Het Theebedrijf

Onderdeel 1: De loods

Stelt u zich een bedrijf voor dat thee importeert. De thee moet in uw loodsen worden opgeslagen totdat de distributeur hem komt afhalen. De thee wordt opgeslagen in containers met labels waarop de naam van de theesoort en de datum van binnenkomst is genoteerd. De thee die het eerst in de loods wordt opgeslagen, komt er weer het laatst uit te voorschijn. U wordt gevraagd de loods te modelleren. Men kan slechts van één kant bij de opgeslagen goederen komen; daarom lijkt het handig om als datastructuur een stack of queue te gebruiken. Er zijn twee functies van de loods: het opslaan van nieuwe zendingen, en het leveren aan de distributeur. Uw programma accepteert opdrachten van de gebruiker die corresponderen met die twee functies. De opdrachten worden gegeven via de command line. Een voorbeeldrun van het programma zou er bijvoorbeeld als volgt uit kunnen zien (wat de gebruiker heeft ingetikt staat op de regels die beginnen met een >):

```
./thee1
Welkom bij ons theebedrijf. Geef opdrachten:
> opslaan 200 kg Darjeeling 05-06-2005
200 kg Darjeeling 05-06-2005 wordt opgeslagen.
> opslaan 300 kg Earl Grey 06-09-2005
300 kg Earl Grey 06-09-2005 wordt opgeslagen.
> leveren
300 kg Earl Grey 06-09-2005 wordt geleverd.
> opslaan 800 kg Earl Grey 13-09-2005
800 kg Earl Grey 13-09-2005 wordt opgeslagen.
> leveren
800 kg Earl Grey 13-09-2005 wordt geleverd.
> leveren
200 kg Darjeeling 05-06-2005 wordt geleverd.
> leveren
Er is geen thee meer in voorraad.
> stop
Tot ziens.
```

Opdracht 1

Schrijf een programma `thee1.c` dat, zoals het bovenstaande voorbeeld, middels 'opslaan'- en 'leveren'-opdrachten de loods modelleert. Het programma bestaat naast het onderdeel dat de interactie met de gebruiker behandelt uit de onderdelen `loods1.h` en `loods1.c` die de loods als een stack of een queue modelleren. Ze bevatten de gekozen data structuur met de volgende functies, waarvan ook gebruik wordt gemaakt in `thee1.c`:

`loods1 *maak_loods(void)`

Een functie die de loods-datastructuur aanmaakt. Retourneert NULL wanneer dit niet lukt.

`int leeg(loods1 *)`

Een functie die `1` oplevert als er geen thee meer in het pakhuis aanwezig is en anders `0`.

`void opslaan(loods1 *, char *)`

Een functie die een container met het gegeven etiket in het pakhuis opslaat.

`char *leveren(loods1 *)`

Een functie die de meest recent opgeslagen container uit het pakhuis opdiept en het etiket ervan teruggeeft. Deze functie retourneert NULL wanneer het pakhuis leeg is.

`int sloop_loods(loods *)`

Een functie die het pakhuis helemaal leeg haalt en vervolgens het pakhuis vernietigt. Deze functie dient door `thee1.c` aangeroepen te worden wanneer de gebruiker het commando 'stop' geeft.

Let erop dat alle output naar het scherm in `thee1.c` wordt gedaan. Anders gezegd: er bevindt zich geen `printf()`-aanroep in `loods1.c`!

Onderdeel 2: Lekkere thee

De distributeur is tamelijk boos op zijn leverancier. De kwaliteit van de thee werd door de klanten namelijk niet hoog aangeslagen. Vermoedelijk kan dit verklaard worden uit het feit dat de thee die het eerst binnenkwam, pas als laatste met de distributeur is meegegeven. Die thee heeft dus mogelijkerwijs erg lang in het pakhuis liggen schimmelen. Het is duidelijk dat een oplossing moet worden gevonden voor dit probleem. De meest voor de hand liggende manier zou natuurlijk zijn om aan de achterzijde van het magazijn een uitgang te maken, vanwaar de oudste goederen zouden kunnen worden weggevoerd.

Helaas grenst de loods aan de achterzijde aan een natuurgebied. Uiteindelijk wordt besloten om een tweede loods naast de eerste op te trekken, met precies dezelfde eigenschappen. Met deze twee loodsen bestaat er een manier om, zonder àl te veel heen en weer gesjouw met de thee, de oudste thee het eerste aan de distributeur af te geven.

Opdracht 2.1

U past het programma uit de vorige opdracht aan, zodanig dat het nu ook opdrachten kent om pakketten te verplaatsen van loods 1 naar loods 2 en andersom. Dit nieuwe programma moet `thee2.c` gaan heten. Het maakt gebruik van hetzelfde soort loodsen als eerst. Hieronder volgt een voorbeelduitvoer:

```
./thee2
Welkom bij ons verbouwde theebedrijf. Geef opdrachten:
> opslaan1 200 kg Darjeeling 05-06-2005
200 kg Darjeeling 05-06-2005 wordt opgeslagen in loods 1.
> opslaan1 300 kg Earl Grey 06-09-2005
300 kg Earl Grey 06-09-2005 wordt opgeslagen in loods 1.
> verplaatsen12
300 kg Earl Grey 06-09-2005 wordt verplaatst van loods 1 naar loods 2.
> leveren1
200 kg Darjeeling 05-06-2005 wordt geleverd uit loods 1.
> opslaan1 800 kg Earl Grey 03-04-2006
800 kg Earl Grey 03-04-2006 wordt opgeslagen in loods 1.
> leveren2
300 kg Earl Grey 06-09-2005 wordt geleverd uit loods 2.
> leveren1
800 kg Earl Grey 03-04-2006 wordt geleverd uit loods 1.
> stop
Tot ziens.
```

Nu u hebt uitgedacht hoe twee loodsen gebruikt kunnen worden om de thee zodanig op te slaan dat de klanten niet vergiftigd worden, kunnen we het loodsensysteem iets abstracter bekijken. Het pakhuizenpaar dient namelijk nog steeds hetzelfde doel als aan het begin: thee moet er in worden opgeslagen, en er weer uit worden gehaald, alleen de manier waarop is iets anders geworden. Het zou dus aardig zijn als er een programma geschreven kon worden dat op dezelfde manier kan worden gebruikt als het onderdeel loods1, maar dat intern beschikt over twéé zulke ouderwetse loodsen en een interne organisatie die het onvermijdelijke heen en weer verplaatsen van containers automatisch voor zijn rekening neemt.

Opdracht 2.2

U schrijft nu code voor een nieuw pakhuissysteem, loods3, dat gebruik maakt van twee loods1-loodsen. loods3 bevat precies dezelfde functies als loods1, maar ze zijn anders geïmplementeerd.

Opdracht 2.3

U maakt een programma `thee3.c` die hetzelfde doet als `thee1.c`, maar dan gebruik makend van het nieuwe loodsensysteem. Als het goed is worden `thee3.c` en `thee1.c` vrijwel identiek! In tegenstelling tot `thee1.c` ligt het voor de hand om de `printf()`'s hier wel in `loods3.c` te plaatsen aangezien ze de interne werking van loods3 beschrijven. Hieronder volgt een voorbeelduitvoer:

```
./thee3
Welkom bij ons moderne theebedrijf. Geef opdrachten:
> opslaan 200 kg Darjeeling 05-06-2005
200 kg Darjeeling 05-06-2005 wordt opgeslagen in loods 1.
> opslaan 300 kg Earl Grey 06-09-2005
300 kg Earl Grey 06-09-2005 wordt opgeslagen in loods 1.
> leveren
300 kg Earl Grey 06-09-2005 wordt verplaatst van loods 1 naar loods 2.
200 kg Darjeeling 05-06-2005 wordt verplaatst van loods 1 naar loods 2.
200 kg Darjeeling 05-06-2005 wordt geleverd uit loods 2.
> opslaan 800 kg Earl Grey 03-04-2006
800 kg Earl Grey 03-04-2006 wordt opgeslagen in loods 1.
> leveren
300 kg Earl Grey 06-09-2005 wordt geleverd uit loods 2.
> leveren
800 kg Earl Grey 03-04-2006 wordt verplaatst van loods 1 naar loods 2.
800 kg Earl Grey 03-04-2006 wordt geleverd uit loods 2.
> leveren
Er is geen thee meer in voorraad.
> stop
Tot ziens.
```