Vous aurez besoin de 4 copies : une par exercice

2 H - Sans documents (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.

---

1. Numération (5 points)

1.1 - Ecrire un programme qui demande à l'utilisateur d'entrer un nombre binaire nb (un entier composé des "0" et des "1") et affiche son correspondant en décimal, nb est de type int.

Rappel de la conversion d'un nombre binaire en un nombre décimal :

$$1101_{(2)} \Rightarrow 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 = 13_{(10)}$$
$$10111_{(2)} \Rightarrow 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0 = 23_{(10)}$$

1.2 - Ecrire un programme qui cette fois demande à l'utilisateur un nombre décimal nb (entier positif) et affiche sa représentation en binaire (exemples : 4 en décimal -> 100 en binaire; 19 en décimal -> 10011 en binaire).
Utilisez les opérateurs "%" et "/".

Nouvelle copie

2. Motif (5 points)

On se propose de dessiner un motif carré de côté c=1, ou c=3, ou c=5, ou c=7 ou c=9 caractère(s) (donnée choisie aléatoirement à l'aide de la fonction `rand()`). Le motif sera lui-même composé de caractère(s) astérisque(s) `'*'` et de caractère(s) point(s) `'.'`. Pour c=1, le motif sera composé d'un seul caractère astérisque `'*'`. Créez le programme correspondant, qui permettra d'afficher aléatoirement à l'écran, à chaque lancement, l'un des motifs suivants :

```
Motif pour c=1 caractère :

*

Motif pour c=3 caractères :

***
*.*
***

Motif pour c=5 caractères :

*****
*...*
*.*.*
*...*
*****

Motif pour c=7 caractères :

*******
*.....*
*.*.*.*
*.....*
*.*.*.*
*.....*
*******

Motif pour c=9 caractères :

*********
*.......*
*.*.*.*.*
*.......*
*.*.*.*.*
*.......*
*.*.*.*.*
*.......*
*********
```

Nouvelle copie

3. Une journée de chasse (5 points)

Une nouvelle loi oblige lors d'une partie de chasse de :

ne pas tuer plus de 100 animaux,
ne pas tuer plus de 40 lièvres,
ne pas tuer plus de 8 chevreuils,
ne pas tuer plus de 10 sangliers,
ne pas tuer plus de 30 perdrix.

Si une de ces conditions est atteinte, la chasse s'arrête immédiatement.
Chaque chasseur est muni d'une tablette, qu'il doit renseigner à chaque fois qu'il abat un animal. Ecrire un programme permettant à chaque chasseur, d'afficher l'état de la chasse (nombre de lièvres tués et son pourcentage par rapport au nombre de gibiers tués, nombre de chevreuils tués et son pourcentage par rapport au nombre de gibiers tués, …), et de comptabiliser le gibier qu'il vient d'abattre afin de mettre fin à la partie de chasse.

Ecrire le programme qui demandera au chasseur s'il veut voir l'état de la chasse et l'afficher ou s'il veut entrer un nouveau gibier tué. Lorsqu'un des critères de départ est obtenu, il affichera « ARRET DE LA CHASSE ».
Exemple :

Que voulez-vous faire (V pour visualiser l'état de la chasse, A ajouter un gibier tué) : V
Etat de la chasse :

nombre de gibiers tués	:	66		
nombre de lièvres tués	:	27	soit	40%
nombre de chevreuils tués	:	4	soit	6%
nombre de sangliers tués	:	6	soit	10%
nombre de perdrix tuées	:	29	soit	44%

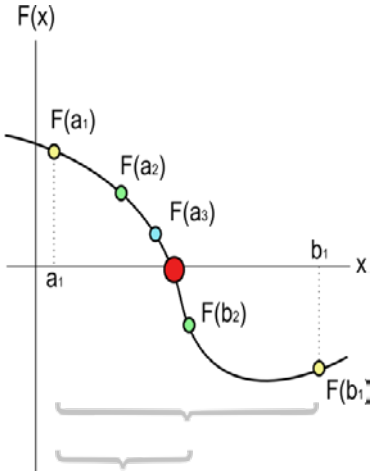
Que voulez-vous faire (V pour visualiser l'état de la chasse, A ajouter un gibier tué) : A
Quel animal avez-vous tué (L : lièvre, C : chevreuil, S : sanglier, P : perdrix) : P
ARRET DE LA CHASSE

Attention aux divisions entières

Nouvelle copie

4. Racines d'un polynôme du 3e degré (5 points)

Un algorithme de recherche d'un zéro d'une fonction $f : \mathbb{R} \to \mathbb{R}$ est un algorithme de recherche d'une valeur approchée d'un $x \in \mathbb{R}$ tel que $f(x)=0$. Un tel x est appelé zéro de f, ou racine de f lorsque f est un polynôme en x.



La méthode de **dichotomie** est un algorithme pour trouver des zéros d'une fonction continue : commencer avec deux points a et $b \in \mathbb{R}$ qui encadrent un zéro de la fonction (qui sont tels que $f(a)$ et $f(b)$ soient de signes opposés). À chaque itération de l'algorithme, les deux intervalles $[a, m]$ ou $[m, b]$, avec $m=(a+b)/2$, sont considérés. Si f(m)=0, l'algorithme s'arrête, sinon le sous-intervalle de $[a, m]$ ou $[m, b]$ qui contient un zéro est alors choisi pour recommencer ce processus dans une itération suivante. L'algorithme s'arrête aussi dès que b-a $<\varepsilon$ où ε est une précision donnée.

Remarque : si $[a,b]$ contient une racine, on doit avoir $f(a)*f(b)<0$.

Ecrire un programme en C, qui demande à un utilisateur de saisir les coefficients c_0, c_1, c_2, c_3 des monômes d'un polynôme du 3e degré (i.e. $f(x)=c_0+c_1x+c_2x^2+c_3x^3$) ainsi que les deux bornes a et b d'un intervalle $[a,b]$. Le programme vérifiera que cet intervalle contient bien une racine du polynôme (ou redemandera à l'utilisateur un autre intervalle dans le cas contraire). Enfin le programme demandera de saisir une précision et cherchera la racine du polynôme dans $[a,b]$.