E02

The program is conceptually equivalent to the one presented in E01, although this time the workload is balanced between two children processes. After having read the two integer arguments and parsed them, the program creates `NUM_CHILDREN` children processes (by means of the `fork()` function), and assigns an unique identifier `id` to each one, in order – this means that the first child will receive an `id=1` and so on.

Using the id variable, we distinguish between the parent process and the two children: the parent process (`id=0`) simply waits for the termination of the children (`wait()`) and collects their status using the macro `WEXITSTATUS`.

On the other hand the two children processes – one with `id=1` and the other with `id=2` – will perform the same operations on the two vectors `v1` and `v2` (initialize, fill, sort) and will create two files each, in a concurrent fashion.