

Nom/prénom :

no camipro :

# Programmation 2017

## Test final – série A + Coor

Durée 1h45.

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

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

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

Si vous devez faire des hypothèses, notez-les.

Merci de signer cette première page

Signature :

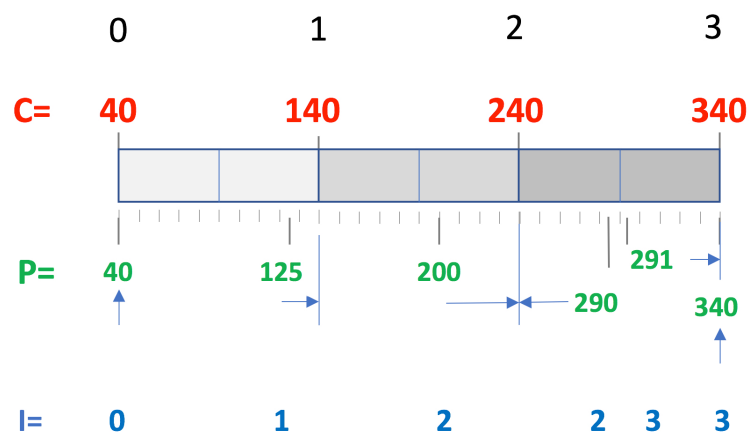
Post-it avec no de place

Nom/prénom :

no camipro :

1. (18 pts) La fonction **Compress()** encode un vecteur de valeurs **P** pour réduire sa taille en mémoire, à la manière des couleurs indexées pour le format PNG. Pour ce faire, dans un premier temps, la fonction choisit une séquence **C** de **n** valeurs croissantes, uniformément réparties entre la plus petite valeur de **P** et la plus grande valeur de **P**. La séquence **C** inclut la plus petite valeur de **P** et la plus grande valeur de **P**. Puis, pour chaque valeur de **P**, elle détermine la valeur correspondante **k** la plus proche dans **C** et mémorise l'index de cette valeur **k** dans le vecteur de valeurs compressées **I**.

La fonction **Compress ()** retourne -1 si **P** est vide ou si **n** est supérieur ou égal à la taille de **P**, ou si une valeur de **P** est inférieure à 0. Elle initialise et met à jour les vecteur **C** et **I** passés par référence.



Exemples :

avec  $n=4$  ; et

$P = \{ 125, 291, 340, 290, 40, 200 \}$  ;

**Compress**(**P**,**n**,**I**,**C**) retourne 0 et

$I = \{ 1, 3, 3, 2, 0, 2 \}$  et

$C = \{ 40, 140, 240, 340 \}$

125 est plus proche de 140, position de 140 dans C est **1**

291 est plus proche de 340, position de 340 dans C est **3**

avec  $n=8$  ; et  $P = \{ 125, 280 \}$  ; **Compress**(**P**,**n**,**I**,**C**) retourne -1

avec  $n=2$  ; et  $P = \{ 125, 280, -340 \}$  ; **Compress**(**P**,**n**,**I**,**C**) retourne -1

avec  $n=2$  ; et  $P = \{ \}$  ; **Compress**(**P**,**n**,**I**,**C**) retourne -1

Ecrire en C++ la fonction **Compress ()** suivante:

```
int Compress(vector <int> P, int n, vector <int> &C, vector <int> &I)
```

Met à jour les vecteur C et I

et retourne : -1 si **P** est vide

si **n** est supérieur ou égale à la taille de **P**,

si une valeur de **P** est inférieure à 0

0 autrement

Nom/prénom :

no camipro :

```
int Compress(vector <int> P, int n, vector <int> &C, vector <int> &I){
    C.clear();    I.clear();           // C & I may not be empty initially
    if (n>=P.size() || P.size()==0) return -1;
    double min(P[0]), max(P[0]);
    for (int i(1); i<P.size();i++) {    // find min & max of P
        if (min>P[i]) min=P[i];
        if (max<P[i]) max=P[i];
        if (P[i]<0) return -1;         // P[i] can't be <0
    }
    double s=(max-min)/(n-1);          // compute the interval size for C
    for (int i(0); i<n;i++)
        C.push_back(min+s*i);
    for (int i(0); i<P.size();i++) {
        I[i]=0;                        // initialize I to the 1st index
        for (int j(0); j<n-1;j++) {    // check all intervals
            if ((P[i]>=C[j])&&(P[i]<=C[j+1])) { // found the interval
                I.push_back(j+(abs(P[i]-C[j])>abs(P[i]-C[j+1]))); break;
            }
        }
    }
    return 0;
}
```

Nom/prénom :

no camipro :

---

2. (8 pts) **V** est une matrice de points qui contient dans la 1<sup>re</sup> colonne les coordonnées **X** et dans la 2<sup>e</sup> colonne les coordonnées **Y** de chaque point de **V**. Ecrire en matlab, **sans boucle**, la fonction **InvertUP(V)** qui retourne les points de **V** dont la coordonnée **Y** est plus grande ou égale à 0. A vous de définir le/les paramètre(s) de retour de la fonction.

```
function [Xout, Yout] = InvertUP(V)
    Xout=V(find(V(:,2)>0),1)
    Yout=V(find(V(:,2)>0),2);
end
```

---

3. (4 pts) Ecrire en matlab, **sans boucle**, la fonction anonyme **reverse(X)** qui inverse l'ordre les positions des éléments du vecteur X. Le dernier éléments de X devient le premier, l'avant dernier devient le deuxième et le premier élément devient de dernier.

Ex. si X= [8, 14, 2, -2, 5], **reverse (X)** retourne [5, -2, 2, 14, 8]

```
reverse=@(X) X([end:-1:1]);
```

---

4. (6 pts) Dans matlab, que valent L,M,M. **Les valeurs des variables se réfèrent aux valeurs des points précédents.**

```
L=linspace(6,1,6)
```

```
L = 6 5 4 3 2 1
```

```
M=L - (1:6)
```

```
M = 5 3 1 -1 -3 -5
```

```
M([4 :6])= []
```

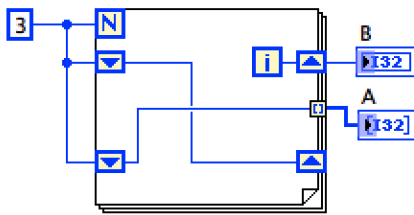
```
M = 5 3 1
```

Nom/prénom :

no camipro :

---

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

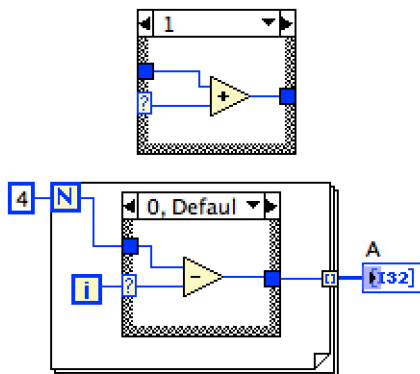


$A = [3 \ 3 \ 0]$     $B = 2$    (a, c)

$A = [0 \ 0 \ 3]$     $B = 3$    (b, d)

---

6. (4 pts) Que vaut **A** après l'exécution du *diagram* ci-dessous, justifiez?

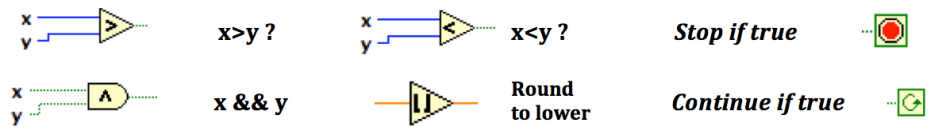


$A = [4 \ 5 \ 2 \ 1]$    (a, c)

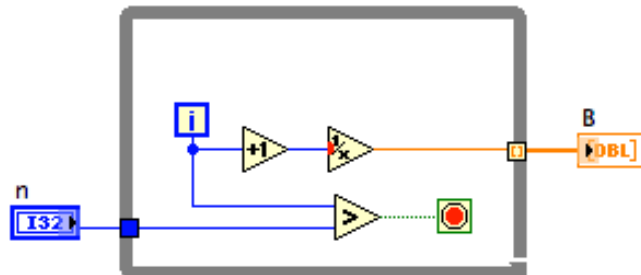
$A = [4 \ 4 \ 2 \ 1]$    (b, d)

Nom/prénom :

no camipro :



7. (6 pts) a) Si  $n$  vaut 3, que vaut  $B$  après l'exécution du *diagram* ci-dessous?



$$B = [1/1, 1/2, 1/3, 1/4]$$

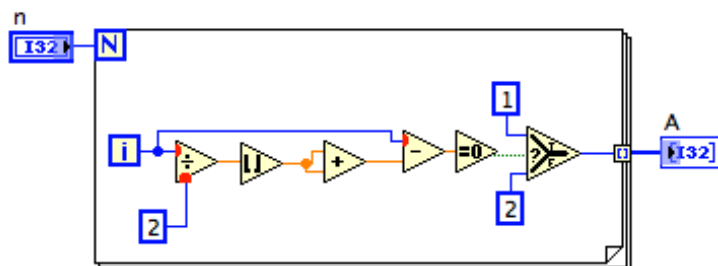
$$B = [1 \ 4 \ 9 \ 16]$$

b) Ecrire en matlab (1 ligne) les codes équivalents aux *diagram*s ci-dessus pour  $B$  fonction de  $n$

$$B = 1./(1:n+1)$$

$$B = (1:n+1).^2$$

8. (6 pts) a) Si  $n$  vaut 5, que vaut  $A$  après l'exécution du *diagram* ci-dessous ?



$$A = [1 \ 2 \ 1 \ 2 \ 1]$$

$$A = [4 \ 0 \ 4 \ 0 \ 4]$$

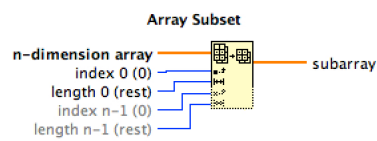
b) Ecrire en matlab (1 ligne) les codes équivalents aux *diagram*s ci-dessus pour  $A$  fonction de  $n$

$$A = 2 - \text{mod}([1:n], 2)$$

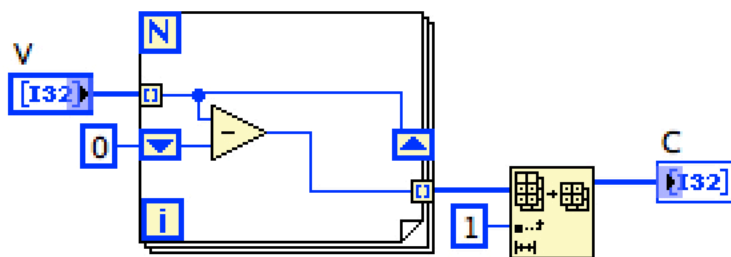
Nom/prénom :

no camipro :

$$A=4*\text{mod}([1:n],2)$$



9. (6 pts) Si  $V$  vaut  $[2\ 4\ 9\ 16]$ , que vaut  $C$  après l'exécution du *diagram* ci-dessous?

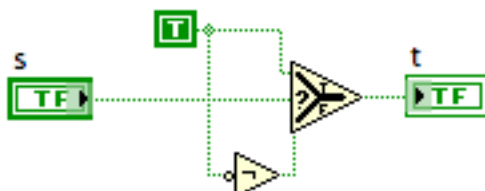


$$C = [2, 5, 7]$$

b) Ecrire en matlab (1 ligne) les codes équivalents aux *diagram*s ci-dessus pour  $V$  fonction de  $C$

$$C = \text{diff}(V)$$

10. (2 pts) Comment simplifier le *diagram* ci-dessous ? Dessinez la solution.



$$S[\text{TF}] \text{-----} [\text{TF}]t$$