Parallel and Real-Time Programming
1. Exercise

Parallelization

Let $n \geq 1$ be a natural number which for $k \geq 0$ neither has the form $9k + 4$ nor $9k + 5$. These are the numbers $\{1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 15, 16, \dots\}$.

A statement, not proven until now, says that for all these n three integers $a, b, c \in \mathbb{Z}$, $|a| > 0$, $|b| > 0$, $|c| > 0$ can be found such that

$$n = a^3 + b^3 + c^3.$$

Simple examples are $1 = 1^3 + (-1)^3 + 1^3$, $29 = 3^3 + 1^3 + 1^3$.

Write an Ada program which for the numbers $n = 1, \dots, r$ finds tuples a, b, c such that $n = a^3 + b^3 + c^3$. The number r shall be given to the program as a shell argument. The structure of the program shall be such that the search for the tuples is parallelized, i.e., Ada tasks are used which synchronize themselves accordingly. (A solution without tasks is not allowed.) Even for one fixed n several tasks shall be used. The number of tasks used shall also be given to the program via an argument.

Example for syntax:

```
$ sum_of_cubes 33 8
```

finds solutions for $n = 1, \dots, 33$ using 8 tasks.

Empirically find the number of tasks on your computer for which the solutions are delivered in the shortest time.