Université Internationale de Casablanca LAUREATE INTERNATIONAL UNIVERSITIES Nous innovons pour votre réussite !	**Ecole d'Ingénierie** **Filières :** CPI & MIAGE **Classe :** 2ème année	**Cours : Programmation Structurée 2** **Professeur** : MOUJAHID Abdallah **Date** : 16/01/2017

Partie I : QCM *(6 points)*

Répondez en entourant la/les lettre(s) correspondant(s) à la/les bonne(s) réponse(s).
+0,75 pour une bonne réponse, 0 pour absence de réponse, -0,25 pour une mauvaise réponse.

1. Donner le résultat de l'exécution du code suivant :

```
#include <stdio.h>
int main() {
   int a, b;
   int *ptr1, *ptr2;
   a = 10;
   b = a+1;
   ptr1 = &a;
   ptr2 = ptr1;
   b = (*ptr2)--;
   printf("a = %d, b = %d,*ptr1 = %d,*ptr2 = %d\n", a, b,*ptr1,*ptr2);
   return 0;
}
```

   a. ne compile pas
   b. provoque une erreur à l'exécution
   c. affiche a = 11, b = 10, *ptr1 = 11, *ptr2 = 11
   d. affiche a = 10, b = 11, *ptr1 = 11, *ptr2 = 10
   e. affiche a = 21, b = 10, *ptr1 = 10, *ptr2 = 21
   f. Aucune des propositions ci-dessus.

2. Donner le résultat de l'exécution du code suivant :

```
#include <stdio.h>
int main(){
   int j = 4;
   int i = 2;
   int *ptr1, *ptr2;
   ptr1 = &j;
   ptr2 = &i;
   printf("%d \n", *ptr1-*ptr2);
   return 0;
}
```

a. ne compile pas
b. provoque une erreur à l'exécution
c. affiche 2 à l'exécution
d. affiche 8 à l'exécution

3. Quelle est la forme équivalente à l'expression **p->champ ?**
   a. p.champ
   b. (*p).champ
   c. Cette expression est erronée
   d. &(p.champ)
   e. *p.champ

4. **Quelle est la différence entre un tableau et une structure ?**
   a. Un tableau peut contenir des données de types différents, tandis qu'une structure ne le peut pas.
   b. Une structure peut contenir des données de types différents, tandis qu'un tableau ne le peut pas.
   c. Tous deux peuvent contenir des données de types différents, mais la structure occupe moins de place mémoire.
   d. Tous deux peuvent contenir des données de types différents, mais la structure permet un accès mémoire plus rapide.

5. Donner le résultat de l'exécution du code suivant :

```
int main() {
int a, b;
int *ptr1, *ptr2;
a = 5;
b = a;
ptr1 = &a;
a++;
ptr2 = &b;
b += a = b;
printf("a = %d, b = %d, *ptr1 = %d, *ptr2 = %d\n", a, b, *ptr1++,
++*ptr2);
return 0;
}
```

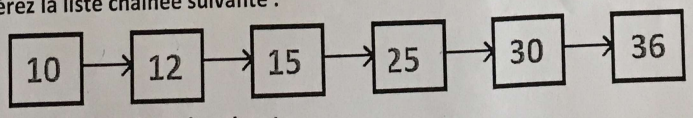
a. ne compile pas
b. provoque une erreur à l'exécution
c. affiche a = 5, b = 11, *ptr1 = 6, *ptr2 = 12
d. affiche a = 5, b = 11, *ptr1 = 5, *ptr2 = 11
e. affiche a = 4, b = 10, *ptr1 = 5, *ptr2 = 11
f. Aucune des propositions ci-dessus.

6. Donner le résultat de l'exécution du code suivant :

```
#include <stdio.h>
#define TAB_LENGTH 4
int main() {
    int tab[TAB_LENGTH];
    int j = 1;
    int *ptr = tab + 1;
    for(; j < TAB_LENGTH; j++)
    {
        tab[j] = 3;
        *(ptr - 1) = 3;
    }
    printf("[ %d %d %d %d ]\n", tab[0], tab[1], tab[2], tab[3]);
    return 0;
}
```

a. ne compile pas
b. provoque une erreur fatale à l'exécution
c. affiche [1 1 3 3]
d. affiche [1 3 4 4]
e. affiche [3 3 3 3]
f. Aucune des propositions ci-dessus.

7. **Considérez la liste chaînée suivante :**



Quel sera le résultat du code suivant :
start->next->next->next->data

a. 15

b. 30
c. 12
d. 25
e. Le code génère une erreur

8. **Considérons le fragment de code ci-dessous :**

```
int tab[]={ 10, 20, 30, 40 };
int *ptr1=&tab[1];
int *ptr2=&tab[3];
```

Dans ce code, une seule des affirmations suivantes est vraie, laquelle ?

a. les expressions *(ptr1-1) et *(ptr2-3) retournent toutes les deux la même valeur, 10
b. l'expression ptr2-ptr1 vaut 20
c. l'expression *(ptr2-ptr1) retourne la valeur 20
d. Ce code génère une erreur

Partie II : Questions directes *(3,5 points)*

1. Le programme suivant contient trois erreurs, lesquelles ? proposez des corrections ***(1,5 pts)*** :

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef struct maillon {
   char nom[80];
   struct t_maillon *suiv;
} t_maillon;

void main()
{
   t_maillon premier;
   premier=(t_maillon*)malloc(sizeof(t_maillon));
   strcpy(premier->nom,"toto");
   premier->suiv=NULL;
   t_maillon *m ;
   m = premier;
   while (m!=0)
   {
      printf("nom : %s\n",m->nom);
```

```
    m=m->suiv;
  }
}
```

2. Considérons les fonctions : ***(2 pts)***

```
int foo(int a) {
    a=a*2 ;
    return a+3 ;
}
```

```
int bar(int *b) {
    *b=(*b)*3 ;
    return *b+7 ;
}
```

```
int foobar(int *c) {
    return foo(bar(c)) ;
}
```

```
int barfoo(int d) {
    int t ;
    t = foo(d);
    return bar(&t) ;
}
```

Donnez la valeur des différentes variables définies ci-dessous (c'est-à-dire les variables a, b, c, d, ra, rb, rc, rd) après exécution du code suivant :

```
int a, b, c, d, ra, rb, rc, rd ;
a = 10; b = 20; c = 50; d = 70;
ra = foo(a);
rb = bar(&b);
rc = foobar(&c);
rd = barfoo(d);
```

Partie III : Exercices de programmation *(10,5 points)*

Exercice 1 - Chiffres, lettres et caractères de ponctuation *(4 pts)*

1. Ecrire une fonction qui compte le nombre de chiffres, de lettres majuscules, de lettres minuscules et de caractères de ponctuation, dans la chaîne de caractères qui lui est passée en paramètre.
   Après appel de la fonction, les 4 valeurs comptées seront récupérées dans les variables entières dont l'adresse aura été donnée en argument de la fonction. Le prototype de la fonction sera donc le suivant :
   void compter (const char *s, int *cpt_chif, int *cpt_lettreMaj*, int *cpt_lettreMin, int *cpt_ponct) ;

Aide : voici quelques fonctions de ***ctype.h*** et ***string.h***

La fonction ***int isalpha(char c)*** retourne 1 si c'est un caractère alphabétique, 0 sinon.
La fonction ***int isupper(char c)*** retourne 1 si c'est une lettre en majuscule , 0 sinon.
La fonction ***int islower(char c)*** retourne 1 si c'est une lettre en minuscule, 0 sinon.
La fonction ***int ispunct(char c)*** retourne 1 si c'est un caractère de ponctuation, 0 sinon.
La fonction ***int strlen(const char *s)*** retourne la longueur de la chaîne passée en paramètre.

2. Ecrire une fonction ***main()*** qui appelle la fonction compter() de la question précédente sur la chaîne de caractère suivante : "Vive la Programmation ! ! ! Et 1, et 2, et 3... ZERO !". Ensuite le programme affichera le nombre de chiffres, de lettres majuscules, de lettres minuscules et de caractères de ponctuation de cette chaîne.

Exercice 2 - Répertoire téléphonique *(6,5 pts)*

On veut gérer un répertoire téléphonique (contenant les noms et numéros de téléphones de personnes).

1. Définir un type de structure PERS qui contient deux champs : le nom d'une personne et son numéro de téléphone. ***(1 pt)***
2. Ecrire une fonction saisir_personne qui permet de saisir une personne (une structure de type PERS). ***(1 pt)***
3. Ecrire une fonction saisir_repertoire qui permet de saisir un tableau de personnes. ***(1 pt)***
4. Ecrire une fonction afficher_repertoire qui permet d'afficher le contenu du répertoire. ***(1 pt)***
5. Ecrire une fonction chercher_personne qui permet de chercher un numéro d'une personne donnée dans le répertoire. ***(1 pt)***
6. Ecrire une fonction main qui saisit un répertoire téléphonique et propose le menu suivant : ***(1,5 pts)***
   a. afficher le contenu du répertoire
   b. chercher le numéro de téléphone d'une personne
   c. quitter le Programme