Nom/prénom : no camipro :

Programmation 2018
Test intermédiaire – série 1A + 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.

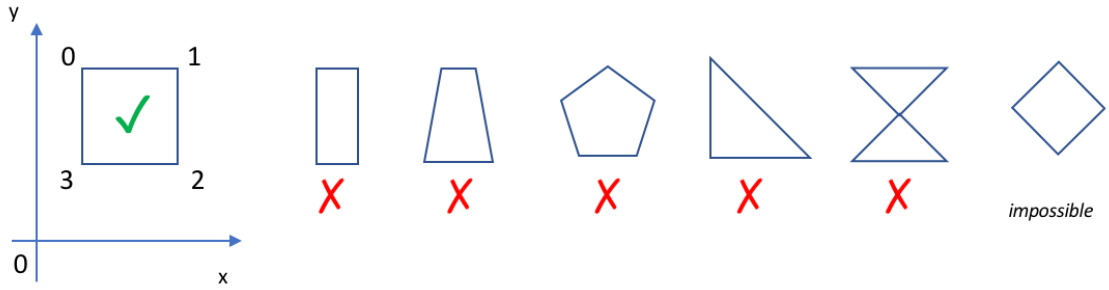
Signature :

Post-it avec no de place

Nom/prénom : no camipro :

1. (12pts) La fonction **EstUnCarre ()** teste si les éléments des vecteurs de coordonnées **X** et **Y** forment un carré ou non. Si c'est un carré, elle retourne la surface du carré, sinon 0 et un code d'erreur spécifique. **Le carré repose obligatoirement sur un de ses cotés, ne pas considérer le cas où il reposerait sur un de ses angles.**

L'ordre des coordonnées des coins du carré est le suivant :



Les erreurs retournées sont :
un carré (*pas d'erreur*) -> retourne 0
pas un carré, -> retourne -1
taille de **X** ≠ taille de **Y**, -> retourne -2

Exemples

avec X = {1, 3, 3, 1} et Y = {4,4,2,2}	**EstUnCarre (X,Y,S)** retourne **0** et **S**=4
avec X = {1, 3, 3, 1} et Y = {4,4,3,3}	**EstUnCarre (X,Y,S)** retourne **-1** et **S**=0
avec X = {1, 3, 3} et Y = {4,4,2}	**EstUnCarre (X,Y,S)** retourne **-1** et **S**=0
avec X = {1, 3, 3, 1} et Y = {4,4}	**EstUnCarre (X,Y,S)** retourne **-2** et **S**=0

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

```
int EstUnCarre(vector <double> X, vector <double> Y, double &S) ;
```

qui retourne : 0, -1 ou -2 en cas d'erreur
et la surface du carré

Contrainte: un seul *return* dans votre fonction.

Nom/prénom : no camipro :

```
int EstUnCarre(vector <double> X, vector <double> Y, double &S){
    S=0.0;      int err(0);
    if (X.size() != Y.size() ) err=-2;        // pb taille vect
      else if (X.size() !=4) err=-1;          // pas 4 cotés, triangle, pentagone, etc
        else {
          if ((X[0]!=X[3]) || (X[1]!=X[2])) err=-1; // coord X pas verticales
          if ((Y[0]!=Y[1]) || (Y[2]!=Y[3])) err=-1; // coord Y pas horizontales
          double dx(X[1]-X[0]), dy(Y[2]-Y[1]);
          if (fabs(dx) != fabs(dy)) err=-1;         // cotés de tailles différentes
          else S=fabs(dx*dy);
    }
    if (err!=0) S=0;
    return err;
}
```

Nom/prénom : no camipro :

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

```
int main() {

  int a(3),b(a);
  if (a=5) {a==6;cout<<a;}        // a=5 -> true -> cout<<a   5
                                  // a==6 ne fait rien
  else ;{ b+=(a-1);} ;            // else ; (rien)
                                  // b+=(a-1);  tjs exécuté
  cout<<"a"<<b;                   // b = b+a-1 -> 3+5-1 ->    7

  return 0;
}
```

5a7

---

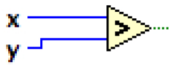
3. (4pts) Qu'affiche la console après l'exécution du code ci-après?
Justifiez votre réponse.

```
int myF1(int &a) {
    double b(a);
    a++;                    // incr &a et donc i boucle for
    return b+0.5;           // return int -> +0.5 sans effet
  }

int main() {

  for (int i(1); myF1(i)<6; i++)      // i incr 2x: boucle + myF1
       { cout<<i;continue; i++;}      // i++ jamais executé

  return 0;
}
```

246

Nom/prénom : no camipro :

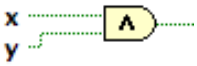
Rappel :



x>y ?

x<y ?

Stop if true

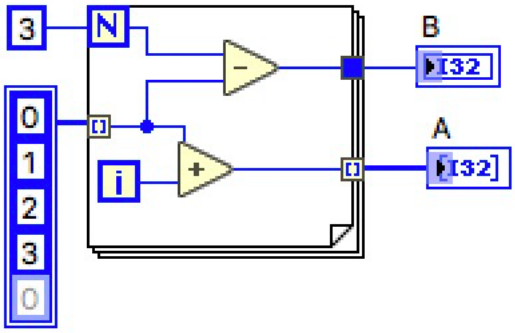


x && y

Continue if true

---

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

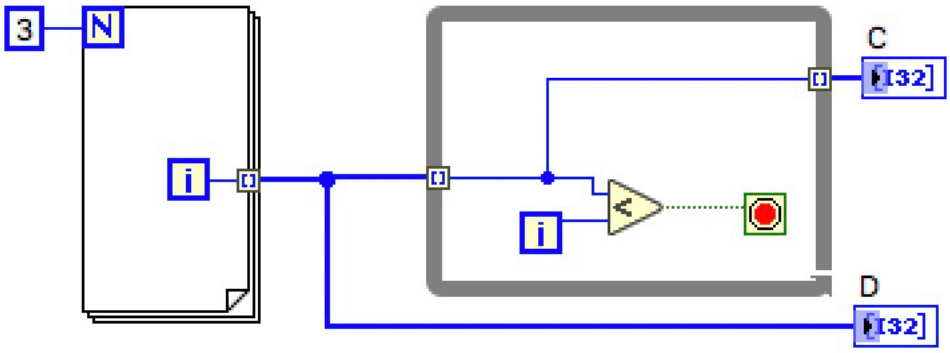


Plus petit de 3, size array (4) gagne -> 3x

B=1
A = [0, 2, 4]

---

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

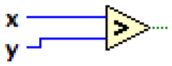


C = [0,1,2,*0*] *0 valeur par défaut, quand une valeur est nécessaire*
A = [0, 1,2]

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

Nom/prénom : no camipro :

Rappel :



x>y ?

not

Stop if true

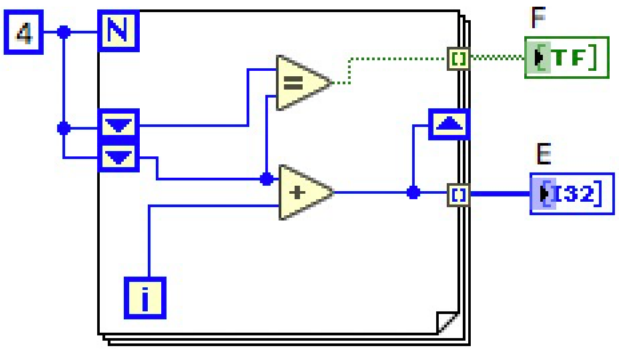
select

build array

Continue if true

---

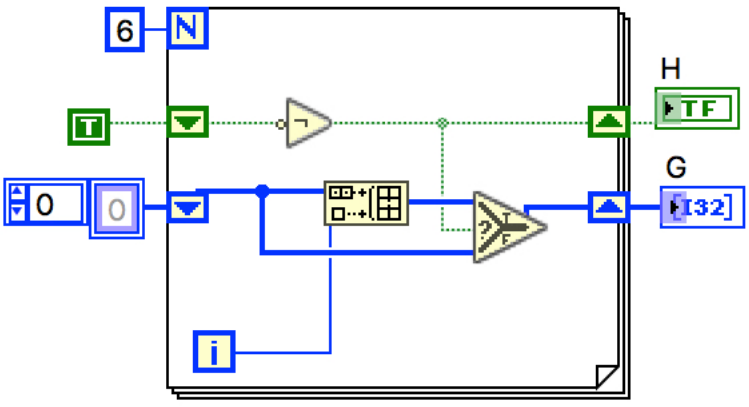
6. (4pts) Que valent **E** et **F** après l'exécution du *diagram* ci-dessous?



E = [4,5,6,8]
F = [T,T,F,F]

---

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



Ajoute ou non les valeurs de 'i' au tableau en fonction du boolean

G = [1,3,5]
H = T