



Contrôle Continu

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

1 Affichage (10 pts)

Qu'affichent les deux programmes suivants :

```
void toto (int a, int b, int *c, int T[10]) {
    a++;
    b--;
    (*c)++;
    for (int i=9 ; i>0 ; i--)
        T[i]=T[i-1];
    T[0]=0;
}
void main () {
    int T[10]={1, 2, 3, 4, 5, 6, 7, 8, 9};
    int A=1, B=1, C=1;
    int *P1=&A ,*P2=&B, *P3=T;
    printf ("*P1=%d, *P2=%d, *P3=%d\n" ,*P1 ,*P2,*P3);
    printf ("T9=%d \n" , T[9]) ;
    toto(A,B,&C,T);
    printf ("A=%d, B=%d, C=%d\n",A,B,C);
    printf("T: ");
    for (int i=0 ; i<10 ; i++)    printf(" %d ",T[i]);
    printf ("\nT9=%d \n" , T[9]) ;
    printf("%d \n" , *P3+*P3++);
    printf("**P3=%d \n" , *P3);
}
```

```
#include <stdio.h>
#include <stdlib.h>

void main ()
{
    int A[3][4]={{1} ,{2 ,1} ,{3 ,2 ,1}};
    int B[4][3]={{1} ,{2} ,{3}};

    printf ("%d \t" , A[1][2]) ;
    printf ("%d \t" , **A);
    printf ("%d \t" , *(*A +1) +1) ;
    printf ("%d \n" , *(*A +1) +1) ;
    printf ("%d \t" , B[3][0]) ;
    for (int i=0 ; i<3 ; i++)
        for (int j=0 ; j<4 ; j++)
            B[j][i]= A[i][j];
    printf ("%d \t" , B[0][1]*B[1][2]) ;
    printf ("%d \t" , **(B+1));
    printf ("%d \n" , (*(B+2)+1)+1);
}
```

2 Matrice creuse (4 pts)

Une matrice est dite creuse si la plupart de ses éléments sont des zéros.

➤ **Exemple** : $M: \begin{pmatrix} 4 & 2 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$

Le nombre de zéros est: 5

Le nombre d'éléments différents de zéro est: 4

Donc cette matrice est creuse.

Écrire une fonction qui vérifie et affiche si une matrice donnée est creuse ou non. La fonction doit afficher le résultat comme indiqué dans l'exemple ci-dessus.

➤ **Remarque** : La matrice doit être passée comme paramètre à la fonction, on ne demande donc ni la saisie de la matrice ni son affichage.

3 Eclater un tableau (6 pts)

On dispose d'un tableau T d'entiers à une seule dimension.

1. Écrire une fonction ***pair_impair*** qui prend en entrée trois tableaux T, P et IMP puis elle remplit les deux tableaux P avec les nombres pairs et IMP avec les nombres impairs

Exemple : Soit le tableau T suivant :

T:

-1	9	5	2	0	-3	12	-6	7	1
----	---	---	---	---	----	----	----	---	---

La fonction ***pair_impair*** remplit les deux tableaux P et IMP comme suit :

P:

2	0	12	-6
---	---	----	----

IMP:

-1	9	5	-3	7	1
----	---	---	----	---	---

2. Écrire un programme principal qui saisie le tableau T et affiche les deux tableaux P et IMP.

« Bon courage »



Correction du Contrôle Continu

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

1 Affichage (10 pts)

Qu'affichent les deux programmes suivants :

```
void toto (int a, int b, int *c, int T[10]) {
    a++;
    b--;
    (*c)++;
    for (int i=9 ; i>0 ; i--)
        T[i]=T[i-1];
    T[0]=0;
}
void main () {
    int T[10]={1, 2, 3, 4, 5, 6, 7, 8, 9};
    int A=1, B=1, C=1;
    int *P1=&A , *P2=&B, *P3=T;
    printf ("*P1=%d, *P2=%d, *P3=%d\n" , *P1 , *P2, *P3);
    printf ("T9=%d \n" , T[9]);
    toto(A,B,&C,T);
    printf ("A=%d, B=%d, C=%d\n",A,B,C);
    printf("T: ");
    for (int i=0 ; i<10 ; i++) printf(" %d ",T[i]);
    printf ("\nT9=%d \n" , T[9]);
    printf("%d \n" , *P3+*P3++);
    printf("*P3=%d \n" , *P3);
}
```

```
#include <stdio.h>
#include <stdlib.h>

void main ()
{
    int A[3][4]={{1} , {2 ,1} , {3 ,2 ,1}};
    int B[4][3]={{1} , {2} , {3}};

    printf ("%d \t" , A[1][2]);
    printf ("%d \t" , **A);
    printf ("%d \t" , *(*A +1) +1);
    printf ("%d \n" , *(*A +1) +1);
    printf ("%d \t" , B[3][0]);
    for (int i=0 ; i<3 ; i++)
        for (int j=0 ; j<4 ; j++)
            B[j][i]= A[i][j];
    printf ("%d \t" , B[0][1]*B[1][2]);
    printf ("%d \t" , *(B+1));
    printf ("%d \n" , (*(B+2)+1)+1);
}
```

Solution

```
*P1 =1, *P2 =1, *P3=1
T9=0
A=1, B=1, C=2
T: 0 1 2 3 4 5 6 7 8 9
T9=9
0
*P3=1
```

```
0 1 1 1
0 4 0 1
```

2 Matrice creuse (4 pts)

Une matrice est dite creuse si la plupart de ses éléments sont des zéros.

➤ **Exemple** : $M: \begin{pmatrix} 4 & 2 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$

Le nombre de zéros est: 5

Le nombre d'éléments différents de zéro est: 4

Donc cette matrice est creuse.

Écrire une fonction qui vérifie et affiche si une matrice donnée est creuse ou non. La fonction doit afficher le résultat comme indiqué dans l'exemple ci-dessus.

➤ **Remarque** : La matrice doit être passée comme paramètre à la fonction, on ne demande donc ni la saisie de la matrice ni son affichage.

Solution

```
void matrice_creuse(int M[][10],int n, int m)
{ int nbre=0;
  for(int i=0;i<n;i++)
    for(int j=0;j<m;j++)
      if(M[i][j]==0) nbre++;
  printf("Le nombre de zeros est: %d\n",nbre);
  printf("Le nombre d'elements differents de zero est: %d\n",(n*m)-nbre);
  if(nbre>(n*m)/2) printf("Cette matrice est creuse.");
  else printf("Cette matrice n'est pas creuse.");
}
```

3 Eclater un tableau (6 pts)

On dispose d'un tableau T d'entiers à une seule dimension.

1. Écrire une fonction **pair_impair** qui prend en entrée trois tableaux T, P et IMP puis elle remplit les deux tableaux P avec les nombres pairs et IMP avec les nombres impairs

Exemple : Soit le tableau T suivant :

T :

-1	9	5	2	0	-3	12	-6	7	1
----	---	---	---	---	----	----	----	---	---

La fonction **pair_impair** remplit les deux tableaux P et IMP comme suit :

P :

2	0	12	-6
---	---	----	----

IMP :

-1	9	5	-3	7	1
----	---	---	----	---	---

2. Écrire un programme principal qui saisie le tableau T et affiche les deux tableaux P et IMP.

Solution

```
#include <stdio.h>
#include <stdlib.h>

void pair_impair(int T[],int t,int P[],int *p,int IMP[],int *imp)
{ int j=0,k=0;
  for(int i=0;i<t;i++)
    if(T[i]%2==0) P[j++]=T[i];
    else IMP[k++]=T[i];
  *p=j;
  *imp=k;
}

main(){
int n,m,l;
int T[100],P[100],IMP[100];
printf("donnez la taille de votre tableau:");
scanf("%d",&n);
printf("Le tableau T:\n");
for(int i=0 ; i<n ; i++)
  scanf("%d",&T[i]);

pair_impair(T,n,P,&m,IMP,&l);

printf("les nombres pairs sont:\n");
for(int i=0 ; i<m ; i++)
  printf("%d ",P[i]);

printf("\nles nombres impairs sont:\n");
for(int i=0 ; i<l ; i++)
  printf("%d ",IMP[i]);
}
```