    Valeur();
    Valeur(int c, char* s=0, float f=0.0,
int i=0);
    int GetCode();
    char* GetLibelle();
    float GetPrix();
    void Affiche();
    virtual ~Valeur();
};
```

```
//Valeur.cpp
Valeur::Valeur()
{
    prix = 0.0;
    type=0;
    libelle = "";
    code=0;
}
Valeur::Valeur(int c, char* s, float f, int i)
{
    code =c;
    libelle=s;
    prix=f;
    type = i;
}
int Valeur::GetCode()
{
    return code;
}
char* Valeur::GetLibelle()
{
    return libelle;
}
float Valeur::GetPrix()
{
    return prix;
}
void Valeur::Affiche()
{
    cout <<code <<"-"<<libelle<<
    "-"<<prix<<"-"<<type<<"\n";
}
```

```
//Marche.h
#include "valeur.h"
#define MAX 10
class Marche
{
    Valeur tab_val[MAX];
    int nb_val;

public:
    Marche();
    bool Chercher_Valeur(int);
```

```

    bool Chercher_Valeur(char*);
    void Ajouter_Valeur(Valeur v);
    void Affiche();
    Valeur PlusHaut();
    Valeur PlusBas();
    virtual ~Marche();
};

//Marche.cpp
Marche::Marche()
{
    nb_val = 0;
}
bool Marche::Chercher_Valeur(int c)
{
    bool r = false;
    for (int i =0; i<nb_val; i++)
        if (tab_val[i].GetCode() == c )
            r = true;
    return r;
}
bool Marche::Chercher_Valeur(char* c)
{
    bool r = false;
    for (int i =0; i<nb_val; i++)
        if (tab_val[i].GetLibelle() == c )
            r = true;
    return r;
}
void Marche::Ajouter_Valeur(Valeur v)
{
    tab_val[nb_val++] = v;
}
void Marche::Affiche()
{
    for (int i =0; i<nb_val; i++)
        tab_val[i].Affiche();
}
Valeur Marche::PlusHaut()
{
    float max = 0.0;
    int max_index;
    for (int i =0; i<nb_val; i++)
        if (tab_val[i].GetPrix() > max)
            {
                max_index = i;
                max = tab_val[i].GetPrix();
            }
    return tab_val[max_index];
}

```

```

}
Valeur Marche::PlusBas()
{
    float min = 555555.0;
    int min_index;
    for (int i =0; i<nb_val; i++)
        if (tab_val[i].GetPrix() < min)
            {
                min_index = i;
                min = tab_val[i].GetPrix();
            }
    return tab_val[min_index];
}

```

Question 2 :

```

#include "marche.h"
void main(int argc, char* argv[])
{
    Marche M;
    Valeur V1(1, "nasdaq", 33.9, 1), V2(2, "cac40", 40.0, 1);
    M.Ajouter_Valeur(V1);
    M.Ajouter_Valeur(V2);

    int code;
    cin >> code;
    if (M.Chercher_Valeur(code))
        cout << "valeur trouvee";
    else
        cout<< "valeur non trouvee";

    Valeur pB = M.PlusBas();
    Valeur pH = M.PlusHaut();
    pB.Affiche();
    pH.Affiche();
}

```

Question 3 :

```

/Valeur.h
    bool operator <=(Valeur);
    bool operator >=(Valeur);

```

```

//Valeur.cpp
bool Valeur::operator <=(Valeur v)
{
    if (prix <=v.prix)
        return true;
    else
        return false;
}
bool Valeur::operator >=(Valeur v)
{
    if (prix >=v.prix)
        return true;
    else
        return false;}

```