

Université Ibn Tofaïl Faculté des sciences Département d'Informatique	Examen : Programmation II SMI/S4 2019/2020
Session : Printemps (Normale)	Le 09 Septembre 2020 Durée : 1h

Q1 : une «Structure » est un agrégat de plusieurs objets de types différents regroupés dans une même variable.

A. Vrai

B. Faux

Q2 : Tout objet de type struct Registre est codé sur 8 octets :

```
struct Registre{
    unsigned int actif : 1;
    unsigned int valeur : 31;
};
```

A. Vrai

B. Faux

Q3 : Si **p** est un pointeur sur une structure, les deux instructions **p->membre** et **(*p).membre** sont équivalentes pour accéder aux membres de p.

A. Vrai

B. Faux

Q4 : Une fonction contient une récursivité mutuelle s'il contient comme paramètre un appel à elle-même.

A. Vrai

B. Faux

Q5 : Les éléments d'une liste chaînée sont placés de façon adjacente dans la mémoire.

A. Vrai

B. Faux

Q6 : (une seule réponse) Dans le traitement des fichiers en C, le mode d'ouverture "**w+**" permet une:

A. **lecture** et **écriture** à partir du début du fichier. Le fichier doit obligatoirement exister, sinon la fonction échoue.

B. **lecture** et **écriture** à partir du début du fichier. Si le fichier n'existe pas, il sera créé. S'il existe, son contenu est effacé.

C. **écriture** à partir de la fin du fichier. Si le fichier n'existe pas, il sera créé.

Q7 : (une seule réponse) En fonction des déclarations suivantes, que donne l'affichage de « p2 » ?

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

A. la valeur de p1

B. l'adresse de A

C. l'adresse de p1

Q8 : (plusieurs réponses possibles) Soient les lignes d'instruction suivantes, Comment accéder à l'année du 3^{ème} timbre de la collection?

```
struct timbre {
    int prix ;
    int annee ;
    char origine[20] ;
    char image[20] ;
};
```

```
struct timbre COLLECTION[10] ;
```

- A. COLLECTION[2,2]
- B. COLLECTION[2].annee
- C. COLLECTION.annee[2]
- D. (COLLECTION+2) ->annee

Q9 : (plusieurs réponses possibles) On considère l'entête (ou prototype) de fonction suivante:

```
void maFonction(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?

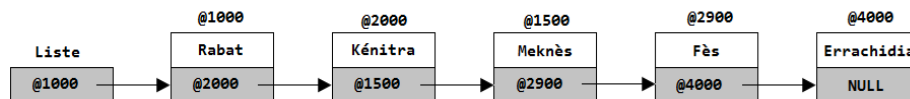
- A. maFonction(a,c;j,b,h);
- B. maFonction(&a,&b,c,j,h);
- C. maFonction(&a,&c,5,'j',b);
- D. maFonction(&a,&c,j,b,h);

Q10 : (une seule réponse) soit un fichier binaire qui contient des entiers de 0 à 40. Après l'ouverture du fichier en mode lecture avec le flot **f_in**, quelle est la sortie des instructions suivantes :

```
fseek(f_in, -8 * sizeof(int), SEEK_END);
printf("\n Position %ld", ftell(f_in));
fread(&i, sizeof(i), 1, f_in);
printf("\t i = %d", i);
```

- A. Position 160 i = 40
- B. Position 133 i = 32
- C. Position 132 i = 33

Q11 : (une seule réponse) soit la liste chaînée suivante :



Sachant que l'insertion des nœuds se fait en tête, quel est le premier élément inséré dans cette liste ?

- A. Errachidia
- B. Rabat
- C. Fès

Q12 : (une seule réponse) sachant que « Liste » est un type qui définit une liste chaînée, que fait la fonction suivante ?

```
void fonction(Liste * adrListe){
    if((*adrListe) != NULL){
        Liste queue = (*adrListe)->suivant;
        free(*adrListe);
        (*adrListe) = queue;
    }
}
```

- A. libérer l'espace mémoire occupé par la liste chaînée adrListe
- B. supprimer un élément de la queue de la liste adrListe
- C. supprimer un élément de la tête de la liste adrListe

Q13 : (une seule réponse) Dans le traitement des fichiers en C, le rôle de la fonction « rewind » est:

- A. renvoyer la taille d'un fichier binaire
- B. se positionner au début du fichier
- C. équivalent à fseek(flott, 0, SEEK_END)

Q14 : (plusieurs réponses possibles) pour déclarer une liste simplement chaînée d'entiers, laquelle des déclarations suivantes est correcte :

- A.

```
typedef int Type;
typedef struct Noeud * Liste;
typedef struct Noeud{
    Type info;
    Liste suivant;
} Noeud;
```
- B.

```
struct Noeud{
    int info;
    Noeud suivant;
};
```
- C.

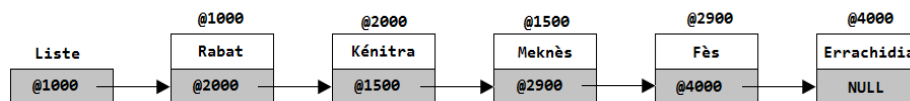
```
struct Noeud{
    int info;
    struct Noeud *suivant;
};
```
- D.

```
typedef struct Noeud{
    char info;
    Liste suivant;
} *Liste;
```

Q15 : (plusieurs réponses possibles) dans le traitement des fichiers en C, Les fonctions d'entrées-sorties binaires **fread** et **fwrite** permettent :

- A. de transférer des données dans un fichier après transcodage.
- B. de transférer des données dans un fichier sans transcodage.
- C. de manipuler des données de grande taille ou ayant un type composé.
- D. de retourner le nombre de données transférées.

Q16 : (une seule réponse) soit la liste chaînée suivante :



Que devient la liste après exécution d'un appel à la fonction suivante :

```
void fonction(Liste * adrListe){
    if((*adrListe) != NULL){
        if((*adrListe)->suivant == NULL){
            free(*adrListe);
            (*adrListe) = NULL;
        }
        Liste tmp = (*adrListe);
        Liste ptmp = (*adrListe);
        while(tmp->suivant != NULL){
            ptmp = tmp;
            tmp = tmp->suivant;
        }
        ptmp->suivant = NULL;
        free(tmp);
    }
}
```

- A. Kénitra ->Meknès->Fès->Errachidia
- B. Rabat->Kénitra ->Meknès->Fès
- C. Meknès->Fès->Errachidia->Rabat
- D. Rabat->Kénitra ->Meknès

Q17 : (une seule réponse) Quel est le résultat d'exécution d'un appel à la fonction suivante sur la même liste chaînée de la question **Q16** ?

```
int fonction(Liste liste){  
    if(liste == NULL)  
        return 0;  
    return fonction(liste->suivant)+1;  
}
```

- A. 6
- B. NULL
- C. 5
- D. erreur d'exécution

Q18 : (plusieurs réponses possibles) Soit le pointeur de fichier défini par FILE *fp ; on veut ouvrir le fichier «C:/fich.txt» en mode lecture. Quelles sont les propositions correctes ?

- A. "C:/fich.txt"=fopen(fp,"r")
- B. fp=fopen("C:/fich.txt","r");
- C. Si le fichier «C:/fich.txt» n'existe pas, il est créé.
- D. le fichier «C:/fich.txt» doit exister.

Q19 : (plusieurs réponses possibles) Quel est le contenu du fichier « F_SORTIE.dat » après l'exécution de ce code :

```
main(){  
    FILE *f_out;  
    int *tab1;  
    int i;  
    tab1 = (int*)malloc(30 * sizeof(int));  
    for (i = 0 ; i < 30; i++)  
        tab1[i] = i;  
    if ((f_out = fopen("F_SORTIE.dat", "rb+")) == NULL){  
        printf("\nImpossible d'écriture dans le fichier\n");  
        return(1);  
    }  
    fwrite(tab1, 10 * sizeof(int), 2, f_out);  
    fclose(f_out);  
}
```

- A. Des entiers de 0 à 30
- B. Le programme affiche le message
« Impossible d'écriture dans le fichier » si le
fichier n'existe pas
- C. Des entiers de 0 à 19
- D. Des entiers de 0 à 10 deux fois

Q20 : (plusieurs réponses possibles) soient les déclarations des variables suivantes:

```
int i ;  
int tab[10] ;  
char c ;  
int *pti ;  
char *ptc ;
```

Cocher ce qui est juste (pour les lignes de code, ce qui compile sans warning) :

- A. pti=&tab; *pti=4;
- B. ptc=&c; *ptc='a';
- C. tab[i] est equivalent à *(tab+i);
- D. tab[i] est equivalent à &(tab+i);

Bonne chance