
Cet examen contient 5 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. Ayant un pointeur P sur entier, on souhaite créer une variable dynamique entière de valeur 5. Comment faire ?

(a) `P = NULL ; P=5 ;`

(b) `*P = 5 ;`

(c) `int a ; P=&a ; *P=5 ;`

(d) `P= malloc (sizeof(int)) ; P=5 ;`

(e) `P= malloc (sizeof(int)) ; *P=5 ;`

2. On considère l'entête (ou prototype) de fonction suivante :

```
void essai(float *x, float *y, int i, char z, char *c)
```

On considère les déclarations suivantes :

```
float a, c ;
```

```
int j ;
```

```
char b, h ;
```

Quels sont les appels de fonction corrects ?

(a) `essai (a, c ; j ;b, h) ;`

(b) `essai (&a, &b, c, j, h) ;`

(c) `essai (&a, &c, 3, 'b', &b) ;`

(d) `essai (&a, &c, j, b, &h) ;`

(e) `essai (a ; j ; b ; &h) ;`

3. On considère la déclaration suivante : `int *ptint ;` ptint peut contenir des valeurs qui sont :

(a) Des valeurs de variables de type entière (int)

(b) Des adresses de variables de type entière (int)

4. Quelle est la différence entre une pile et une file ?

► Exercice 2. Récursivité (4 Points)

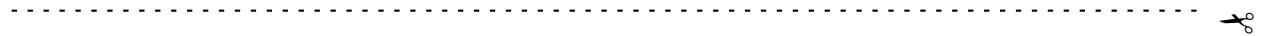
```
int Devine (int n)
{
    if (n/10==0) return (n) ;
    else
    return n%10+ Devine(n/10) ;
}
```

1. Quelle est la valeur retournée par l'appel de Devine(542) ?
2. Que permet de faire la fonction Devine ?
3. Écrire en C la version **itérative** de la fonction Devine().



1. 11 (1pt)
2. elle permet calculer la somme des chiffres d'un nombre donné. (1pt)

```
3. int Devine(int n) {
    int s ;
    s = 0 ;
    while ( n != 0){
        s = s +( n % 10) ;
        n = n/10 ;
    }
    return (s) ;
}
```



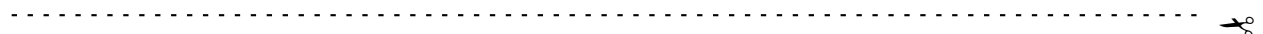
► Exercice 3. Les chaînes de caractères et pointeurs (3 Points)

Quel affichage est produit par le code suivant :

```
void app(char *c,int **p,int *l){
    **p +=++(*l);
    *l+=2;
    *p=1;
    *c='N';}
void main(){
    char ch[] = "ENSAI", *ct;
    ct=ch+1;
    int *ptr, x=4, y=7;
    ptr=&y;
    app(ct,&ptr,&x);
    printf("%d %d %d %s\n", *ptr,x,y, ch);
    ct++;
    printf("%c", *ct);}
```



7 7 12 ENSAI (2pt)
S(1pt)



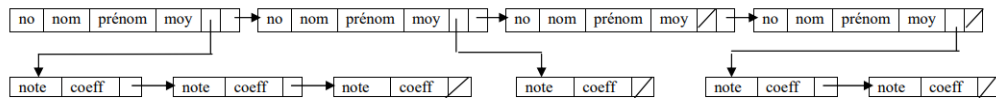
► Exercice 4. Les listes et fichiers : calcul de moyenne des étudiants (10 Points)

Le département dans lequel vous êtes inscrit souhaite gérer les notes de ses étudiants. Les étudiants ont pour identifiant leur numéro d'étudiant. Ils ont un nom et un prénom. Ces informations sont stockées dans une liste chaînée dont chaque élément comporte aussi un champ moy pour la moyenne de l'étudiant et un champ eval qui est un pointeur sur sa liste de notes. La liste de notes de chaque étudiant est aussi une liste chaînée dont la tête est le champ eval de la cellule de l'étudiant. On dispose des déclarations suivantes :


```
typedef struct not{
float note;
float coeff;
struct not* suivant;
}notes ;

typedef struct etud{
char no[10];
char nom [30];
char prenom [30];
float moy;
notes * eval;
struct etud* suivant;
} etudiant ;
```

On peut représenter cette structure par la figure ci-dessous :



Remarques :

 Cette notation équivaut à Nil

Il se peut qu'un étudiant n'ait pas encore de note. C'est le cas du 3^{ème} étudiant de la liste de l'exemple ci-dessus. Le pointeur eval est égal à Nil.

On suppose que tous les champs de la liste des étudiants sont remplis sauf le champ moy. On suppose que toutes les notes des étudiants et tous les coefficients sont remplis

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

1. Écrire une procédure **moyennesEtudiants()** qui prend en paramètre une liste des étudiants et la parcourt et qui calcule et met à jour le champ moy de chaque étudiant à l'aide de la liste des notes sur laquelle pointe le champ eval. La procédure moyennesEtudiants prend en paramètre la tête de la liste des étudiants. (4pt).

```
void moyennesEtudiants(etudiant* e){
/* Procédure qui calcule la moyenne de chaque etudiant
et met a jour le champ moy de chaque etudiant de la liste
passee en parametre */
...
}
```

```

void moyennesEtudiants(etudiant* etu){
float totCoeff,totNotes;
etudiant *petu ; /* pointeur de parcours de la liste des etudiants
*/
notes *pmat ; /* pointeur de parcours de la liste des notes de
chaque et.*/
petu =etu;
while petu != NULL{ /* parcours de la liste des etudiants */
totCoeff = 0; /* au debut il n'y a ni note ni coefficient */
totNote = 0; /* pour l'etudiant */
pmat = petu->eval; /* eval est la tete de liste des notes de l'et.
*/
while pmat !=NULL {/* parcours de la liste des notes de l'etudiant
*/
totCoeff = totCoeff + pmat->coeff; /* somme des coefficients */
totNote = totNote + pmat->note * pmat->coeff; /* somme ponderee */
pmat = pmat->suivant; /* on passe a la note suivante */
}
if petu->eval !=NULL
/* calcul et memorisation dans la cellule de l'etudiant de la
moyenne de ses notes, s'il en a */
petu->moy = totNote / totCoeff;
else
petu->moy=0;
}}

void moyennesEtudiants(etudiant* e){
/* Procedure qui calcule la moyenne de chaque etudiant
et met a jour le champ moy de chaque etudiant de la liste passee en
parametre */
float s,c;
notes *n;
for(;e!=NULL; e=e->suivant){
s=0; c=0;
n=e->eval;
for(;n!=NULL; n=n->suivant){
s+=n->note*n->coeff;
c+=n->coeff;
}
if(s==0)e->moy=0;
else e->moy=s/c;
}}

```

2. Écrire une procédure **affReussite()** récursive qui prend en entrée une liste d'étudiants et affiche les noms de tous les étudiants ayant une moyenne supérieure ou égale à 10 sur 20.(3pt).

```

void AffReussite ( etudiant* l ){ // 0.5 pt

```

```

if ( l !=0) // 0.5 pt
{ if ( l -> moy >=10) // 0.5 pt
printf ( " % s \ n" , l -> nom ); // 0.5 pt
l=l->suivant; // 0.5 pt
AffReussite ( l ); // 0.5 pt
}
}

```

3. Ecrire une procédure en C **Imprimer()** qui prend une liste d'étudiants et créer un fichier "etudiants.txt" (écriture seule et destruction de l'ancienne version si elle existe et création si le fichier n'existe pas) avec tous les éléments de la liste des étudiants (les informations d'un seul étudiant (à savoir son no, nom, prenom et moy) par ligne suivi de ses notes (note et coeff) sur une autre ligne. *(3pt)*

453244 Ben Ali Mohamed 13,875
13 2 14 1 15,5 1

```

void Imprimer(etudiant* l){
FILE *f=fopen("etudiants.txt","w");
notes* p;
if(f){
while (l!=NULL){
fprintf(f, "%s %s %s %f\n", l->no, l->nom, l->prenom, l->moy);
printf( "%s %s %f\n", l->nom, l->prenom, l->moy);
p=l->eval;
for(;p!=NULL; p=p->suivant){
fprintf(f, "%f %f ", p->note, p->coeff);
}
if(l->eval!=NULL){fprintf(f,"\n");}
l=l->suivant;
}}
else
printf("impossible d'ouvrir");
fclose(f);
}

```

Bon travail