



Kristianstad  
University  
Sweden

Kristianstad University  
SE-291 88 Kristianstad  
Sweden  
+46 44 250 30 00  
[www.hkr.se](http://www.hkr.se)

# Analysis Report

## The one with the travel company

Anton Carlsson

June 15, 2022

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Step 1</b>	<b>1</b>
2.1	Rephrasing the problem . . . . .	1
2.2	Understanding the words . . . . .	1
<b>3</b>	<b>Devising a plan</b>	<b>2</b>
3.1	Approach of choice . . . . .	2
3.2	Additional comment . . . . .	5

# **1 Introduction**

This report aim to solve a simple problem utilizing Polya's Problem Solving Technique. This method consists of first gathering an understanding of the problem, then a plan of how to solve it is devised. These steps constitutes the main focus of this report. Additionally with a plan at hand the last steps consists of execute according to the plan and reflecting whether the strategy sufficed to achieve a suitable solution to the problem.

## **2 Step 1**

### **2.1 Rephrasing the problem**

The task consists of developing a script that manages a ticket system for a travel company. After running the script the user is prompted to make a choice out of three different types of tickets with varying costs. After a user has selected a ticket by typing it's number followed by the key "Enter", the user shall then be prompted with a new menu of selections. This menu consists of five options where up to one bag can be added for a fee of 200 kr, or removed and likewise for a meal equalling 150 kr. The fifth choice consists of finalizing the order in which a summary of the order is printed to the user.

### **2.2 Understanding the words**

The description of the problem was crystal clear and no words were difficult to understand. No research was required to solve this problem other than some inspiration from some previous scripts that's been written by the author of this report.

## 3 Devising a plan

### 3.1 Approach of choice

The idea is to split the script up into different parts, different functions that combined accomplishes the desired output. The advantage of this is to break the problem up into smaller parts that are more manageable and get more structure over the script. For this case with the given problem a suitable way to demonstrate the approach could be with the help of pseudo code. Pseudo code is not the main focus of the report, but hopefully the following snippets will be clear enough for a reader to understand and simultaneously depict the thought process for solving the problem. The main function of the problem could be structured as:

```
# Script overlay pseudo
```

```
def main():
    """Main function to run the script."""
    display_greeting_menuue()          (Function to print the
                                        startup menu.)
    choice := input("Your choice >> ") (String representation of
                                        a number 1-3 given by
                                        user.)
    generate_ticket(choice)             (Takes in a string choice
                                        and generates a ticket
                                        for the user.)
    initiate_session()                 (Updates the users order
                                        status and allows the
                                        user make further
                                        choices.)
```

The function *initiate\_session()* prints an update view of the order status, asks the user to make a new choice (1-5). Then the choice is used as input for the function *run\_program()* shown below.

```
def run_program(choice):
    """Takes in a string choice and determines the right
    operations based on the users choice."""
    global nbr_bags
    global nbr_meals

    if choice == "1"
        nbr_bags := 1
        initiate_session()
    elif choice == "2"
        nbr_meals := 1
        initiate_session()
    elif choice == "3"
        nbr_bags := 0
```

```

        initiate_session()
    elif choice == "4"
        nbr_meals := 0
        initiate_session()
    elif choice == "5"
        print_receipt(ticket)
    else:
        choice = input(
            ("Error not a valid choice! Please choose a number
between 1 and 5 >> ")
        )
        run_program(choice)          (Recursively runs the program
                                     until the user decides
                                     to finalize the order)

```

This strategy utilizes a few global variables, this instead of using local ones that gets passed as input and or output through the functions. This can sometimes result in less writing and more readability, but the solution works the same. If the problem is solved according to the devised plan above, the expected output could look something like that of figure 1.

```
Ticket types:
1. Budget ( 500kr)
2. Economy ( 750kr)
3. VIP ( 2000kr)

Your choice >> 2

Currently you have:
    0 bag(s) registered
    0 meal(s) registered

Here are your options:
1. Add bag (max 1)
2. Add meal (max 1)
3. Remove bag
4. Remove meal
5. Finalize ticket
Your choice >> 1

Currently you have:
    1 bag(s) registered
    0 meal(s) registered

Here are your options:
1. Add bag (max 1)
2. Add meal (max 1)
3. Remove bag
4. Remove meal
5. Finalize ticket
Your choice >> 5

Receipt:
Ticket   : 750kr
Bag      : 200kr
         -----
Total    : 950kr
```

Figure 1: Output from running the script and choosing an economy ticket as well as one bag.

### **3.2 Additional comment**

Note that some bugs such as the fact that no new line is printed after the order menu to separate from the users choice, was not fixed since the solution video had the same bug. The goal with this report is to practice solving a problem using the Polya's Problem Solving Technique and to get the output as close as possible to the solution example video, but the first thing that comes to mind for improving the script would be this. Other improvements could be to change the script so if a user types "1.0" instead of "1", the first choice will still be selected and perhaps adding the choice to choose multiple bags and meals.