

2 H - Sans documents ni calculatrice (sauf diagrammes de C)
 La clarté de vos réponses sera prise en compte. Ne pas écrire au crayon.
N'oubliez pas de commenter vos programmes.

Vous aurez besoin de 3 copies : une par partie

Partie 1 → copie n°1

1. Affichage

On voudrait afficher la figure suivante :

```
***** Som
*      1      * 1
*     555     * 15
*    88888    * 40
*   222222   * 14
*  99999999  * 81
*88888888888* 88
* 555555555  * 45
*  7777777   * 49
*   33333    * 15
*    333     * 9
*     4      * 4
*****
```

Les chiffres à l'intérieur du cadre sont des nombres aléatoires entre 0 et 9.

La dimension du carré est un nombre demandé à l'utilisateur, sachant qu'elle doit être comprise entre 3 et 21 et qu'elle doit être impaire.

Les nombres sous Som sont la somme des chiffres de la ligne.

Ecrire un programme qui demande à l'utilisateur de saisir la dimension de la figure jusqu'à ce que celle-ci respecte les consignes puis l'affiche.

Rappel : la fonction rand() retourne un entier positif ou nul.

Partie 2 → copie n°2

Nouvelle copie

2.1 Pointeurs (2 points)

Qu'affiche ce programme ?

```
#include <stdio.h>

int main(void)
{ int i,j;
  int *p1,*p2;

  i=7;
  j=i!=0;
  p1=&i;
  p2=&j;
  j++;
  i/=j;
  printf ("i=%d j=%d p1= %d p2 = %d", i,j,*p1,*p2);

  *p2 = i*j;
  *p1 *= j+1;
  p1=p2;
  printf(" i=%d j=%d p1= %d p2 = %d", i,j,*p1,*p2);
}
```

2.2 Analyse d'un texte.

Ecrire un programme qui lit un texte (suite de caractères) terminé par le caractère # et détermine si ce texte est extrait d'un manuel littéraire, de comptabilité ou autre. On dira que l'extrait est littéraire si 60% de caractères sont des caractères alphabétiques (minuscule ou majuscule), qu'un extrait est comptable si 60% des caractères sont des chiffres **et autre sinon**.

2.3 Encadrement d'un réel par 2 entiers

Ecrire un programme qui demande à l'utilisateur une valeur réelle x et affiche les 2 entiers les plus proches qui encadrent cette valeur. Par exemple,

pour $x = 3.14519$, le programme affichera $3 \leq 3.14519 \leq 4$
pour $x = -219.815$ le programme affichera $-220 \leq -219.815 \leq -219$

Remarque importante : Le programme devra utiliser uniquement des conversions explicites (aucune conversion implicite ni fonctions prédéfinies).

Partie 3 → copie n°3

Nouvelle copie

3.1 Table de multiplication (2 points)

Ecrire un programme qui saisit un entier nb , si celui-ci est compris entre 1 et 9 alors le programme affichera la table de multiplication correspondante puis on réitérera le processus, sinon on affichera "Ce n'est pas dans les possibilités du programme, recommencez !". L'utilisateur tapera 0 (zéro) pour **arrêter**.

```
Quelle table ? : 12
Ce n'est pas dans les possibilités
du programme, recommencez !
Quelle table ? : 8
8 * 0 = 0
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
8 * 10 = 80
Quelle table ? : 0
```

3.2 Robot dans un cercle

Un robot situé sur un quadrillage muni d'un repère orthonormé (O, i, j) se déplace d'une case à chaque pas dans une des quatre directions (haut, bas, droite, gauche) en fonction d'une valeur comprise entre 0 et 1. Cette valeur appelée « direction » est générée aléatoirement.

Au départ le robot est en O , à chaque pas, il se déplace vers la droite si la direction est inférieure strictement à 0.2, vers le haut si la direction est comprise entre 0.2 et 0.4, vers la gauche si la direction est supérieure entre 0.4 et inférieure à 0.7 et vers le bas si la direction est supérieure à 0.7.

Le robot s'arrête dès qu'il arrive sur le contour d'un cercle de centre O et d'un rayon égal à R (valeur donnée par l'utilisateur).

- 1) Expliquez en quelques mots la manière dont vous allez calculer la valeur direction à l'aide de la fonction `rand()` qui retourne un entier positif ou nul.
 - 2) Ecrire un programme qui permet de déplacer le robot à partir de sa position initiale, affiche ses coordonnées à chaque étape et affiche le nombre de déplacements lorsqu'il est arrivé sur le contour.
 - 3) Ecrire un programme qui répète le processus précédent nb fois (donné par l'utilisateur) et qui détermine le nombre moyen de déplacements avant l'arrêt du robot ainsi que le pourcentage de fois où le robot arrive sur le demi-cercle positif (ordonnée positive). (Vous n'êtes pas obligés de tout réécrire).
-