Nom/prénom : no camipro :

Programmation 2017
Test intermédiaire – série A + Corr

Durée 45min.

Mettre votre nom/prénom/no camipro sur toutes les pages.

Aucun document, ni appareil électronique (calculette, natel, etc.) n'est autorisé.

Pas de feuille volante reprise, uniquement les feuilles distribuées seront reprises.

Si vous devez faire des hypothèses, notez-les.

Signature :

Post-it avec no de place

Nom/prénom : no camipro :

1. (10pts) La fonction **MaxSpeed()** retourne le module de la plus grande vitesse calculée à partir des *vectors* **Pos** représentant les positions successives en [m] et **Time** représentant les instants (en [s]) auxquels les mesures de positions ont été effectuées.
Le calcul de la vitesse se fait intervalle par intervalle. Il doit y avoir au minimum un intervalle pour pouvoir calculer la vitesse. Un intervalle de temps négatif ou nul n'est pas possible et doit être considéré comme une erreur. Un intervalle de position nul indique que la voiture est à l'arrêt, un intervalle de position négatif indique que la voiture recule. Rappel : *abs(x)* retourne la valeur absolue de x.
MaxSpeed() retourne **-1** en cas d'erreur, ou le module de la vitesse maximale autrement.
Les erreurs possibles sont : taille de **Pos** ≠ taille de **Time**,
Pos et/ou **Time** vide,
intervalle de temps nul (0[s]) ou négatif

Exemples

avec Pos = {3, 6, 9} et Time = {0, 5, 7} **MaxSpeed (Pos, Time)** retourne **1.5**
avec Pos = {3, 6} et Time = {5, 4} **MaxSpeed (Pos, Time)** retourne **-1.0**
avec Pos = {3} et Time = {5, 7} **MaxSpeed (Pos, Time)** retourne **-1.**

Ecrire la fonction **MaxSpeed** () suivante:

```
double MaxSpeed (vector <double> Pos, vector <double> Time)
```

qui retourne : -1 en cas d'erreur
le module de la vitesse maximale autrement

```
double MaxSpeed(vector <int> Pos,vector <int> Time){
    double max(-1), dT(0), dP(0);
    if ((Pos.size()==Time.size()) && (Pos.size()>1))
            for (int p(1);p<Pos.size();p++){
             dP=abs(Pos[p]-Pos[p-1]);
             dT=Time[p]-Time[p-1];
             if (dT<=0.0)  {
                   max=-1;
                   break;
               }
             if (dP/dT > max)
                        max= dP/dT;
             }
       return max;
   }
```

gest err -> 3
boucle/indice ok -> 3
find min/max -> 2
syntaxe + return + autre: 2
cout au lieu de return : -1
max en testant uniquement les 2 dernieres valeurs : -1
serie B, val départ vitesse pas OK : -1

Nom/prénom : no camipro :

2.(4pts) Qu'affiche la console après l'exécution du code ci-après?

```
int main() {
    int b(4),c(2);
    for (int b(0); b<4; b++){
        c+=(b==c);
        cout<<c<<endl;
    }
}
```

```
2
2
3
4
```

---

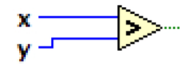
3. (4pts) Qu'affiche la console après l'exécution du code ci-après?

```
int myF1(int &A){
    int B(10);
    {
      int A(9);
      B=A/2;
    }
    return A=2*B;
    A++;
}


int main() {
 int A=3;

 int B = myF1(A);

 cout<<"A= "<<A<<", B= "<<B<<endl;

}
```

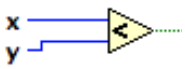
```
A= 8, B= 8
```

Nom/prénom : no camipro :

Rappel :

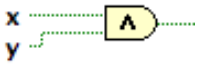


x>y ?



x<y ?

Stop if true

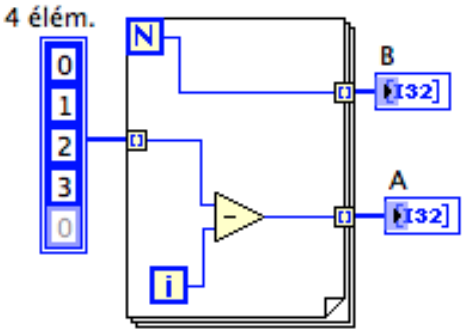


x && y

Continue if true

---

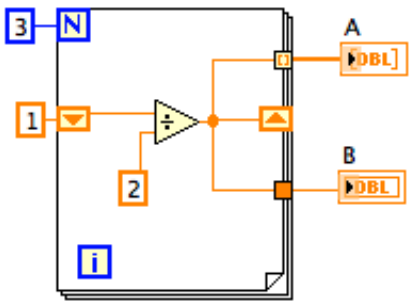
4. (4pts) Quelle est la valeur de *A et B* après l'exécution du *diagram* ci-dessous?



A=[0,0,0,0] B=[4,4,4,4]

---

5. (4pts) Que valent **A** et **B** après l'exécution du *diagram* ci-dessous?

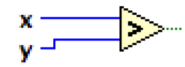


A=[0.5, 0.25, 0.125] B = 0.125

Nom/prénom : no camipro :

Note: pour les diagrams ci-après, tous les éléments des tableaux sont visibles ; le dernier '0' grisé ne fait pas partie du tableau.

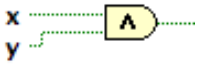
Rappel :



x>y ?

x<y ?

Stop if true

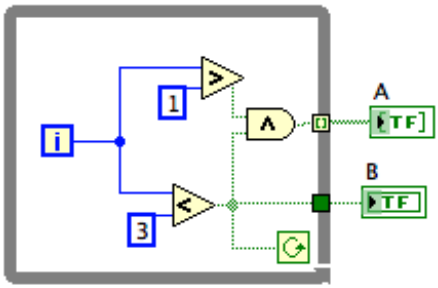


x && y

Continue if true

---

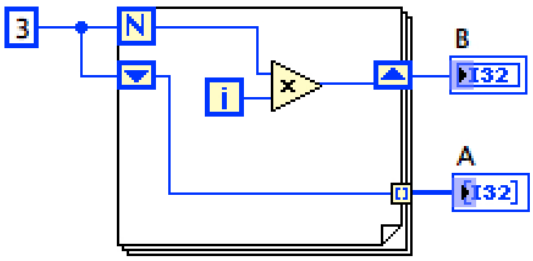
6. (4pts) Que valent **A** et **B** après l'exécution du *diagram* ci-dessous?



A=[F,F,T,F] B=F

---

7. (4pts) Que valent **A** et **B** après l'exécution du *diagram* ci-dessous?



A=[3, 0, 3] B= 6