



## Examen de rattrapage

Aucun document n'est autorisé  
Les solutions doivent être rédigées en C

### 1 Affichage

10 pts. ⌚35'

1. Qu'affichent les deux programmes suivants :

```
1 #include<stdio.h>
2 void Toto(int T[5][5],int A[],int n,int m){
3     int i, j, k=0 ;
4     for (i=0 ; i<n ; i++)
5         if (i%2)
6             for (j=0 ; j<m ; j++)
7                 A[k++] = T[i][j];
8         else
9             for (j=m-1 ; j>=0 ; j--)
10                A[k++] = T[i][j];
11 }
12 void main(){
13     int i;
14     int T[5][5]={{2,1},{3,4},
15                 {6,5},{7,8}};
16     int A[25];
17     Toto(T, A, 4,2);
18     for (i =0; i<8; i++)
19         printf("%d | " , A[i]);
20 }
```

```
1 #include<stdio.h>
2 #include<string.h>
3 void Loulou(char *A, char *B){
4     char *s = A, *r = B ;
5     int t = strlen (A)-1;
6     while (t >= 0){
7         *r = *(s+t);
8         t--;
9         r++;
10    }
11    *r = '\0';
12    printf ("r = %s\n", r-1);
13 }
14 int main (){
15     char S [10]= "abcd", T [10];
16     Loulou(S,T);
17     printf ("S = %s\n", S );
18     printf ("T = %s\n", T );
19     printf ("%s \n", strchr(S,'c'));
20 }
```

2. Que font les deux fonctions Toto et Loulou ? (deux lignes au maximum pour chacune d'entre elles).

### 2 Positifs et négatifs

5,5 pts. ⌚30'

On dispose d'un tableau T d'entiers à une seule dimension. La taille du tableau T est inférieure à 100.

Écrire une fonction `positifs_negatif` qui prend en entrée trois tableaux T, P et N ainsi que la taille réelle de T puis elle remplit les deux tableaux P avec les nombres positifs ou nuls et N avec les nombres strictement négatifs de T.

**Exemple :** soit le tableau T suivant :

T: 

-3	22	8	-12	41	-1	-17	-93	64	-23	13	-9	56	-61	-80
----	----	---	-----	----	----	-----	-----	----	-----	----	----	----	-----	-----

La fonction `positifs_negatifs` remplit les deux tableaux P et N comme suit :

P: 

22	8	41	64	13	56
----	---	----	----	----	----

N: 

-3	-12	-1	-17	-93	-23	-9	-61	-80
----	-----	----	-----	-----	-----	----	-----	-----

**Remarque.** On ne demande ni la saisie du tableau T ni l'affichage des tableaux P et N.

### 3 Palindrome

4,5 pts. ⌚25'

Un palindrome est une séquence de caractères symétrique. Par exemple il peut s'agir d'un mot qui peut-être lu indifféremment dans un sens ou dans l'autre. Par exemple : radar, elle, kayak, etc.

Écrire une fonction `est_Palindrome` qui permet de tester si une chaîne de caractères est un palindrome, elle renvoie 1 si c'est le cas et 0 sinon.

**Remarque.** écrire la fonction en utilisant des pointeurs (au lieu de la notation avec crochets []).

« Bon courage »



## Correction de l'examen final

Aucun document n'est autorisé  
Les solutions doivent être rédigées en C  
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

### 1 Affichage

10 pts. ⌚30'

1. Montrer l'historique d'exécution ainsi que l'affichage des deux programmes suivants : (4+4 pts)

```
1 void tabfct(int * P, int * Q, int t){
2     int * Pf = (P + (t - 1));
3     int * Qf = (Q + (t - 1));
4     int a=0;
5     while(P <= Pf && Q <= Qf){
6         a = *Q;
7         *Q = *P;
8         Q++;
9         *P = a;
10        P++;
11    }
12 }
13 void main()
14 {
15     int i;
16     int A[6]={2, 4, 6};
17     int T[6]={1, 3, 5};
18     tabfct(T, A, 2);
19     for (i =0; i<4; i++)
20         printf("%d | %d | " , A[i], T[i]);
21 }
```

```
1 int strfct(char *s1, char *s2){
2     while((*s1&&*s2)&&(*s1==*s2)){
3         s1++;
4         s2++;
5     }
6     return *s1 - *s2;
7 }
8 int main(){
9     char S[100] = "abcd";
10    char T[100] = "abcd";
11    char *s = S, *t = T, c;
12    if (S != T)
13        printf("S != T \n");
14    S[2] = 'd';
15    c = strfct(S,T);
16    if (c)
17        printf("%s, %d \n", S, c);
18    while(*(++s));
19    while(*(s++) = *(t++));
20    printf("S = %s \n", S );
21 }
```

2. Expliquer brièvement ce que font les deux boucles while (lignes 18 et 19) ainsi que le rôle des deux fonctions tabfct et strfct (2 pts).

### Solution

Affichage

```
1|2|3|4|6|5|0|0|
```

Affichage

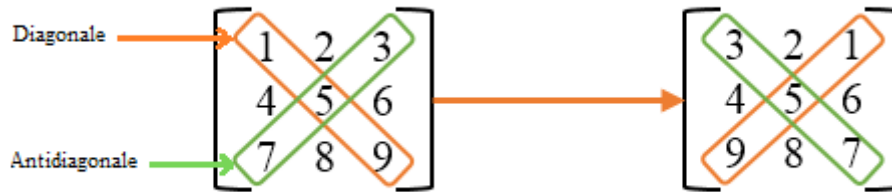
```
S!= T
abdd, 1
S = abddabcd
```

2. Les deux boucles while (lignes 18 et 19) permettent de concaténer les deux chaînes de caractères S et T. La fonction tabfct échange le contenu de deux tableaux et la fonction strfct compare deux chaînes de caractères.

### 2 Échange de diagonales

4 pts. ⌚25'

Écrire une fonction qui échange les valeurs de la diagonale d'un tableau à deux dimensions avec celles de l'antidiagonale comme indiqué dans l'exemple ci-dessous :



**Remarque.** Le tableau doit être passé comme paramètre à la fonction, on ne demande donc ni la saisie du tableau ni son affichage.

### Solution

```

1 #include <stdio.h>
2 void Echange_Diagonales(int A[][5], int N)
3 {
4     int i, temp;
5     for(i=0; i<N; i++)
6     {
7         temp = A[i][i];
8         A[i][i] = A[i][N-i-1];
9         A[i][N-i-1] = temp;
10    }
11
12 }
13 // La fonction main n'est pas demandée dans l'énoncé.
14 int main()
15 {
16     int A[5][5]={{1,2,3,8},
17                 {4,5,6,0},
18                 {7,8,9,0}
19                };
20     int i,j;
21
22     Echange_Diagonales(A,4);
23
24     for(i=0; i<4; i++)
25     {
26         for(j=0; j<4; j++)
27             printf("%d ", A[i][j]);
28
29         printf("\n");
30     }
31
32     return 0;
33 }

```

### 3 Le caractère le plus fréquent

6 pts. ⌚35'

- Écrire une fonction `plus_frequent` qui recherche dans une chaîne de caractères (passée en paramètre) la lettre minuscule la plus fréquente (qui présente le plus grand nombre d'occurrences), et retourne cette lettre. Si le plus grand nombre d'occurrences est commun à plusieurs lettres, c'est la première dans l'ordre alphabétique qui sera renvoyée par la fonction. Par ailleurs, si la chaîne reçue en paramètre est vide ou ne contient aucune lettre minuscule, la fonction doit renvoyer 0.

### Solution

```

1 char le_plus_frquent(char * str)
2 {
3     int freq[26] = {0};
4     char alphabet[26] = "abcdefghijklmnopqrstuvwxy";
5     int i = 0, max = 0;
6     if (!strcmp(str, ""))
7         return 0;
8     else{
9         while(*str != '\0'){
10            if (islower(*str))
11                freq[*str - 'a'] += 1;
12                str++;
13            }
14
15            for(i=0; i<26; i++)
16            {
17                if(freq[i] > freq[max])
18                    max = i;
19            }
20            return alphabet[max];
21        }

```

### Exemples :

- La lettre 'o' est la plus fréquente dans la chaîne "Toto&Loulou",
- La lettre 'e' est la plus fréquente dans la chaîne "abgcgdefef"
- La chaîne "AB5653@COM" ne contient aucune lettre minuscule.

2. Écrire une fonction main qui demande à l'utilisateur de saisir une chaîne de caractère puis appelle la fonction plus\_frquent et enfin affiche le résultat comme indiqués dans les exemples ci-dessus.

### Solution

```

1     }
2 }
3 int main()
4 {
5     char str[100];
6     char c ;
7
8     printf("Donnez une chaine ");
9     gets(str);
10
11    c = le_plus_frquent(str);
12    if (c)
13        printf("La lettre \'%c\' est la plus frequente dans la chaine \"%s
14            \".", c, str);
15    else
16        printf("La chaine \"%s\" ne contient aucune lettre minuscule.", str)
17        ;
18    return 0;

```