

Cet examen contient 4 pages.

Les Documents, calculatrices, téléphone portable sont interdits. Veuillez rendre une copie propre et claire. La qualité de l'écriture et de la présentation sera prise en compte dans la note finale.

► Exercice 1. Questions de réflexion (3 Points)

1. P pointe sur une cellule d'une liste linéaire, on souhaite libérer la mémoire qu'il occupe. Comment faire ?

- (a) `P = NULL;` (c) `free(P);`
(b) `*P = 0;` (d) `free (*P);`



.....
`free(P); // (1pt)`
.....



2. On considère les déclarations suivantes : `int *p, i;`

- (a) Écrire l'instruction qui permet à p de pointer la variable entière i statique
(b) Écrire l'instruction qui permet à p de pointer une variable entière dynamique.



.....
(a) `p=&i;`
(b) `p=(int *) malloc(sizeof(int));`
(1pt)
.....



3. En fonction des déclarations suivantes, que donne l'affichage de "p2" ?

```
int A = 5;  
int *p1 = &A;  
int **p2 = &p1;
```

- (a) La valeur de A (c) L'adresse de A
(b) L'adresse de p1 (1pt)

► Exercice 2. Les chaînes de caractères et pointeurs (4 Points)

1. Écrire en langage C une procédure **récursive** void ftnirp (char *chaine) qui affiche une chaîne de caractères à l'envers, sans la modifier. Exemple : ftnirp("abcdef") doit afficher fedcba



.....
`void ftnirp (char *chaine)`
`{`
 `if(*chaine){`
 `ftnirp(chaine+1);`
 `printf("%c",chaine[0]);`
 `}`
`}`
.....



2. Dans le code suivant, que vaut pc après l'exécution de l'instruction while ?

```
char ch[] = "Bonjour", *pc = ch;
while (*pc) pc++;
```

✂
Le caractere de fin de chaine (1pt)
..... ✂

3. Donnez l'état des deux tableaux T et POS, ainsi que les valeurs des deux indice I et J, après l'exécution du code suivant :

```
void main( )
{int T[10] = {-3, 4, 0, -7, 3, 8, 0, -1, 4, -9}, POS[10];
int I,J;
for (J=0,I=0 ; I<10 ; I++)
    if (*(T+I)>0)
        { *(POS+J) = *(T+I);
          J++;
        }
}
```

✂
..... ✂

► Exercice 3. Les listes chaînées (7 Points)

Soit les types de données suivants :

<pre>typedef struct{ int Numero; char Type[30]; char Proprio[30]; float Solde; }Compte ;</pre>	<pre>typedef struct cel{ Compte info; struct cel* suiv; } Cellule;</pre>	<pre>typedef struct list{ Cellule *premier; int nb_compte; } List_compte;</pre>
--	--	---

Toutes les fonctions et procédures doivent être écrites en langage C.

1. Écrire une fonction **récursive Nb_compte_rouge()** qui prend en paramètre une liste des comptes et retourne le nombre des comptes dont le soldes ≤ 0 (4pt).

```
int Nb_compte_rouge(List_compte lc)
```

✂


```
int Nb_compte_rouge(List_compte lc){
Cellule *p=lc.premier; //0,5pt
if(lc.premier==NULL) return 0; //1pt
lc.premier=lc.premier->suiv;
lc.nb_compte--; //0,5pt
if(p->info.Solde<=0)
```

```
return(1+Nb_compte_rouge(lc)); //1pt
else
return(0+Nb_compte_rouge(lc)); //1pt
}
```

..... ✂

2. Écrire une fonction **réursive** **Total_soldes()** qui prend en paramètre une liste des comptes et retourne le total des soldes de la liste. (3pt).

✂

```
float Total_solde(List_compte lc){
Cellule *p=lc.premier; //0,5pt
if(lc.premier==NULL) return 0; //1pt
lc.premier=lc.premier->suiv;
lc.nb_compte--; //0,5pt
return(p->info.Solde+Total_solde(lc)); //1pt
}
```

..... ✂

► Exercice 4. Les files/piles (4 Points)

En vous basant sur les primitives vues en cours, dessiner l'état des deux piles P et pp, déclarées au niveau de la main(), après la suite des instructions suivantes :

```
Pile* devine(Pile **P){
int x;
Pile *p1, *p2, *p3;
CreerPile(&p1);
CreerPile(&p2);
CreerPile(&p3);
while(!PileVide(*P)){
x=(*P)->donnee;
if ((x%2)==0)
empiler(&p1,depiler(P));
else empiler(&p2,depiler(P));
}
while (!PileVide(p2))
empiler(P,depiler(&p2));
```

```
while (!PileVide(p1))
empiler(&p3,depiler(&p1));
return p3;
}
void main()
{ Pile *P, *pp;
CreerPile(&P);
empiler(&P,10);
empiler(&P,11);
empiler(&P,12);
empiler(&P,13);empiler(&P,14);
empiler(&P,15);empiler(&P,16);
pp=devine(&P);
}
```

[Avant]

16
15
14
13
12
11
10

P

→

[Après]

15
13
11

P

16
14
12
10

pp

► Exercice 5. Les fichiers (2 Points)

Ayant un fichier `essai.txt` qui contient deux lignes :

3 6 18

7 20 3

Donnez le contenu du fichier après l'exécution du code suivant :


```
FILE *f=fopen("essai.txt","r+");  
int x=6;  
fprintf(f,"%d %d %d\n",x,x+1,x+5);  
fclose(f);
```



.....
Contenu du fichier : (2pt)

6 7 11

7 20 3

..... 

Bon travail