Exercice 5

On souhaite implémenter un algorithme pour trier des valeurs entières de manière totalement inefficace : pour trier n valeurs v_k (avec $1 \leq k \leq n$), le programme qu'on vous demande de rédiger doit créer n processus fils. Chaque processus fils k se voit attribuer la valeur v_k : il doit attendre v_k unités de temps avant d'afficher la valeur.

Le programme doit admettre l'une des deux syntaxes suivantes :

- `sleepsort -r [-m max] n`
  Avec l'option `-r`, les n valeurs v_k doivent être déterminées pseudo-aléatoirement dans l'intervalle [0,n]. Si l'option `-m` est indiquée, les n nombres doivent être dans l'intervalle [0,max].
- `sleepsort` v_1 `...` v_n
  Sans le `-r`, les valeurs à trier doivent être fournies sur la ligne de commande.

Votre programme doit déterminer les valeurs v_k à partir de l'une des deux méthodes ci-dessus, puis il crée les n processus fils. Chaque processus fils attend $\frac{v_k}{10}$ secondes, puis affiche v_k et se termine avec le code de retour v_k. Une fois tous les processus fils terminés, le père affiche la somme des codes de retour des fils.

Quelques éléments vous sont utiles :

- vous utiliserez la fonction de bibliothèque `rand` pour déterminer une valeur pseudo-aléatoire, en vous assurant que les valeurs sont réparties à peu près uniformément dans l'intervalle donné ;
- vous utiliserez la fonction de bibliothèque `usleep` pour attendre la durée spécifiée ;
- vous utiliserez la fonction de bibliothèque `getopt` pour analyser les options. Notamment, vous accepterez toutes les manières usuelles d'indiquer des options à une commande, comme par exemple :
  - `sleepsort -r -m 10 20`
  - `sleepsort -rm 10 20`
  - `sleepsort -rm10 20`
  - `sleepsort -m 10 -r 20`
  - `sleepsort -m10 -r -- 20`
  - `sleepsort -r -- 20`
  - etc.

Le r ne prend pas d'arguments, le 20 est toujours present a la fin si on a fourni -r

- vous vérifierez qu'aucune valeur ne dépasse 255 : si ce n'est pas le cas, cela constitue une erreur ;
- le temps pris par votre programme doit être (à la précision de la mesure près) à $\frac{\max v_k}{10}$ secondes ;
- voici deux exemples de résultats de votre programme :

```
> ./sleepsort 5 20 7
5
7
20
32
```

```
> ./sleepsort -rm 10 3
4
8
9
21
```