

## 2 H - Sans documents (sauf diagrammes de C) ni calculatrice

N'oubliez pas de commenter vos programmes.  
Évitez le crayon à papier.

**Vous aurez besoin de 4 copies : une par exercice**

### 1. Questions diverses (5 points)

a) Qu'affiche les `printf` ?

```
int x,y,z ;
int *u,*v ;
x = 25;
y = x/2;
z = 2 ;
u = &x;
printf("\nP1 :  x= %d, y= %d, z= %d, u= %d", x, y, z, *u);
v = u ;
(*v)++;
z*= x + y ;
u = &y ;
y--;
printf("\nP2 :  x= %d, y= %d, z= %d, u= %d, v =%d", x, y, z, *u, *v);
```

b) Écrire un programme qui demande à l'utilisateur combien il boit de tasses de café par jour et affiche son risque d'intoxication selon l'étude suivante :

0 : aucun risque,  
1 à 2 cafés : risque faible,  
3 à 4 cafés : risque de pré-diabète,  
au delà de 5 : risque d'intoxication à la caféine.

c) Soit la boucle `do while` suivante:

```
int i ,n, f;
printf ("entrez 2 nombres positifs SVP");
scanf ("%d %d", &n, &i);
f=1;
do {f*=i;
    n--;}
while (n!= 0);
```

Écrire le code équivalent en utilisant une boucle `while`

### 2. Figures géométriques (5 points) **NOUVELLE COPIE**

Soit 4 fonctions permettant de réaliser les 4 figures suivantes :

<code>carrePlein(10)</code>	<code>carreVide(10)</code>	<code>triangle(10)</code>
***** ***** ***** ***** ***** ***** ***** ***** ***** *****	***** *      * *      * *      * *      * *      * *      * *      * *      * *      *	* ** *** **** ***** ***** ***** ***** ***** *****

1) Écrire la fonction `void carrePlein(int n)` qui affiche un carré plein de côté `n`.

2) Écrire la fonction `void carreVide(int n)` qui affiche un carré vide de côté `n`.

3) Écrire la fonction `void triangle(int n)`, qui affiche un triangle rectangle plein de côté `n`.

5) Écrire le programme principal qui demande à l'utilisateur de rentrer `n` et de spécifier quelle figure il désire afficher parmi les 3 puis l'affiche.

---

### 3. Sommes et suite (5 points) **NOUVELLE COPIE**

#### a) Sommes

Ecrire un programme qui affiche la somme des valeurs absolues des entiers compris entre les entiers relatifs (entiers naturels positifs ou négatifs)  $f$  et  $k$ . Les valeurs de  $f$  et  $k$  sont demandées à l'utilisateur et on s'assurera que  $f$  est inférieur ou égal à  $k$ .

Par exemple si  $f = -5$  et  $k = 3$  alors la somme est égale à  $5+4+3+2+1+0+1+2+3 = 21$

#### b) Suite

On considère la suite:

$$\begin{aligned} u_0 &= a && a \text{ entier} \\ u_{n+1} &= u_n/2 && \text{si } u_n \text{ pair} \\ u_{n+1} &= 3 * u_n + 1 && \text{si } u_n \text{ impair} \end{aligned}$$

Pour toutes les valeurs de  $a$ , il existe un entier  $N$  tel que  $u_N=1$  (hypothèse admise).

Ecrire un programme qui demande à l'utilisateur la valeur de  $a$  et qui affiche toutes les valeurs de  $u_n$  de  $n=0$  à  $n=N$ .

---

### 4. Codage d'un mot (5 points) **NOUVELLE COPIE**

Afin de coder un mot, on voudrait programmer un algorithme simple de cryptographie à clé. Imaginons que l'on veuille encoder le mot "bonjour" et qu'on choisisse comme clé de cryptage le mot "bac". On convertit chaque lettre du mot et de la clé en un nombre ( $a=1, b=2, \dots, z=26$ ), on les additionne et on reconvertit les nombres obtenus en lettre. Comme la clé est souvent un simple mot, on la répète autant de fois que nécessaire pour coder l'ensemble du message.

Exemple :

mot	b	o	n	j	o	u	r
nombre	2	15	14	10	15	21	18
clé	b	a	c	b	a	c	b
nombre	2	1	3	2	1	3	2
addition	4	16	17	12	16	24	20
mot codé	d	p	q	l	p	x	t

Ecrire un programme qui demande la clé de cryptage (composé de 3 lettres) puis un mot terminé par un point, et affiche le mot codé selon le principe décrit ci-dessus. Le dialogue utilisateur pourra avoir la forme suivante :

**entrez un mot de 3 lettres (cle de cryptage) :** bac  
**ecrite votre mot a crypter termine par un point :** bonjour.  
**mot crypte :** dpqlpxt

Voici d'autres exemples :

**entrez un mot de 3 lettres (cle de cryptage) :** bac  
**ecrite votre mot a crypter termine par un point :** buzz.  
**mot crypte :** dvcb

**entrez un mot de 3 lettres (cle de cryptage) :** mer  
**ecrite votre mot a crypter termine par un point :** bonjour.  
**mot crypte :** ofwftme

#### Remarques importantes :

On n'utilisera pas les chaînes de caractères dans cet exercice. La clé sera lue lettre par lettre et chaque lettre sera stockée dans une variable de type caractère. Le mot à crypter sera aussi lu lettre par lettre et chaque caractère sera traité au fur et à mesure de la lecture.

#### Rappel :

On peut utiliser les opérations  $+$  et  $-$  sur les caractères : par exemple  $'c' - 'a' = 2$   $'d' - 'd' = 0$  et  $'u' + 2 = 'w'$ .

Attention : c'est à vous de gérer le codage lorsque celui-ci sort de l'alphabet. En effet, dans cette méthode, on veut que  $'z'+1$  donne  $'a'$ ,  $'c'+25$  donne  $'b'$  etc.