

Course 'Imperative Programming' (IPC031)

Assignment 12: 'Pakjes-avond' unbounded

1. Background

On the basis of *coins*, you solve a search problem which is related to the most important Dutch event of the year: 'pakjes-avond' (gift evening)¹. Instead of a collection of coins you have access to a collection of gifts, and instead of a target value you have a budget and a wish list.

2. Learning objectives

After doing this assignment you are able to:

- implement search problems recursively;
- apply search problem solving techniques for new search problems.

3. Assignment

Every year at 'pakjes-avond' St Nicholas faces the delicate problem of finding matching sets of gifts for the budget of each person. St Nicholas has a huge store of gifts. The name and price of each gift is recorded. You may safely assume that St Nicholas never runs out of gifts. However, you cannot assume that he has every possible gift. Miraculously, St Nicholas knows every person's wish list and budget. A wish list contains names of gifts desired by a person, but without an order of preference.

Part 1: Compute the best gift set

Design and implement the *recursive function* `gifts`, that, for a *wish list* and *budget*, computes a solution containing gifts that are on the wish list and are available in the gift store. Each gift on the wish list is given at most once. In a solution the sum of values of the gifts is *maximal* with respect to the budget (obviously it *may not exceed* it).

Part 2: Getting data

On Brightspace, in "IPC031_2018_assignment_12_files.zip" you may find a number of text files:

- `giftstore.txt`: this file contains information about the gift store. Each line begins with the price of a gift (an integer value, representing the price in cents), followed by the name of the gift (which may contain white-space characters, e.g. "Playstation 4"). This file contains all possible gifts by St Nicholas.
- `Andrew.txt`, `Belle.txt`, `Chris.txt`, `Desiree.txt`, `Edward.txt`, `Fabienne.txt`: these files are wish lists. In a wish list file, the first line is a budget (an integer value, representing the budget in cents). All remaining lines are gift names (a single gift name per line).

Design and implement functions for reading these files into data structures introduced in **part 1**.

Part 3: Testing

Use the functions in **part 2** and test your implementation of `gifts` of **part 1**. Include the results in `main.cpp` by means of comments.

4. Products

As product-to-deliver you only need to upload to Brightspace "`main.cpp`" that you have created with solutions for each part of the assignment.

⇒ **Deadline:** Thursday, December 6, 2018, 13:30h.

¹ For more information about this event, read <https://en.wikipedia.org/wiki/Sinterklaas>.