Université de Carthage
École Supérieure de la Statistique et de l'Analyse de l'Information

Mme Aïcha El Golli Jabbes/ Mme Sihem Mansour

Examen Session Contrôle

Année Universitaire **2021-2022**

Durée : 1h30 **Algorithmique et programmation C 2** $1^{ère}$ année

---

Cet examen contient 2 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 (4 Points)

1. 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, *k ;
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, *k, 'b', &h) ;`

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

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

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

2. On considère la déclaration suivante :int *ptint ; Ecrire l'instruction qui permet à ptint de pointer une variable entière dynamique.
3. Soit le pointeur de fichier défini par

```
FILE *fp ;
```

on veut ouvrir le fichier «essai.txt» en mode lecture.
Quelles sont les propositions correctes.

(a) `"essai.txt"=fopen(fp, "r") ;`

(b) `fp=fopen("essai.txt", "r") ;`

(c) `fp=fopen(essai.txt, "r") ;`

(d) `fopen(fp, "essai.txt", "r") ;`

(e) le fichier «essai.txt» doit exister

(f) Si le fichier «essai.txt» n'existe pas, il est créé.

4. Comment lire un réel et un entier et séparés par un espace (déclarés par : double d ; int x ;) dans un fichier pointé par f de type FILE ?

(a) `fscanf ("%lf %d" , &d, &x);`

(b) `fscanf (f,"%f %i" ,&d, x);`

(c) `fscanf (f,"%lf %d" , &d, &x);`

(d) `fscanf (f,"%d %i" , &d, &x);`

▸ Exercice 2. Récursivité (4 Points)

```
void devine(int x){
if(x!=0)
{     devine(x/8);
      printf("%d",x%8);
}}
```

1. Quelle est la valeur retournée par l'appel de devine(10) ?
2. Que permet de faire la fonction devine ?

---

1. 12 (2pt)
2. elle permet de représenter un entier en nombre octale (base 8). (2pt)

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

Quel affichage est produit par le code suivant :

```
void change(char *c,int *p){
(*p )++;
*c='U';}
```

```
void main(){
char ch[] = "ESSAI", *ct;
ct=ch+3;
int *ptr, x=5;
ptr=&x;
change(ch,ptr);
printf("%d %d %s\n", x,*ptr, ch);
ct--;
printf("%c", *ct);}
```

---

6 6 USSAI (2pt)
S(1pt)

▸ Exercice 4. Les listes et fichiers (9 Points)

Ayant une liste entière positive simplement chaînée définit comme suit :

```
typedef struct cel{
int info;
struct cel *suiv;
}cellule;
```

```
typedef struct l{
cellule *tete;
int taille;
}list;
```

et ayant une fonction principale définit comme suit :

```
int main()
{
    list l;
    FILE *pf=fopen("essai.txt", "w");
    l=creer();
    remplir(&l,4);
    printf("min des elements: %d\n",rechercheMin(l));
  sauvegarde(l, pf);
    return 0;
}
```

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

1. Écrire la fonction **list creer()** qui permet de créer une liste vide. ***(2pt)***
2. Écrire la procédure **void remplir(list *pl, int nb)** qui permet de remplir une liste passée en paramètre avec nb entiers **POSITIFS** demandés à l'user. L'ajout se fait en tête de liste. ***(3pt)***
3. Écrire une fonction **int rechercheMin(list l)** qui recherche la valeur minimale dans une liste l (qui n'est pas ordonnée). La fonction doit renvoyer la valeur minimale de la liste, ou -1 si la liste est vide. ***(2pt)***.

```
list creer(){
list ll;
ll.taille=0;
ll.tete=NULL;
return ll;
}
void remplir(list *ll, int nb){
  int i, val;
  cellule *p;
  for(i=1;i<=nb;i++){
    p= (cellule*)malloc( sizeof(cellule));
    do{
        printf("entrez la %d valeur",i);
        scanf("%d",&p->info);
    }while(p->info<0);
    p->suiv=ll->tete;
    ll->tete=p;
    ll->taille++;
  }
}
int rechercheMin(list l){
    int min=-1, i;
    for(i=1;i<=l.taille;i++){
        if(i==1) min=l.tete->info;
        else{
        if(min>l.tete->info)
            min=l.tete->info;
                }
    l.tete=l.tete->suiv;
    }
    return min;
}
```

4. Écrire la procédure **sauvegarde(list l, FILE *f)** qui permet de sauvegarder une liste donnée dans un fichier (éléments séparés par un espace). ***(2pt)***

```
void sauvegarde(list l, FILE *f){

    for(;l.tete!=NULL;l.tete=l.tete->suiv)
 fprintf(f, "%d ", l.tete->info);
}
```

Bon travail