La syntaxe et les entrées-sorties...

▸ **Exercice 1**. Ecrire un programme qui lit un caractère sur l'entrée standard, l'affiche sur la sortie standard et passe à la ligne.

▸ **Exercice 2**. Ecrire un programme qui demande à l'utilisateur de rentrer un caractère; le programme devra alors afficher le caractère ainsi que son code ASCII.

▸ **Exercice 3**. Refaire le même exercice, mais cette fois le programme devra afficher le caractère et son code ASCII seulement si c'est une minuscule[1]. Si c'est une majuscule, on la transformera en minuscule puis on fera le même travail. Enfin, si c'est autre chose, on affichera juste son code ASCII.

▸ **Exercice 4**. Donner une **fonction** qui renvoie le minimum entre deux entiers passés en argument.

▸ **Exercice 5**. En déduire le programme complet de calcul du minimum de deux entiers lus sur l'entrée standard.

▸ **Exercice 6**. Ecrire une fonction qui teste l'égalité de deux entiers.

▸ **Exercice 7**. Que dire de la fonction suivante ?

```
int biz(int a, int b)
{ int c=a;
  if (a=b)
  { b=c;
    printf("0\n0\n");
  }

  else
  { printf("%d\n",b);
    a*=2;
    printf("%d\n",a);
  }
  return a==0;
}
```

Les boucles...

▸ **Exercice 8**. Exprimer avec un autre type de boucle, les instructions suivantes : on précisera par ailleurs la valeur de la variable `i`[2] à la sortie de chaque boucle.

```
for(i=5;i>0;i=i-1){...}

i=5;for(;i>0;i-=1){...}

i=10;while(i>0 || i%2) i-=1;

i=10;while(i>0 && !(i%2)) i-=1;

i=0;do { i=i+1; ...; } while(i!=5);
```

▸ **Exercice 9**. Ecrire un programme qui compte et affiche le nombre de caractères lus sur l'entrée standard.

(1) On rappelle que les minuscules sont stockées dans le code ASCII de manière contigüe et dans l'ordre alphabétique.
(2) On supposera que `i` est une variable entière signée.