



Examen final

Aucun document n'est autorisé
Les solutions doivent être rédigées en C
Les appareils portables doivent être éteints
La note de l'examen final sera également attribuée au CC et au TP

1 Affichage

12 pts. ⌚55'

Qu'affichent les deux programmes suivants (0.5 pt pour chaque valeur affichée) :

```
1 #include <stdio.h>
2 int toto(int *a, int b)
3 {
4     a = &b;
5     b=*a;
6     return *a*b;
7 }
8 int main()
9 {
10     int i=2, j, k=3, l, m=5, n=0;
11     int *p=&k, *q=&l, *r=&j;
12     j = i*2;
13     printf("i=%d, j=%d\n", i, j);
14     *q = *p;
15     printf("p=%d, q=%d\n", *p, *q);
16     p = q;
17     l = 4;
18     printf("p=%d, q=%d\n", *p, *q);
19     *p = (*p)+(*q);
20     printf("*k=%d, *l=%d\n", k, l);
21     toto(&m, n);
22     printf("*m=%d, *n=%d\n", m, n);
23     printf("r**r : %d\n", *r**r);
24     printf("r : %d\n", (*r+r)-r**r);
25 }
```

```
1 #include <stdio.h>
2 int main()
3 {
4     int i, j;
5     int A[4]={1}, B[8];
6     int *q = A;
7     int T[3][4]={{1, 2},{3}}, R[4][3];
8     char c, S1[6]="Toto", S2[8];
9     printf("A[1]=%d,*q=%d\n", A[1], *q);
10    for(i=0; i<4; i++)
11        B[i]=A[3-i];
12    printf("B0=%d,B3=%d\n", B[0], B[3]);
13    for (i=0; i<3; i++)
14        for (j=0; j<4; j++)
15            R[j][i]=T[i][j];
16    printf("R:%d %d\n", R[1][0], R[0][1]);
17    c='A'; //le code ascii du 'A' est 65
18    printf("%d, %c\n", c, c);
19    i=c+1;
20    printf("%c, %d\n", i, i);
21    strcpy(S2, S1);
22    strcat(S1, S2);
23    S2[2]='\0';
24    printf("S1=%s, S2=%s\n", S1, S2);
25 }
```

2 Séparer les nombres pairs et impairs

8 pts. ⌚35'

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

Écrire une fonction Pairs_Impairs qui prend en entrée trois tableaux T, P et IMP ainsi que la taille réelle de T puis elle remplit les deux tableaux P avec les nombres pairs et IMP avec les nombres impairs de T.

Exemple : soit le tableau T suivant :

T :

3	22	8	11	4	1	17	93	64	23	48	9	56	61	87
---	----	---	----	---	---	----	----	----	----	----	---	----	----	----

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

P :

22	8	4	64	48	56
----	---	---	----	----	----

IMP :

3	11	1	17	93	23	9	61	87
---	----	---	----	----	----	---	----	----

☞ **Remarque.** Pensez à stocker les tailles réelles des deux tableaux P et IMP.

« 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
La note de l'examen final sera également attribuée au CC et au TP

1 Affichage

12 pts. ⌚55'

Qu'affichent les deux programmes suivants (0.5 pt pour chaque valeur affichée) :

```
1 #include <stdio.h>
2 int toto(int *a, int b)
3 {
4     a = &b;
5     b=*a;
6     return *a*b;
7 }
8 int main()
9 {
10     int i=2, j, k=3, l, m=5, n=0;
11     int *p=&k, *q=&l, *r=&j;
12     j = i*2;
13     printf("i=%d, j=%d\n", i, j);
14     *q = *p;
15     printf("p=%d, q=%d\n", *p, *q);
16     p = q;
17     l = 4;
18     printf("p=%d, q=%d\n", *p, *q);
19     *p = (*p)+(*q);
20     printf("*k=%d, *l=%d\n", k, l);
21     toto(&m, n);
22     printf("*m=%d, *n=%d\n", m, n);
23     printf("*r**r : %d\n", *r**r);
24     printf("r : %d\n", (*r+r)-r**r);
25 }
```

```
1 #include <stdio.h>
2 int main()
3 {
4     int i, j;
5     int A[4]={1}, B[8];
6     int *q = A;
7     int T[3][4]={{1, 2},{3}}, R[4][3];
8     char c, S1[6]="Toto", S2[8];
9     printf("A[1]=%d,*q=%d\n", A[1], *q);
10    for(i=0; i<4; i++)
11        B[i]=A[3-i];
12    printf("B0=%d,B3=%d\n",B[0],B[3]);
13    for (i=0; i<3; i++)
14        for (j=0; j<4; j++)
15            R[j][i]=T[i][j];
16    printf("R:%d %d\n",R[1][0],R[0][1]);
17    c='A'; //le code ascii du 'A' est 65
18    printf("%d, %c\n", c, c);
19    i=c+1;
20    printf("%c, %d\n", i, i);
21    strcpy(S2,S1);
22    strcat(S1, S2);
23    S2[2]='\0';
24    printf("S1=%s, S2=%s\n", S1, S2);
25 }
```

Solution

Affichage

```
i=2, j=4
p=3, q=3
p=4, q=4
*k=3, *l=8
*m=5, *n=0
*r**r: 16
r: 8
```

Affichage

```
A[1]=0,*q=1
B0=0,B3=1
R :2 3
65, A
B, 66
S1=TotoToto, S2=To
```

2 Séparer les nombres pairs et impairs

8 pts. ⌚35'

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

Écrire une fonction `Pairs_Impairs` qui prend en entrée trois tableaux `T`, `P` et `IMP` ainsi que la taille réelle de `T` puis elle remplit les deux tableaux `P` avec les nombres pairs et `IMP` avec les nombres impairs de `T`.

Exemple : soit le tableau `T` suivant :

`T` :

3	22	8	11	4	1	17	93	64	23	48	9	56	61	87
---	----	---	----	---	---	----	----	----	----	----	---	----	----	----

La fonction `Pairs_Impairs` remplit les deux tableaux `P` et `IMP` comme suit :

`P` :

22	8	4	64	48	56
----	---	---	----	----	----

`IMP` :

3	11	1	17	93	23	9	61	87
---	----	---	----	----	----	---	----	----

Remarque. Pensez à stocker les tailles réelles des deux tableaux `P` et `IMP`.

Solution

```
1 #include<stdio.h>
2 void Tab_Pairs_Impairs(int T[100], int lt ,int P[100], int *lp, int IMP
  [100], int *limp) //1pt les deux pointeurs *lp et *limp ne
  sont pas comptabilisés ici
3 {
4     int i,p, imp;
5     i=0; p=0; imp=0;
6     while(i<lt ) //1pt
7         if (T[i]%2) //1pt
8             IMP[imp++]=T[i++]; //1,5pt
9         else //1pt
10            P[p++]=T[i++]; //1,5pt
11     *lp = p; //0,5pt
12     *limp = imp; //0,5pt
13 }
14
15 void main()
16 {
17     int i;
18     int lp, limp;
19     int T[100]={3 , 8 , 11 , 17 , 23 , 48 , 56 , 61 ,87, 93,98};
20     int P[100], IMP[100];
21     printf("\nT : ");
22     for(i=0; i<11; i++) printf("%d ", T[i]);
23     Tab_Pairs_Impairs(T, 11, P, &lp, IMP, &limp);
24     printf("\nP : ");
25     for(i=0; i<lp; i++) printf("%d ", P[i]);
26     printf("\nIMP : ");
27     for(i=0; i<limp; i++) printf("%d ", IMP[i]);
28 }
```