

Kääntäminen

Ohjelma kääntyy "lipuilla -Wall -std=c99" ilman varoituksia

Lähdekoodin katselmointi

- Kaikki nimet vaikuttivat olevan englanniksi, ja kommentit englanniksi ja suomeksi. Mieluiten kaikki jommallakummalla kielellä?
- Ohjelmassa vaikuttaisi olevan aika paljon turhia include:ja, esim. program.c ei taida käyttää time.h:ta missään?
- Miksi joissakin funktioissa on sisääntuleva argumentti "uudelleennimetty" heti toisella rivillä? Esim.

```
bottleLink getBottleNode(bottleLink b, int n) {  
    bottleLink temp = b;
```

- Jotkin yksikirjaimiset muuttujien nimet eivät ole kovin selkeitä
- Kommentteja on mukavasti ja lähes kaikki ovat c kielen kommenttisyntaksilla.
- Lähdekoodi on yleisesti selkeä ja ymmärrettävä. Mainin voisi halutessaan jakaa useampaan funktioon.

Toteutusten eroavaisuus

Ohjelmien suurin eroavaisuus on listoissa. Testattavassa ohjelmassa on kaksi instanssia samasta listasta, meidän tekemässä on kaksi erillistä listaa. Meidän tulkinnan mukaan tehtävänannossa lukee, että palautettujen pullojen listassa on vain "id", jonka avulla tuotetietoja haetaan pullotietojen listasta.

Jos ymmärsin oikein, niin testattavassa ohjelmassa pulloja palauttaessa:

1. Lisätään bottleTypes listaan oikean pullon kohdalle määrään +1
2. Lisätään oikea pullo bottleReturned listaan
3. Palautuksen päätyttyä lasketaan bottleReturned listassa olevien alkioiden määrä ja käytetään tätä vain palautusten yhteismäärän tulostamiseen
4. Varsinainen pantinlaskenta tapahtuu bottleTypes listassa olevien määrien avulla

Syötteiden tarkastus

Ohjelma tarkastaa syötteet hyvin, eikä kaadu virheellisistä syötteistä. Ainoana ongelmana on, että ylipitkät syötteet jäävät puskuriiin. Kun palautusvalikossa valinnaksi syötetään esimerkiksi pitkä merkkijono, jossa on 99 kertaa 0 ja lopussa kerran 1, niin ensin ohjelma antaa virheilmoituksen väärästä valinnasta. Sen jälkeen 1 luetaan puskurista ja ensimmäinen pullo palautetaan. Jos merkkijono on monta sataa merkkiä pitkä, niin ohjelma toistaa automaattisesti valikkoa, kunnes puskurin on tyhjä.

Yli 100 merkin syötteen tullessa käyttäjältä puskurin voisi siis tyhjentää, jotta saataisiin vain yksi virheilmoitus.

Jos ohjelman pantit tai niiden summat ovat erittäin suuria, ohjelma tulostaa inf€. Tosin niin suuret palautusmäärät eivät ole käytännön tilanteessa realistisia.

Tuotetiedoston muokkaus

Tuotetiedoston muokkaaminen (esimerkiksi pantin poistaminen joltain tuoteriviltä, niin, että vain nimike ja koko jäävät) kaataa ohjelman.

Valgrind

Valgrind ei löytänyt muistivuotoja. Virheitä se löysi 374, mutta niihin ei tässä testauksessa oteta kantaa.

Valgrindin tuloste alla:

```
==23441== HEAP SUMMARY:
==23441==    in use at exit: 0 bytes in 0 blocks
==23441==   total heap usage: 31 allocs, 31 frees, 6,213 bytes allocated
==23441==
==23441== All heap blocks were freed -- no leaks are possible
==23441==
==23441== For counts of detected and suppressed errors, rerun with: -v
==23441== Use --track-origins=yes to see where uninitialised values come from
==23441== ERROR SUMMARY: 374 errors from 101 contexts (suppressed: 0 from
0)
```

Kehitysehdotukset

Harjoitustyö vaikutti pääosin hyvältä. Kommentit saisivat olla kaikki samalla kielellä. Työssä voisi olla tehtävänannossa mainittu palautuslista erikseen, vaikka toimii varsin hyvin näinkin. Tuotetiedoston lukuvirheestä voisi tulla käyttäjälle virheilmoitus ohjelman kaatumisen sijaan. Aiemmin mainittu puskurin tyhjennyskin voisi olla paikallaan, vaikka niin pitkiä syötteitä harvemmin käyttäjältä tulee.