



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

6 pts. ⌚30'

1. Qu'affichent les deux programmes suivants :

```
1 void Toto(int T[10][10], int A[], int n, int m)
2 {
3     int i, j, k=0 ;
4     for (i=0 ; i<n ; i++)
5         if (i%2 == 0)
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[10][10]={{1, 2, 3},{6, 5, 4},
15                  {7, 8, 9},{12,11,12}};
16     int A[100];
17     Toto(T, A, 4,3);
18     for (i =0; i<12; i++)
19         printf("%d | " , A[i]);
20 }
```

```
1 void Loulou(char str[100])
2 {
3     int i = 0, m=0;
4     int c;
5     int T[255]={0};
6     while(str[i] != '\0')
7     {
8         T[str[i]]++;
9         i++;
10    }
11    for(i=0; i<255; i++)
12        if(T[i] > T[m])
13            m = i;
14    printf("m=%c ,T[m]=%d\n",m,T[m]);
15 }
16 int main(){
17     char str[100] = "Toto&Loulou";
18     Loulou(str);
19     return 0;
20 }
```

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

☞ **Remarque.** Les codes ascii des caractères 'L', 'l' et '&' sont 76,108 et 38 respectivement.

2 Gestion des salles de la faculté

14 pts. ⌚60'

On souhaite écrire un programme qui permet de gérer l'ensemble des salles de la faculté des sciences. La faculté des sciences comprend environ 100 salles réparties en trois types : Amphis, salles de TD et salles de TP.

Une salle est caractérisée par :

- son identificateur : un entier positif, par exemple : 001, 002, ...
- son nom : par exemple Labo1, Amphi3, N107, ...
- sa capacité : qui représente le nombre des étudiants maximal qu'elle peut accueillir.
- et son type : Amphi, TD, ou TP.

1. Créer la structure Salle.

1pt

2. Écrire les deux fonctions Saisir_Salle et Afficher_Salle.

2pt

3. Écrire une fonction qui permet de saisir toutes les salles de la faculté des sciences et une autre fonction qui permet de les afficher également. Le nombre de salles doit être passé comme paramètre aussi. **1pt**

4. Écrire la fonction Rechercher_Salle qui permet de rechercher une salle par son identificateur dans l'ensemble des salles de la faculté, le parcours doit s'arrêter une fois la salle est trouvée en retournant 1. La fonction retourne 0 si la salle n'existe pas. **2pt**

5. Écrire la fonction Supprimer_Salle permettant de supprimer une salle de l'ensemble des salles de la faculté. Si la salle à supprimer n'existe pas la fonction affiche un message d'erreur. **2pt**

6. Écrire une fonction Reserver_Salle qui prend en entrée un nombre d'étudiants et le type de la salle ainsi qu'un ensemble de salles et renvoie l'identificateur de la salle satisfaisant les critères souhaités. La fonction renvoie -1 sinon. **2pt**

7. Écrire une fonction Trier_Salles qui permet de trier les salles de la faculté par ordre lexicographique croissant de leurs noms. **3pt**

8. Écrire la fonction main qui permet de tester toutes les fonctions ci-dessus en utilisant le menu suivant jusqu'à ce que l'utilisateur décide de quitter le programme : **1pt**

```
----- Menu -----  
1: Saisir les salles.  
2: Afficher la liste des salles.  
3: Supprimer une salle.  
4: Reserver une salle.  
5: Trier les salles par nom.  
6: Quitter le programme.  
-----  
Donner votre choix :
```

« Bon courage »



Correction 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

6 pts. ☹️30'

1. Qu'affichent les deux programmes suivants :

```
1 void Toto(int T[10][10], int A[], int n, int m)
2 {
3     int i, j, k=0 ;
4     for (i=0 ; i<n ; i++)
5         if (i%2 == 0)
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[10][10]={{1, 2, 3},{6, 5, 4},
15                  {7, 8, 9},{12,11,12}};
16     int A[100];
17     Toto(T, A, 4,3);
18     for (i =0; i<12; i++)
19         printf("%d | " , A[i]);
20 }
```

```
1 void Loulou(char str[100])
2 {
3     int i = 0, m=0;
4     int c;
5     int T[255]={0};
6     while(str[i] != '\0')
7     {
8         T[str[i]]++;
9         i++;
10    }
11    for(i=0; i<255; i++)
12        if(T[i] > T[m])
13            m = i;
14    printf("m=%c ,T[m]=%d\n",m,T[m]);
15 }
16 int main(){
17     char str[100] = "Toto&Loulou";
18     Loulou(str);
19     return 0;
20 }
```

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

☞ **Remarque.** Les codes ascii des caractères 'L', 'l' est '&' sont 76,108 et 38 respectivement.

Solution

1.

Affichage

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 12 | 11 | 12 |

Affichage

m = o, T[m] = 4

2. La fonction Toto **transforme** le tableau T de **deux dimensions** en un tableau **linéaire** A en parcourant T **en allers-retours**.

La fonction Loulou **affiche le caractère le plus fréquent** dans la chaîne str ainsi que **sa fréquence**.

2 Gestion des salles de la faculté

14 pts. ☹️60'

On souhaite écrire un programme qui permet de gérer l'ensemble des salles de la faculté des sciences. La faculté des sciences comprend environ 100 salles réparties en trois types : Amphis, salles de TD et salles de TP.

Une salle est caractérisée par :

- son identificateur : un entier positif, par exemple : 001, 002, ...
- son nom : par exemple Labo1, Amphi3, N107, ...
- sa capacité : qui représente le nombre des étudiants maximal qu'elle peut accueillir.
- et son type : Amphi, TD, ou TP.

1. Créer la structure Salle.

1pt

Solution

```
1 typedef enum Type Type;
2 enum Type{Amphi, TD, TP};
3 typedef struct Salle Salle;
4 struct Salle {
5     int id;
6     char nom[20];
7     int capacite;
8     Type type;
9 } ;
```

2. Écrire les deux fonctions Saisir_Salle et Afficher_Salle.

2pt

Solution

```
1 void Saisir_Salle(Salle * s)
2 {
3     printf("Donner l'identificateur : ");
4     scanf("%d", &s->id);
5     printf("Donner le nom : ");
6     scanf("%s", s->nom);
7     printf("Donner la capacite : ");
8     scanf("%d", &s->capacite);
9     printf("Donner le type : 0 -> Amphi, 1 -> TD et 2 -> TP ");
10    scanf("%d", &s->type);
11 }
```

```
1 void Afficher_Salle(Salle s)
2 {
3     printf("[ Id : %d; ", s.id);
4     printf("nom : %s; ", s.nom);
5     printf("Capacite : %d; ", s.capacite);
6     switch (s.type)
7     {
8         case Amphi : printf(" Type : Amphi ] \n"); break ;
9         case TD : printf(" Type : TD ] \n"); break ;
10        case TP : printf(" Type : TP]\n"); break ;
11    }
12 }
```

3. Écrire une fonction qui permet de saisir toutes les salles de la faculté des sciences et une autre fonction qui permet de les afficher également. Le nombre de salles doit être passé comme paramètre aussi. 1pt

Solution

```

1 void Saisir_Salles_Faculte(Salle SF[], int n)
2 {
3     int i;
4     for(i=0; i<n; i++)
5         Saisir_Salle(&SF[i]);
6 }
7 void Afficher_Salles_Faculte(Salle SF[], int n )
8 {
9     int i;
10    for(i=0; i<n; i++)
11        Afficher_Salle(SF[i]);
12 }

```

4. Écrire la fonction Rechercher_Salle qui permet de rechercher une salle par son identificateur dans l'ensemble des salles de la faculté, le parcours doit s'arrêter une fois la salle est trouvée en retournant 1. La fonction retourne 0 si la salle n'existe pas. **2pt**

Solution

```

1 int Rechercher_Salle(Salle SF[], int n, int id)
2 {
3     int i=0;
4     while(SF[i].id != id && i<=n)
5         i++;
6     if (i<n)
7         return 1;
8     else
9         return 0;
10 }

```

5. Écrire la fonction Supprimer_Salle permettant de supprimer une salle de l'ensemble des salles de la faculté. Si la salle à supprimer n'existe pas la fonction affiche un message d'erreur. **2pt**

Solution

```

1 void Supprimer_Salle(Salle SF[], int *n, int id )
2 {
3     int i, j;
4     if(!Rechercher_Salle( SF, *n, id))
5         printf("Salle n' existe pas!\n");
6     else
7     {
8         //On suppose que l'id est unique
9         for(i=0; i<*n; i++)
10            {
11                if(SF[i].id == id)
12                {
13                    for(j=i; j<(*n)-1; j++)
14                        SF[j] = SF[j+1];
15                    (*n)--;
16                    //i--; si l'id n'est pas unique
17                }
18            }
19    }
20 }

```

6. Écrire une fonction Reserver_Salle qui prend en entrée un nombre d'étudiants et le type de la salle ainsi qu'un ensemble de salles et renvoie l'identificateur de la salle satisfaisant les critères souhaités. La fonction renvoie -1 sinon. **2pt**

Solution

```
1 int Reserver_Salle(Salle SF[], int n, int c, Type t)
2 {
3     int i=0;
4     while(i<n && (SF[i].capacite < c || SF[i].type!=t))
5         i++;
6     if (i<n)
7         return SF[i].id;
8     else
9         return -1;
10 }
```

7. Écrire une fonction Trier_Salles qui permet de trier les salles de la facultés par ordre lexicographique croissant de leurs noms. **3pt**

Solution

```
1 void Trier_Salles(Salle SF[], int n)
2 {
3     int i, j;
4     Salle s;
5     for(i=0; i<n; i++)
6     {
7         for(j=i+1; j<n; j++)
8         {
9             if(strcmp(SF[i].nom , SF[j].nom) > 0)
10                {
11                    s = SF[i];
12                    SF[i] = SF[j];
13                    SF[j] = s;
14                }
15        }
16    }
17 }
```

8. Écrire la fonction main qui permet de tester toutes les fonctions ci-dessus en affichant le menu suivant: **1pt**

```
----- Menu -----
1: Saisir les salles.
2: Afficher la liste des salles.
3: Supprimer une salle.
4: Reserver une salle.
5: Trier les salles par nom.
6: Quitter le programme.
-----
Donner votre choix :
```

Solution

```

1 int main(){
2 int n, choix, id, c;
3 Type t;
4 Salle SF[100], * s;
5 do{
6     puts("-----Menu-----");
7     puts("1: Lire les salles.");
8     puts("2: Afficher la liste des des salles.");
9     puts("3: Supprimer une salle.");
10    puts("4: Reserver une salle.");
11    puts("5: Trier les salles par nom.");
12    puts("6: Quitter le programme.");
13    puts("Donnez votre choix :");
14    scanf("%d", &choix);
15    switch(choix){
16        case 1:
17            printf("Donner le nombre de salles ");
18            scanf("%d", &n);
19            Saisir_Salles_Faculte(SF, n);
20            break;
21        case 2:
22            Afficher_Salles_Faculte(SF,n);
23            break;
24        case 3:
25            printf("Donner l'identificatuer de la salle a supprimer: ");
26            scanf("%d", &id);
27            Supprimer_Salle(SF, &n, id);
28            break;
29        case 4:
30            printf("Donner le nombre d'etudiant: ");
31            scanf("%d", &c);
32            printf("Donner le type : 0 -> Amphi, 1 -> TD et 2 -> TP ");
33            scanf("%d", &t);
34
35            printf("salle reserve %d\n",Reserver_Salle(SF, n, c, t));
36            break;
37        case 5:
38            Trier_Salles(SF,n);
39            break;
40        case 6:
41            printf("Fin du programme\n");
42            break;
43        default:
44            printf("Choix invalid!\n");
45
46    }
47 }while(choix != 6);
48 }

```

« Bon courage »