Task 2

Report

Liubov Mikhailovna 60204 and Jorge Fresco 60209

Concurrency

- **Concurrency is ensured by modeling each conveyor belt and truck as independent Erlang processes** using spawn, with all processes operating concurrently. The package generator (*generator_loop*), conveyor belts (*conveyor_loop*), and trucks (*truck_loop*) work in parallel without interference.
- Each conveyor belt has an independent queue of packages received from the generator, communicating with trucks via message passing, while trucks maintain their own state (remaining space), accepting packages until full and replacing themselves instantly by resetting their remaining space.

Deadlock-Free Operation

- The system avoids deadlocks by ensuring that all processes are non-blocking, with asynchronous message handling and making sure no packages are lost.
- **Conveyor belts** never wait indefinitely; they either periodically process their queue by sending a package to its truck to be loaded or buffer new received packages for future processing in their queue.
- **Trucks** are always ready to receive packages and automatically replace themselves when full, ensuring that all received packages are always loaded into it without any delay.

Progress Guarantee

- **Package generator** continuously creates packages at a regular interval and sending them to a randomly chosen conveyor belt.
- **Conveyor belts** are always working either periodically processing their queue or sending packages to their assigned truck.
- **Trucks** continuously handle incoming packages and replace themselves instantly when full so it can continue handling incoming packages without a delay.
- This all ensures that all components are always active, and every package follows a logical flow of first being generated by the package generator, sent to a conveyor belt and finally loaded into a truck

Message Passing

- All interactions between components or even to themselves occur via message passing:
  - **Generators → Conveyors**: {new_package, Package}
  - **Conveyors → Trucks**: {load_package, Package}
  - **Conveyors → Conveyors**: {process_queue}