

L2 INFORMATIQUE - ALGO ET PROG - CC LONG 1 2019-2020

NOM :

PRENOM:

EXO-1 (5 pts)

Dans la suite, vous considérez que les codes présentés sont intégrés dans un programme en C

1 - Que vaut x après l'exécution de la séquence suivante

```
int x;
int *p;
p = &x;
x = 5;
*p += 5;
```

a) 5
b) 10
c) ni 5 ni 10
d) erreur de compilation ou à l'exécution

2 - Qu'est ce que le code suivant affiche si l'utilisateur entre 3 au clavier

```
int x;
scanf("%d", &x);
if (x = 2) {
    printf("b");
}
else if (x == 4) {
    printf("d");
}
else {
    printf("z");
}
```

a) b
b) d
c) z
d) rien

3 - Qu'est ce que le code suivant affiche

```
int *n;
n = malloc(sizeof(int));
*n = 5;
free(n);
n = malloc(sizeof(int));
printf("%u\n", n);
```

a) 5
b) 0
c) on ne peut pas savoir
d) rien

4 - que vaut x après l'exécution de

```
int x;
int tab[4] = {0, 5, 10, 15};
int *p = tab + 1;
x = *p + 2;
```

a) 5
b) 15
c) 7
d) on ne peut pas savoir

5 - Qu'est ce que le code suivant affiche

```
typedef struct {
    int mon_int;
    float mon_float;
} int_float;

int_float trois;
trois.mon_float = 3.0;
printf("%d\n", trois.mon_int);
```

a) 3.0
b) 3
c) on ne peut pas savoir
d) 0

EXO-2 Traitement d'images (10 pts)

Dans la suite, une image (format bitmap en échelle de gris) est représentée par la structure suivante:

```
typedef struct{
    unsigned char **im;
    unsigned int H;
    unsigned int L;
}Image;
```

où L est la largeur de l'image, H sa hauteur et im est un tableau à 2 dimensions, contenant la valeur des pixels en échelle de gris de 0: noir à 255 : blanc, le pixel en haut à gauche de l'image étant im[0][0].

- 1- Ecrire une fonction prenant en paramètre un entier h et un entier l, qui crée une image de taille hxl avec allocation dynamique, initialisée en noir (tous les pixels sont à 0), et qui renvoie l'adresse de l'Image ainsi créée.
- 2- Ecrire une fonction prenant une Image en paramètre et qui renvoie un réel égal à la valeur moyenne des pixels de l'image.
- 3- Ecrire une fonction prenant une Image i et un entier s en paramètre, et qui modifie l'image de telle manière que tous les pixels dont la valeur est inférieure ou égale à s passe à 0, les autres prenant la valeur 255.
- 4- Ecrire une fonction prenant une Image i et un entier s en paramètre, et qui modifie l'image pour la transformer en sa symétrique par une symétrie horizontale.
- 5- Ecrire une fonction prenant une Image i, un entier x, un entier y et un entier r et qui modifie l'image en ajoutant un cercle blanc centré sur (x,y) et de rayon r.

EXO-3 Traitement de chaînes de caractères (10 pts)

Rappels

Les chaînes de caractères sont des tableaux de `char`.

Le caractère de fin de chaîne est le caractère `'\0'`

Les `char` sont des entiers (codés sur 1 octet).

En particulier, `c - 'a'` vaut 0 si c est un 'a' et 26 si c est un 'z'.

Nota Bene : pour cet exercice vous ne devez pas utiliser les fonctions définies dans string.h

- 1- Ecrire une fonction prenant une chaîne de caractères s et qui modifie la chaîne en transformant les minuscules en majuscules
- 2- Ecrire une fonction prenant deux chaînes de caractères s1 et s2 et qui renvoie un entier égal à -1 si s1 est inférieure (au sens de l'ordre du dictionnaire) à s2, 1 si s2 est inférieure à s1, et 0 si elles sont égales.
- 3- Ecrire une fonction prenant une chaîne de caractères s et un char c et qui modifie la chaîne en supprimant toutes les occurrences de c.
- 4- Ecrire une fonction prenant en paramètre une chaîne de caractères s, une chaîne de caractère m et un entier p et qui renvoie 1 si s contient toute la chaîne m à la position p et 0 sinon. (On suppose $p \geq 0$ et $p < \text{longueur de s}$.)

5- Ecrire une fonction prenant en paramètre une chaîne de caractères s et une chaîne de caractère m et qui renvoie 1 si m est présente dans s, et 0 sinon.

EXO-4 Listes chaînées (6 pts)

Soit la structure de données de liste chaînée définie par les structures :

```
typedef struct _cell{
    int valeur;
    struct _cell *suivant;
} Cellule;

typedef struct {
    Cellule *tete;
} Liste;
```

1- Ecrire une fonction prenant deux listes chaînées l1 et l2 et qui renvoie une liste chaînée dont les éléments sont les éléments de l1 et l2 (copie superficielle : les éléments ne sont pas clonés) : le premier élément est le premier élément de l1, le deuxième est le premier de l2, le troisième est le deuxième de l1, etc. Si une chaîne est plus grande que l'autre, ses éléments sont ajoutés à la fin.

2- Ecrire une fonction prenant deux listes chaînées l1 et l2 et qui renvoie 1 si l2 est un préfixe de l1 (les éléments de l2 sont égaux aux premiers éléments de l1).