Nom/prénom : no camipro :

ME-213, Programmation pour ingénieur
Test intermédiaire 2023 fall + corr

Durée 45min.

Merci de signer cette page.

Aucun document, ni appareil électronique (calculette, smartphone, etc.) n'est autorisé. Vous pouvez utiliser un crayon et/ou un stylo et répondre en français ou 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.

Si vous devez faire des hypothèses, notez-les.

Signature :

Nom/prénom : no camipro :

1a. (15 pts) Écrire en C la fonction **`int PrintSubStr1(char* S, int from, int to)`** qui affiche les caractères de **S** des positions **from** à **to**, positions from et to incluses. Les caractères sont affichés dans la console à l'aide de **`putchar(char c)`**
Si le caractère de fin de chaine '\0' est rencontré avant ou à la position **to,** ne pas afficher les suivants et retourner l'erreur -1.
Si le caractère de fin de chaine '\0' est rencontré avant la position **from,** ne rien afficher et retourner l'erreur -2.
Si **from** < **0** ou **to < 0**, **`PrintSubStr1()`** retourne -3 et ne produit pas d'affichage;
Si **from** > **to**, **`PrintSubStr1()`** retourne -4 et ne produit pas d'affichage;
S est une chaine valide, i.e. non NULL, pas nécessaire de tester **S**.

Exemples
`PrintSubStr1`("Hello\n", 1,3) retourne **0** et affiche **ell**
`PrintSubStr1`("Hello\n ", 0,9) retourne **-1** et affiche **Hello\n**
`PrintSubStr1`("Hello\n ", 0,6) retourne **-1** et affiche **Hello\n**
`PrintSubStr1`("Hello\n ", 7,9) retourne **-2** et affiche *-rien-*
`PrintSubStr1`("Hello\n", -1,3) retourne **-3** et affiche *-rien-*
`PrintSubStr1`("Hello\n ", 3,1) retourne **-4** et affiche *-rien-*
`PrintSubStr1`("", 0,0) retourne **0** et affiche *-rien-*

Écrire la fonction **`PrintSubStr1`**() suivante:

```
int PrintSubStr1(char* S, int from, int to)
```

qui retourne : -1 si '\0' est rencontré avant la position **`to`**
-2 si '\0' est rencontré avant la position **`from`**
-3 si **`from`** <0 ou si **`to`** <0
-4 si **`from`** > **`to`**
0 autrement

```
int PrintSubStr(char *S, int from, int to){
    if (from < 0)      return -3;
    if (to < 0)   return -3;
    if (from > to)     return -4;

    for (int i=0; i< from ; i++)
        if (S[i]=='\0')
                       return -2;

    for (int i=from; i<= to ; i++) {
        if (S[i]=='\0')
                       return -1;
        putchar(S[i]);
    }
    return 0;
}
```

Nom/prénom : no camipro :

2a. (10 pts) La fonction matlab **[position value] = max(V)** retourne la valeur et la position du maximum de **V**. Dans notre cas **V** est uniquement un tableau 1D d'entiers `unsinged int`.
Ecrire en c la fonction **`int max_C(…)`** qui calcul la position et la valeur de **V** de manière équivalente de la fonction matlab ci-dessus. A vous de définir les paramètres manquants de la fonction **`int max_C(unsigned int V[] …)`**. **`max_C(…)`** retourne -1 si **V** est vide (i.e. sa taille <= 0) et 0 autrement. **V** est un tableau 1D valide, i.e. non NULL, pas besoin de le tester.

```
int max_C(unsigned int V[], int sz, int *pos, int *val ) {
  if (sz <=0) return -1;
    *val = V[0];
    *pos = 0;
    for (int i=0; i<sz; i++){
        if (V[i] > *val){
            *val = V[i];
            *pos= i;
        }
    }
    return 0;
}
```

3a. (4 pts) En Matlab, quelles sont les valeurs de A après l'exécution des lignes ci-dessous en utilisant les résultats intermédaires ?

```
A(1:5) = linspace(5,1,5)
```

```
A = [ 5   4   3   2   1 ]
```

```
A(8:10) = -2
```

```
A = [ 5   4   3   2   1  0  0  -2  -2  -2 ]
```

4a. (6 pts) En Matlab, avec la valeur initiale de `A = [2 7; 5 6]`, que valent A, B, C en utilisant les résultats intermédaires?

```
A(:,1)=[]
```

```
A = [ 7 ; 6 ]
```

```
B=[A(end-1) A(2)]
```

```
B = [ 7  6]
```

```
C=B*A(1:2,:)
```

```
C = 85
```

Nom/prénom : no camipro :

5a. (4 pts) En Matlab, avec n= 5, que valent D, E, F ?

```
D = (1:n).*(-1).^(n:-1:1)
```

```
D = [ -1    2   -3    4   -5 ]
```

```
E = find((2*n:-2:1)>2 )
```

```
E = [ 1    2    3    4  ]
```

6a. (2 pts) Ré-écrire en matlab, sans boucle, le code suivant, (i.e. vectorisez le code), avec **a** et **b** des vecteurs **lignes** de même tailles et **S** le résultat.

```
S = 0;
for n = 1:length(a)
    S = S + a(n)*b(n);
end
```

```
S = a * b'
```

7a. (6 pts) Qu'affiche **myPrint(3, 5.0)**?

```
int myPrint(int a, double b){

    double c= a/2;
    printf("1) %f\n",c);

    if ((a<10) && (a>100)){
        printf("2) %d\n",a);
    }
    double d= b/2;
    printf("3) %f\n",d);

    int e = b/2;
    printf("4) %d\n",e);

    unsigned char f = (a & 1) << 2;
    printf("5) %d\n", (int)f);

    return -1;
    printf("6) %d, %f\n", a,b);

}
```

```
1) 1.00000

- pas exécuté -

3) 2.50000

4) 2

5) 4

- pas exécuté -
```