

□ Exercice 1 : Déclarations

1. Ecrire les instructions permettant de :
 - a) Déclarer une variable `n` de type entier.
 - b) Déclarer une variable `x` de type flottant initialisée à 1.
 - c) Déclarer une variable `test` de type booléen, quelle librairie est nécessaire ?
 - d) Déclarer un tableau de 5 entiers initialisés aux valeurs {1, 4, 9, 16, 25}
 - e) Déclarer une chaîne de caractères initialisée à « *Trop bien le langage C* » ;
 - f) Déclarer une variable de type `char` contenant le caractère \$.
2. Ecrire les signatures des fonctions suivantes :
 - a) `divisible_par` qui prend en argument deux entiers n et p et renvoie un booléen.
 - b) `somme` qui prend en argument un tableau de flottant et renvoie un flottant.
 - c) `carre` qui prend en argument un entier renvoie un entier.
 - d) `affiche` qui prend en argument un booléen et ne renvoie rien.

□ Exercice 2 : Comportements indéfinis

1. Rappeler ce qu'est un *comportement indéfini* en C.
2. Donner au moins un exemple de programme ayant un comportement indéfini.

□ Exercice 3 : Opérations sur les types de bases

1. On considère le programme suivant :

```

1  #include <stdio.h>
2
3  int main()
4  {
5      float r = 5 % 3;
6      printf("Résultat = %f\n", r);
7      char c = 'A';

```

- a) Quel est le résultat produit ? Expliquer.
 - b) Que se passe-t-il si on change la ligne 5 en `float r = 5.0 % 3.0;`
2. Ecrire une expression booléenne valant vraie si `a` est égale à `b+c` ou si `c` est non nul et que `a` est divisible par `c`. L'évaluation de cette expression génère-t-elle une erreur si `c` est nul ?
 3. On suppose déjà déclarée la variable `char c = 'A';`. Quel est le résultat de l'instruction suivante : `printf("%c \n", c+1);`

□ Exercice 4 : Portée

On considère le programme C suivant :

```

1  #include <stdio.h>
2
3  const float pi = 3.1415;
4  int k = 1;
5
6  int main() {
7      float s = 0;
8      int k = 1;
9      while (pi * pi / 6 - s > 0.25) {
10         float v;
11         v = 1.0 / (k * k);
12         s += v;
13         k = k + 1;
14     }
15     return 0;
16 }
17

```

1. Pour chacune des variables du programme, indiquer si elle est globale ou locale et donner sa portée.
 2. Déterminer la valeur de chacune des variables existantes juste avant l'instruction `return` de la ligne 15.
- ☒ On peut utiliser une calculatrice !

□ Exercice 5 : *Conversion*

Déterminer le type et la valeur des expressions suivantes. Indiquer lorsqu'une conversion implicite ou explicite a eu lieu.

1. `!(5<7)`
2. `3 + 0.14`
3. `(int)7.5 + (int)12.3`
4. `7.0 / 2`
5. `(true || false) && (false || true)`
6. `(int) 19.6 % 4`

□ Exercice 6 : *Analyser un programme*

On considère le programme suivant :

```

1  #include <stdio.h>
2
3  void echange(int a, int b)
4  {
5      int temp = a;
6      a = b;
7      b = temp;
8  }
9
10 int main()
11 {
12     int a = 12;
13     int b = 50;
14     echange(a, b);
15     printf("a = %i\n", a);
16     printf("b = %i\n", b);
17     return 0;
18 }
```

1. Quel sera le résultat de l'exécution de ce programme ? Pourquoi ?
2. Quel sera l'affichage produit si on déplace l'affichage des variables `a` et `b` dans la fonction `echange` ? Pourquoi ?

□ Exercice 7 : *Programmes à commenter*

Que penser des programmes suivants (erreurs, avertissements, comportements indéfinis, ... ?)

- Programme A :

```

1  #include <stdio.h>
2  int main()
3  int tab[5] = {42}
4  for (i=0;i<=5;i++){
5      printf("%d \n",tab[i]);
6  }
```

- Programme B :

```
1  #include <stdio.h>
2  int main(){
3  int tab[5] = {42}
4  int i = 0;
5  while (true)
6  {
7      printf("%d \n",tab[i]);
8      i = i + 1
9      if (i==5) {break;}
10 }
11 }
```

- Programme C :

```
1  #include <stdio.h>
2  int programme() {
3      float s=0;
4      for int (i=0;i<10000;i++)
5      {
6          s = s + 1/i;
7      }
8      print("somme =%f\n",%s);
9  }
```

- Programme D :

```
1  #include <stdio.h>
2
3  int fonction(int a)
4  {
5      int b = a + 1;
6      printf("Valeur de b = %d\n", b);
7  }
8
9  int main()
10 {
11     int c = fonction(2);
12     printf("valeur de c = %d \n", c);
13 }
```

□ **Exercice 8** : *A propos de ++*

1. Rappeler le type auquel il s'applique et le rôle de l'opérateur ++
2. On suppose déjà déclarée une variable `int a = 42;`.
Que vaut b si on a écrit `int b = a++ + a;` ?
Détaillez votre raisonnement.