



Kristianstad  
University  
Sweden

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

## **Analysis report**

### *Task 3 - The one where we help the police*

***Sandra Kaljula***

## Introduction

This report shall be written by the student as he or she reflects on selected parts from the first two steps in [Polya's Problem Solving Technique](#). The report is intended to guide the student in devising a plan that will help the student solve the problem at hand.

The report shall be written in a formal way, void of any personal opinions. An easy to digest article about writing an academic report can be found [here](#). The student can choose to write the report in Swedish or English.

The student shall:

- Add the name of the task
- Add his or her name
- Remove any italic text in the template
- Rephrase the problem
- Reflect on difficult words
- Explain how the plan was devised

## **Step 1 - Understand the problem**

### **Rephrase the problem**

The script starts with a menu where the user can choose to input several files and then execute the rest of the script. First the text files with the phone numbers must be read and cross-referenced, then returned. Then another function called `map_numbers_to_names` will be run where the program connects the numbers to the suspects. Lastly, in the `display_suspect` function the suspect names will be displayed on the screen. If there were no matches text - “No matches” will be shown.

### **Understanding the words**

There was one sentence in the lab description that I misunderstood at first. “The function shall create, and return, a Set containing all the phone numbers that occurs in all files.”. After reading that I thought that I would just have to get all the numbers from all the files and not cross-reference them.

## Step 2 - Devising a plan

Pseudocode

**Def display\_menu():**

Prints all the menu options on the screen

**Def cross\_reference():**

Create a list for all the phone numbers and a set to store the suspect numbers to

Open and read files and their lines with a for loop, save the numbers into a list.

If the number is counted as many times there are files in the list the number will be added to the suspect numbers set.

**Def map\_numbers\_to\_names(numbers, filename):**

Create a new list where the names of the suspects will be stored

Open the file with the dictionary in read binary mode

    Unpickle the contents

    Loop through the contents

        If number is found as a key the name of the suspect is saved

        Else the suspect is unknown and saved as (Unknown + number)

Returns the names

**Def display\_suspects(names):**

If there are more than 0 names it displays the suspects on the screen

Else prints "no matches"

**Def main():**

Create a new list for the filenames

Create a menu where the user can input their choice and full path. + display\_menu

Try:

    num\_set = cross\_reference(filenames)

    suspects = map\_numbers\_to\_names(num\_set, ownersfile)

    display\_suspects(suspects)

Except FileNotFoundError: An error message

Execute main if file was run directly.