

1. Que font les fonctions m1 et m2 suivantes ?

```
1 int m1(int a)
2 {
3     return a*((a<0)-(a>0));
4 }
```

```
1 int m2(int a, int b)
2 {
3     if (a+=b)
4         return a;
5     return b;
6 }
```

2. Écrire une fonction `swap_char()` ; (prototype à compléter) qui échange les contenus de deux variables de type `char`.
3. Que fait la fonction `m3()` suivante ?

```
1 char * m3(char * s)
2 {
3     int i;
4     int again;
5     do{
6         again=0;
7         for (i=0;s[i+1]!='\0';i++)
8             if (s[i]>s[i+1])
9                 {
10                    swap_char(&s[i],&s[i+1]);
11                    again=1;
12                }
13    }while(again);
14    return s;
15 }
```

4. Réécrire `m3` en utilisant des pointeurs au lieu des indices entiers.
5. Réécrire la fonction `int strcmp(char*s1, char*s2)` (qui renvoie -1,0 ou 1 selon que `s1` soit plus petite, égale ou plus grande dans l'ordre lexicographique que `s2`) en utilisant des pointeurs pour parcourir les chaînes au lieu d'entiers pour indexer les caractères.
6. Écrire une fonction `arraysort` qui trie par ordre lexicographique un tableau de chaînes de caractères.
7. Écrire un programme qui affiche ses paramètres dans l'ordre lexicographique.
8. Écrire une fonction `void print_binary(int)` ; qui prend en paramètre un entier positif et qui affiche son écriture binaire (en base 2).
 - (a) en itératif avec un tableau temporaire de taille 32.
 - (b) modifier pour utiliser un tableau temporaire de la bonne taille (allocation dynamique!)
 - (c) (bonus) proposer une version récursive.