The problem solving process involves:

1. Understanding the problem

The problem:

A&B Company Ltd. is a company located in Accra that deals with spare parts of cars. The company sells various car parts such as ball joints, engines, gearbox, mirrors, light, etc. In a day the company is able to sell to its cherished customers more than 100 spare parts. You have been contracted as a developer to **build a point-of-sale system for the company**. Develop **an algorithm that can handle all the sales of the company.** Develop a **simple C++ application to calculate the average sales for each day, week and month.**

1. Understanding the problem requirements.

The program requires interaction with the user. The program manipulates data and produces output.

Output is numerical. Floating point to be specific.

Data is numerical. Floating point as well. The data includes sales.

1. Breaking it down.

The output is average of the sales for every day, week and month. Results will be generated as such. Text will be included. Problem is producing an output of the mean sales for the days.

Then the week. Then the month.

1. Design an algorithm

**Algorithm**

1. Initialize necessary variables and data structures:

* Create a data structure (e.g., a list or an array) to store daily sales.
* Create variables to store weekly and monthly totals and counts.
* Create variables to store the current day, week, and month.

1. Create a loop to simulate daily sales:

* Prompt the user to enter the number of spare parts sold for the day.
* Add the daily sales to the list of daily sales.

1. Calculate the daily average:

* Sum up the daily sales for the current day.
* Divide the daily total by the number of daily sales to get the daily average.

1. Update the weekly and monthly totals and counts:

* Check if the current day is the end of the week or month.
* If it is, calculate the weekly and monthly averages:
* Sum up the sales for the current week and month.
* Divide the weekly and monthly totals by the corresponding counts to get the averages.
* Reset the weekly and monthly totals and counts.

1. Repeat the loop for each day, updating the averages and totals accordingly.
2. Provide an option for the user to view the average sales for the current day, week, and month as needed.

**Pseudocode**

Initialize variables and data structures

daily\_sales\_list = []

weekly\_total = 0

weekly\_count = 0

monthly\_total = 0

monthly\_count = 0

Repeat for each day:

Prompt user for daily sales

Add daily sales to daily\_sales\_list

Calculate daily\_average = sum(daily\_sales\_list) / length(daily\_sales\_list)

If it’s the end of the week:

Calculate weekly\_average = weekly total / weekly count

Reset weekly\_total and weekly\_count

If it’s the end of the month:

Calculate monthly\_average = monthly\_total / monthly\_count

Reset monthly\_total and monthly\_count

Display daily\_average, weekly\_average, and monthly\_average if needed

Next step: Algorithm can be implemented in a programming language like C++.