Exercice 1:

Qu'affiche le programme C suivant :

```
#include <stdio.h>
int main (void)
{
  int i;
  i=-3;
  while (i){
    printf("La valeur de i est %d. \n", i);
    i++;}
return(0);}
```

Exercice 2:

Grâce à l'instruction switch, écrire un programme C qui lit un caractère (qui représente une lettre en hexadécimal) et affiche son équivalent (un entier) en décimal. Par exemple, si l'utilisateur rentre le caractère 'A' le programme affichera 10. Si l'utilisateur rentre un caractère qui n'appartient pas {'0', ... , '9', 'A', 'B', 'C', 'D', 'E', 'F'} l'entier -1 est affiché.

Exercice 3:

- Ecrire un programme C qui compte le nombre de caractères dans texte saisi (caractère par caractère) sur une seule ligne.
  - Par exemple, si le texte saisi est : "Bonjour je travaille."
  - le programme affichera : 21
- Reprendre l'exercice précédant en ne déclarant qu'une seule variable (de type unsigned char).

Exercice 4:

Ecrire un programme C qui affiche les n premiers nombres parfaits. Tester votre programme avec n=3, n=5 puis n=9.

- Si n=3 le programme affichera :

```
1  6=1+2+3
2  28=1+2+4+7+14
3  496
```

- Si n=5 le programme affichera :

```
1  6=1+2+3
2  28=1+2+4+7+14
3  496
4  8128
5  33550336
```

- Si n=9 le programme affichera :

```
1  6=1+2+3
2  28=1+2+4+7+14
3  496
4  8128
5  33550336
6  8589869056
7  137438691328
8  2305843008139952128
9  2658455991569831744654692615953842176
```

Exercice 5:

Un étudiant a écrit le programme C ci-dessous. Il s'est rendu compte qu'il avait fait 7 erreurs de syntaxe (de compilation). Corriger ces erreurs puis compiler et tester votre programme.

```
/****
 Les sept erreurs de compilation
 ***/

#include <stdio.h>
int main (void)
{
  int nombre, somme, i,
  somme=0:
  printf (Merci d\'introduire un nombre entier positif : ");
  scanf("%d", &nombre;
  for (i=0; nombre!=0; i++)
  {
    somme = somme + (nombre % 10);
    nombre = nombre / 10;

  printf (La somme des chiffres de ce nombre est : %d\n", somme);
  return(0);
```

Exercice 6:

- Écrire un programme qui lit, caractère par caractère, une séquence de lettres différentes, puis les affiche dans l'ordre croissant (d'abord les minuscules, ensuite les majuscules). La lecture s'arrête lorsque l'utilisateur rentre le caractère '$'.
  - Par exemple, si l'utilisateur entre "EaxcDA$", le programme affichera "acxADE".
- Modifier le programme précédent de manière à générer une erreur si l'utilisateur rentre deux fois le même caractère.

Remarque : L'utilisation des tableaux, des pointeurs, des chaînes de caractères et de la récursivité n'est pas autorisé dans cet exercice. Il n'est pas permis d'utiliser plus de 5 variables pour résoudre cet exercice.

Exercice 7:

Qu'afficherait le programmes ci-dessous (écrit par un étudiant qui a oublié de respecter les styles d'indentation)?

```
int a=-12, b=8;
if (a>0)
 if (b>0) printf ("%d \n", 3*a);
 else printf ("%d \n", 4*a);
```

- Qu'afficherait la suite d'instructions ci-dessous avec :
  - a=12, b=-2
  - a=12, b=2
  - a=-12, b=2

Exercice 8:

On a demandé à un étudiant de réaliser l'exercice suivant :

Grâce à l'instruction switch, écrire un programme C qui lit un caractère (qui représente une lettre en hexadécimal) et affiche son équivalent (un entier) en décimal. Par exemple, si l'utilisateur rentre le caractère 'A' le programme affichera 10. Si l'utilisateur rentre un caractère qui n'appartient pas {'0', ... , '9', 'A', 'B', 'C', 'D', 'E', 'F'} l'entier -1 est affiché.

L'étudiant a écrit le programme suivant :

```
#include <stdio.h>
int main (void)
{
  char h;
    printf("Merci d'introduire la lettre en hexadécimal : ");
    scanf("%c", &h);
    switch (h) {
        case 0:
            printf("Le nombre décimal associé est : %d\n", 0);
            break;
        case 1:
            printf("Le nombre décimal associé est : %d\n", 1);
            break;
        case 2:
            printf("Le nombre décimal associé est : %d\n", 2);
            break;
        case 3:
            printf("Le nombre décimal associé est : %d\n", 3);
            break;
        case 4:
            printf("Le nombre décimal associé est : %d\n", 4);
            break;
        case 5:
            printf("Le nombre décimal associé est : %d\n", 5);
            break;
        case 6:
            printf("Le nombre décimal associé est : %d\n", 6);
            break;
        case 7:
            printf("Le nombre décimal associé est : %d\n", 7);
            break;
        case 8:
            printf("Le nombre décimal associé est : %d\n", 8);
            break;
        case 9:
            printf("Le nombre décimal associé est : %d\n", 9);
            break;
        case 'A':
            printf("Le nombre décimal associé est : %d\n", 10);
            break;
        case 'B':
            printf("Le nombre décimal associé est : %d\n", 11);
            break;
        case 'C':
            printf("Le nombre décimal associé est : %d\n", 12);
            break;
        case 'D':
            printf("Le nombre décimal associé est : %d\n", 13);
            break;
        case 'E':
            printf("Le nombre décimal associé est : %d\n", 14);
            break;
        case 'F':
            printf("Le nombre décimal associé est : %d\n", 15);
            break;
        default:
            printf("Le nombre décimal associé est : %d\n", -1);
            break;
          }
     return(0);
}
```

L'étudiant pense que son programme n'est pas juste.

- Pouvez-vous l'aider à trouver son erreur ?
- Proposer une solution avec un minimum d'appels à la fonction printf et sans introduire de nouvelles variables.

Exercice 9:

Ecrire un programme C qui réalise le jeu des allumettes suivant :

- On dispose de n allumettes (n est un nombre strictement positif à lire depuis le clavier).
- Nous avons deux joueurs : j_0 et j_1.
- Chaque joueurs doit retirer 1, 2 ou 3 allumettes.
- Les joueurs jouent à tour de rôle. On suppose que c'est j_0 qui commence.
- Le joueur qui retire la (ou les) dernière(s) allumettes a perdu.

Le programme affichera le joueur gagnant.

Exercice 10:

Les fourmis et les éléphants souhaitent vivre en harmonie, chacun ayant un toit bien défini.

Toutefois, lorsqu'ils se promènent, les fourmis confondent les pieds des éléphants avec des troncs d'arbres, et finissent par pincer les éléphants à chaque rencontre. De leur côté, les éléphants, sans faire attention, écrasent régulièrement les pattes des fourmis.

Le chef des éléphants (le colonel Hathi) et la représentante des fourmis (la princess Atta) vous sollicitent pour trouver une solution. Ils se demandent s'il serait possible de tracer une bordure (sous forme d'une ligne blanche) qui séparerait les toits des éléphants d'un côté, et ceux des fourmis de l'autre.

Pourriez-vous les aider en écrivant un programme en C qui répond à leur demande ?

- Pour simplifier, nous supposons que les toits sont représentés par des points $t_i = (x_i, y_i)$ sur un plan avec $x_i > 0$ et $y_i > 0$.
- Les données en entrée de vous programme sont donc deux ensembles de points : $\mathcal{E}$, représentant les toits des éléphants, et $\mathcal{F}$, représentant les toits des fourmis.
- La sortie est une valeur booléenne : vraie s'il existe une telle ligne, et fausse dans le cas contraire.