

Programme « Série de pièces »

Les mesures d'une série de 250 pièces sont stockées dans un tableau nommé `cote[ ]` sous la forme d'un réel, de type `float`. Une pièce est déclarée valide si la valeur de sa cote est comprise entre la cote minimum et la cote maximum. Au tableau `cote[ ]` est associé un tableau `pieceValide[ ]`. Il renseigne sur la validité de chaque pièce de la série. Pour la pièce `N`, la valeur de sa cote est disponible à `cote[N]` et de sa validité est disponible à `pieceValide[N]`. Les informations sur la validité de la pièce sont de 3 sortes : pièce valide, pièce trop petite et pièce trop grande.

Les cotes minimum et maximum ont pour valeur :

COTE_MIN : 254,80

COTE_MAX : 255,35

Extrait du programme

```

1.  _____ COTE_MIN = _____ ;
2.  _____ COTE_MAX = _____ ;
3.  _____ cote[____]=_____ ;
4.  _____ pieceValide[____]=_____ ;

5.  int nbreCote=0 ;
6.  // Acquiert la mesures des cotes
7.  // Renseigne le tableau cote avec la valeur des mesures
8.  // Renseigne nbreCote avec le nombre de mesures acquises
9.  nbreCote=acquerirMesureCote(cote);
10. // Renseigne le tableau pieceValide cote
11. for(int i=0; i<nbreCote; i++){
12.     if(cote[i]< COTE_MIN)
13.         pieceValide[i]=1;
14.     else{
15.         if(cote[i]>COTE_MAX)
16.             pieceValide[i]=3;
17.         else
18.             pieceValide[i]=2;
19.     }
20. }
```

- Complétez les déclarations aux lignes 1 et 2.
- Complétez la déclaration à la ligne 3, afin de déclarer le tableau `cote` et qu'il soit initialisé avec la valeur 0
- Quelle est la valeur maximum que peut prendre `nbreCote` ?
- Combien de fois la boucle `for`, ligne 10, est-elle parcourue ?

Le fragment de code est exécuté. Un extrait de valeurs contenues dans le tableau `cote` à l'issue de l'acquisition est donné ci-dessous. Complétez, ci-dessous, les valeurs du tableau `pieceValide` en conséquence

index	180	181	182	183	184	185	186	187	188	189
cote[]	255,31	254,80	254,92	255,13	255,35	255,44	255,12	254,99	254,81	254,75

index	180	181	182	183	184	185	186	187	188	189
pieceValide[]										

La plus grande cote à partir du tableau `cote`

A l'issue de la séquence ligne 20, on veut connaître la valeur de la mesure de la cote la plus grande. A cette fin, le tableau `cote[ ]` est parcouru afin d'identifier la valeur de la mesure la plus grande.

Proposez le fragment de code qui réalise cette opération.

```
float mesureMaxi=0 ;
```

La valeur moyenne des mesures des cotes

A l'issue de la séquence ligne 20, on veut calculer la valeur moyenne des mesures des cotes. Proposez le fragment de code qui réalise cette opération.

```
float mesureMoyenne=0 ;
```

La plus grande cote à partir du tableau pieceValide

A l'issue de la séquence ligne 20, on souhaite connaître la valeur de la mesure de la cote la plus grande. A cette fin, le tableau `pieceValide` est parcouru pour identifier si la pièce est trop grande puis, dans le tableau `cote`, la valeur de la mesure est analysée pour savoir si elle est la plus grande.

Proposez le fragment de code qui réalise cette opération.

```
float measureMaxi=0 ;
```

Modification du fragment de programme

Le tableau est déclaré et initialisé avec la valeur 0. Une mesure ne peut pas prendre la valeur 0. A l'issue de l'acquisition, la dernière mesure contenue dans le tableau cote est donc suivie par la valeur 0. Complétez en conséquence le fragment de programme ci-dessous, à partir du point //1.

```

#define FIN_MESURE 0

int nbreCote=0 ;

// Acquiert la mesures des cotes
// Renseigne le tableau cote avec la valeur des mesures
acquerirCote(cote);

// Renseigne le tableau pieceValide et nbreCote
// avec le nombre de mesures contenues dans le tableau cote
//1

    if(cote[i]< COTE_MIN)
        pieceValide[i]=1;
    else{
        if(cote[i]>COTE_MAX)
            pieceValide[i]=3;
        else
            pieceValide[i]=2;
    }

```

Afficher les valeurs des cotes valides

A l'issue de la séquence ligne 20, on veut afficher, sur une ligne, la valeur de la mesure des 16 premières cotes valides. Proposez le fragment de code qui réalise cette opération.