

Nom/prénom :

no camipro :

ME-213, Programmation pour ingénieur

Test Final 2023 Fall + corr

Durée 1h45min.

0

Veillez signer cette page.

Aucun document, ni appareil électronique (calculatrice, smartphone, etc.) n'est autorisé. Vous pouvez utiliser un crayon et/ou un stylo et répondre en français ou en anglais. Vos réponses doivent être lisibles ! Note : Il n'y a pas deux copies identiques !

Aucune feuille volante n'est reprise, uniquement les feuilles distribuées seront reprises. Vous pouvez utiliser le verso des pages comme brouillon. La dernière page est une page blanche.

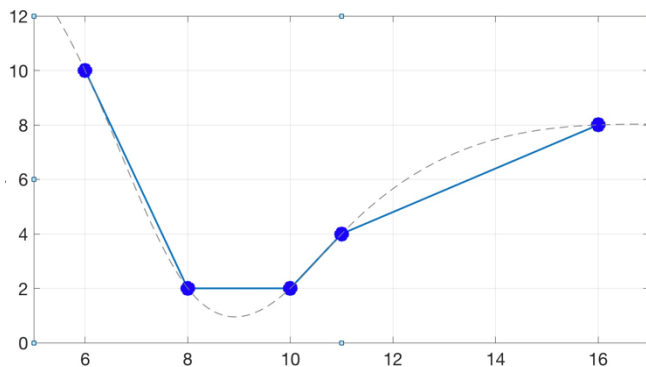
Si vous devez faire des hypothèses, notez-les.

place:

Signature :

--	--	--	--	--	--	--	--

1. (20 pts) Le vecteur **Y** contient les valeurs d'une fonction f , évaluée aux points définis par le vecteur **X**. Pour la figure ci-dessous :



$$\mathbf{Y} = [10, 2, 2, 4, 8]$$

$$\mathbf{X} = [6, 8, 10, 11, 16]$$

La fonction `myDiff(szX, X, szY, Y, D)` calcul les approximations des pentes successives ($\Delta Y / \Delta X$) de f et met le résultat dans **D**.

- Si le vecteur **X** est vide ($\mathbf{szX}=0$), la fonction `myDiff(szX, X, szY, Y, D)` fonctionne comme la fonction matlab `diff(Y)` et l'espace (ΔX) entre les points est l'unité (1).
Pour l'exemple ci-dessus, avec $\mathbf{szX}=0$ et **Y** (figure) $\rightarrow \mathbf{D} = [-8, 0, 2, 4]$
- Si le vecteur **X** contient uniquement une valeur ($\mathbf{szX}=1$) et $\mathbf{X}[0] > 0$ cette valeur définit l'espace (ΔX) entre les points.
Pour l'exemple ci-dessus, avec $\mathbf{X} = [2]$ et **Y** (figure) $\rightarrow \mathbf{D} = [-4, 0, 1, 2]$
- Dans les autres cas, le vecteur **X** contient les points auxquels la fonction f est évaluée. Dans ce cas, les vecteurs **X** et **Y** doivent avoir la même taille ($\mathbf{szX}=\mathbf{szY}$).
Pour l'exemple ci-dessus, avec **X** (figure) et **Y** (figure) $\rightarrow \mathbf{D} = [-4, 0, 2, 4/5]$

La fonction `myDiff()` retourne 0 ou les erreurs suivantes:

- -1, si la taille de **Y** (\mathbf{szY}) est plus petite que 2
- -2, si **X** contient une seule valeur qui est ≤ 0
- -3, si la taille de **X** (\mathbf{szX}) est plus grande que 1 et plus petite que la taille de **Y**,
ou si les tailles de **X** et **Y** sont différentes quand $\mathbf{szX} > 1$
- -4, si les différences entre les valeurs consécutives de **X** ne sont pas dans un ordre strictement croissant (>0), *i.e.* $\mathbf{X}[i+1] > \mathbf{X}[i]$

Le vecteur **D** à la même taille que le vecteur **Y**, il n'est pas forcément vide lors de l'appel à `myDiff()`. En cas d'erreur, le vecteur **D** doit être initialisé à '0'.

Ecrire en C la fonction :

```
int myDiff(int szX, double X[], int szY, double Y[], double D[] )
```

avec

X et **Y** les vecteurs définissant l'approximation de la fonction f .
szX et **szY** les tailles de vecteurs **X** et **Y**
D contient les valeurs calculées ou est rempli de 0 en cas d'erreur.
`myDiff()` retourne 0 ou un code d'erreur.

```
int myDiff(int szX, double X[], int szY, double Y[],double D[] )
```

```
// same as ML diff(),
// if Xsz = 0 -> delta= 1, if Xsz = 1 -> delta= X[0]
// => D[] =  $\Delta Y/\Delta X$  or D[] =  $\Delta Y/\text{delta}$ 
// err
// -1) Ysz <= 1
// -2) delta <= 0
// -3) Xsz != Ysz or Xsz = 1 or Xsz =0
// -4) X[n+1] <= X[n]
```

```
int myDiff(const double Y[], int szY, const double X[], int szX,
double D[] ) {
    int err=0;
    double deltaX =0.0;

    if (szY>1) { // more than 1 val in Y
        if (szX ==0) //  $\Delta X = 1$ 
            deltaX=1.0;
        else if (szX ==1) { //  $\Delta X = X[0]$ 
            if (X[0] > 0.0) deltaX = X[0];
            else err = -2; }
        else if (szX==szY){ //  $\Delta X = X[i] - X[i-1]$ 
            for (int i=0;i<szX-1;i++) {
                if (X[i] >= X[i+1]) // non increasing order !
                    err=-4;
            }
        }
        else err = -3;

    if (err==0) {
        for (int i=0;i<szY-1;i++)
            if (deltaX==0.0)
                D[i]=(Y[i+1]-Y[i])/(X[i+1]-X[i]);
            else
                D[i]=(Y[i+1]-Y[i])/deltaX;
        }
    }
    else
        err=-1;

    return err;
}
}
```

2. (4 pts) En C, que vaut n après l'appel à `myF()`? Justifiez !

```
int myF(int E[]) {
    return sizeof(E)/sizeof(E[0]);
}
int E[] = {4,5,6};
int n = myF(E);
```

`E` est un pointeur dont la taille est toujours 8 (64 bits OS)

$n = 8(\text{ptr size in 64bit system}) / 4 (\text{size of int}) = 2$

3. (4 pts) Ecrire en matlab, **sans boucle**, la fonction anonyme `RemoveN Flip(X,n)` qui retourne l'ordre des valeurs inversé de le série fournie.

Ex. si $X = [1\ 2\ 3\ 4]$, `FlipH(X)` retourne $[4\ 3\ 2\ 1]$

`FlipH = @(V) V([end:-1:1]);`

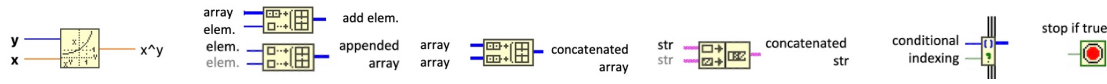
4. (4 pts) Ecrire en matlab, **sans boucle**, la fonction anonyme `RemoveN(X,n)` qui retourne les éléments du vecteur X sans les valeurs qui sont un multiple de n .

Ex. si $X = [1\ 2\ 3\ 4\ 5\ 6]$, `RemoveN(X,3)` retourne $[1\ 2\ 4\ 5]$

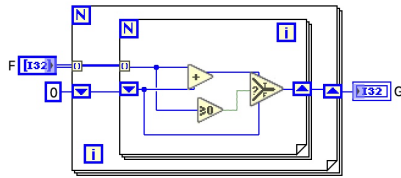
`RemoveN = @(X,n) X(find(mod(X,n)~=0))`

Nom/prénom :

no camipro :



5a. (2 pts) Si F vaut $[-3 \ 2; -1 \ 5]$, que vaut G après l'exécution du *diagram* ci-dessous.

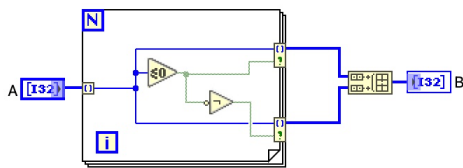


$$G = 7$$

5b. (2 pts) Ecrire en matlab (1 ligne, sans boucle for) le code équivalent au *diagram* ci-dessus pour G fonction de F

$$G = \text{sum}(F(F \geq 0))$$

6a. (2 pts) Si A vaut $[3, -2, 1, -4]$, que vaut B après l'exécution du *diagram* ci-dessous.



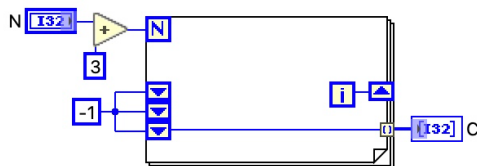
$$B = [-2, -4, 3, 1]$$

2D .. -4 pts

6b. (2 pts) Ecrire en matlab (1 ligne, sans boucle for)) le code équivalent au *diagram* ci-dessus pour B fonction de A

$$B = [A(\text{find}(A \leq 0)), A(\text{find}(A > 0))]$$

7a. (2 pts) Si $N = 3$, que vaut C après l'exécution du *diagram* ci-dessous.



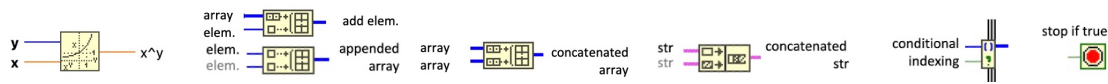
$$C = [-1, -1, -1, 0, 1, 2]$$

7b. (2 pts) Ecrire en matlab (1 ligne, sans boucle for)) le code équivalent au *diagram* ci-dessus pour C fonction de N

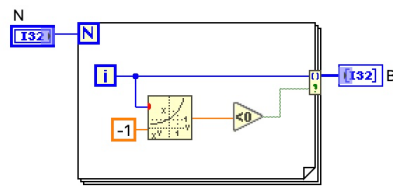
$$C = [-1 \ -1 \ -1 \ 0:n-1]$$

Nom/prénom :

no camipro :



8a. (2 pts) Si $N = 10$, que vaut B après l'exécution du *diagram* ci-dessous.

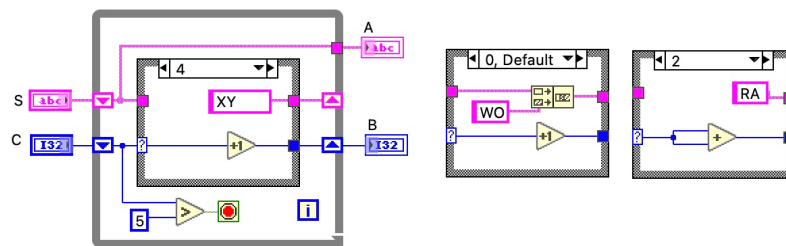


$$B = [1 \ 3 \ 5 \ 7 \ 9]$$

8b. (2 pts) Ecrire en matlab (1 ligne, sans boucle for)) le code équivalent au *diagram* ci-dessus pour B fonction de N .

$$B = \text{find}((-1 \wedge n) .* 1:n)$$

9. (6pts) Pour la machine d'état ci-après, si S vaut SU et C vaut 3 , que valent A et B après l'exécution du *diagram* ci-dessous ? Montrez vos étapes intermédiaires



$$A = XYWO (3) - SUW0 (5)$$

$$B = 7$$

Nom/prénom :

no camipro :

Feuille de brouillon