

## Exercice 1

Tracer et indenter chacun des quatre programmes suivants :

```
/* Programme 1*/

#include<stdio.h>

main(){
int n, B, c;

n=53058; //n comme nombre
B=10; //B comme base

//Point d'observation 1

while(n!=0){
c=n%B; //c comme chiffre
n=n/10; //Point d'observation 2
printf("%i", c);
}
printf("\n");
}

/* Programme 2*/

#include<stdio.h>

main(){
int n, B, c;

n=53058; //n comme nombre
B=10; //B comme base

//Point d'observation 1

while(n!=0)
c=n%B; //c comme chiffre
n=n/B; //Point d'observation 2
printf("%i", c);
printf("\n");
}
```

```

/* Programme 3*/

#include<stdio.h>

main(){
int n, B, c;

n=31; //n comme nombre
B=2; //B comme base

//Point d'observation 1

while(n!=0){
c=n%B; //c comme chiffre
n=n/B; //Point d'observation 2
printf("%i", c);
}
printf("\n");
}

/* Programme 4*/

#include<stdio.h>

main(){
int n, B, c;

n=31; //n comme nombre
B=2; //B comme base

//Point d'observation 1

while(n!=0)
c=n%B; //c comme chiffre
n=n/B; //Point d'observation 2
printf("%i", c);
printf("\n");
}

```

## Exercice 2

### QCM

1. gcc est :
  - a. Un identificateur du langage C
  - b. Une commande qu'on tape dans la fenêtre de commande
  - c. Un mot-clef du langage C
  - d. Un opérateur du langage C

2. `while` est :
  - a. Un identificateur du langage C
  - b. Une commande qu'on tape dans la fenêtre de commande
  - c. Un mot-clef du langage C
  - d. Un opérateur du langage C
3. Dans le programme `main(){int a; a=3%5;}`, `%` est :
  - a. Un identificateur du langage C
  - b. Une commande qu'on tape dans la fenêtre de commande
  - c. Un mot-clef du langage C
  - d. Un opérateur du langage C
4. Dans le programme `main(){int a; a=3%5;}`, `a` est :
  - a. Un identificateur du langage C
  - b. Une commande qu'on tape dans la fenêtre de commande
  - c. Un mot-clef du langage C
  - d. Un opérateur du langage C
5. `=` permet en langage C de :
  - a. réaliser une affectation
  - b. tester une égalité
  - c. comparer deux identificateurs
  - d. convertir un `int` en `float`
6. En typographie le caractère `&` s'appelle :
  - a. La lulette
  - b. L'espagnolette
  - c. L'esperluette
  - d. L'arpette

### Exercice 3

Écrire un programme permettant d'afficher les nombres de 1 à  $n$ , ainsi que leur carré, et leur cube (soit trois colonnes). L'entier  $n$  est au choix de l'utilisateur. Écrire ce programme trois fois : avec `while`, avec `do ... while` et avec `for`.

### Exercice 4

Écrire un programme qui demande deux entiers  $a$  et  $b$  à l'utilisateur et indique en retour si  $b$  divise ou non  $a$ .

### Exercice 5

Écrire un programme qui lit une note  $n$  sur 20, puis affiche la mention correspondante : ajourné si  $n < 10$ , passable si  $10 \leq n < 12$ , assez bien si  $12 \leq n < 14$ , bien si  $14 \leq n < 16$ , et très bien sinon.

## Exercice 6

Écrire un programme qui demande 3 nombres à l'utilisateur, et les affiche en retour, mais triés par ordre croissant. Idem pour 4 nombres.

## Exercice 7

On décrit le jeu des allumettes : au départ, il y a un tas de 50 allumettes, (ou tout autre objet : cailloux, jetons, ...). Chacun à son tour, les deux joueurs ôtent obligatoirement entre 1 et 6 allumettes. Celui qui ôte la dernière allumette gagne.

1. Préférez-vous commencer ou jouer en deuxième ? Justifiez votre choix par une stratégie gagnante. Conseil : commencer par raisonner avec un petit nombre d'allumettes, puis généraliser.
2. Écrire un programme qui joue au jeu des allumettes contre l'utilisateur.

## Exercice 8

Exercice de révision : jusqu'où irez vous ? À chacune des questions suivantes, on demande d'écrire un programme. On conseille à chaque fois d'enregistrer, de compiler et d'exécuter le programme, de toujours travailler sur le même fichier et de faire "grossir" le programme de questions en questions.

Si vous voulez vous auto-évaluer : faites ce test en salle machine. Comptez deux points pour les questions 1 et 2, trois pour les suivantes. Ca vous fera une note assez approximative sur 19.

1. Écrire un programme qui affiche "Bonjour".
  2. Faire en sorte que le programme demande à l'utilisateur d'entrer un nombre de son choix. Le programme doit ensuite afficher : "Le nombre que vous avez demandé est : *valeur*".
  3. Le programme doit ensuite afficher un message indiquant si le nombre est oui ou non divisible par 7.
  4. Le programme doit maintenant afficher la liste des diviseurs du nombre initialement entré. Par exemple, si l'utilisateur entre initialement le nombre 12, le programme doit écrire : "Les diviseurs de 12 sont 1, 2, 3, 4, 6 et 12".
  5. On rappelle qu'un nombre est premier s'il n'est divisible que par 1 et par lui-même. Par convention, 1 n'est pas considéré comme un nombre premier. Le programme doit indiquer si le nombre entré par l'utilisateur est ou non premier.
  6. Le programme doit maintenant afficher la liste de tous les nombres premiers inférieurs au nombre entré par l'utilisateur.
  7. Le programme doit maintenant afficher seulement les nombre premier jumeaux. Deux nombres premiers sont dits *jumeaux* si leur différence est égale à 2 (ou à  $-2$ ).
- Remarque : à l'heure actuelle, personne ne sait s'il existe un nombre fini ou infini de nombres premiers jumeaux.