

Wstęp do programowania, potok imperatywny (Info, I rok) 16/17, laboratorium

Kokpit ► Moje kursy ► WPI.LAB.INFO.I.16/17 ► Zadanie 3 ► Zadanie 3: Liliczby

Zadanie 3: Liliczby

Wprowadzenie

Liliczba (ang. *nunumber*) to rekurencyjna reprezentacja nieujemnej liczby całkowitej za pomocą ciągu cyfr, które również są liliczbami. Reprezentacja ta, sformułowana w nieco inny ale równoważny sposób, była rozważana przez Donalda Knutha.

Wartością liliczy jest suma potęg dwójki o wykładnikach będących wartościami cyfr tej liliczy. Powiemy, że liliczba jest znormalizowana, jeśli jej cyfry są znormalizowanymi liliczbami i są uporządkowane rosnąco według wartości.

Tekstowym zapisem liliczy jest słowo języka z poniższą gramatyką w rozszerzonej notacji BNF:

```
<liliczba> ::= { "Y" <cyfra> } "Z"  
<cyfra> ::= <liliczba>
```

Można zauważyć, że jest to język słów powstałych przez dopisanie symbolu **Z** na koniec wyrażenia nawiasowego, w którym **Y** pełni rolę nawiasu otwierającego a **Z** to nawias zamykający.

Wszystkie liliczy, których zapisem są poniższe słowa

```
YZYZYZYZYZYZYZYZYZZ  
YZYZYZYZYZYZYZYZZZ  
YZYZYZYYYZZYZYZZZZ  
YYYZZYZYZYZYZYZZZZ  
YYYZZYZYZYZYZYZZZZ  
YYYZZYZYZYZYZYZZZZ  
YYYZZYZYZYZYZYZZZZ  
YYYZZYZYZYZYZYZZZZ  
YYZZYYYZZYZYZZZZ  
YYZZYYYZZYZYZZZZ  
YYZZYYYZZYZYZZZZ
```

mają wartość 10. Ostatnia z nich jest znormalizowana.

Inne przykłady znormalizowanych liliczb są w poniższej tabelce:

Wartość	Liliczba
0	Z
1	YZZ
2	YYZZZ

Wartość	Liliczba
3	YZYYZZZ
4	YYYZZZZ
5	YZYYYYZZZ
6	YYZZYYYYZZZ
7	YZYZZZYYZZZ
8	YYZYZZZZ
9	YZYZZYYZZZ
10	YYZZYYZYZZZ
11	YZYZZZYZYZZZ
12	YYYZZZYZYZZZ
13	YZYYYYZZZYZYZZZ
14	YYZZYYYYZZZYZYZZZ
15	YZYZZYYYYZZZYZYZZZ
16	YYYYZZZZZ
17	YZYYYYZZZZZ
18	YYZZYYYYZZZZZ
19	YZYZZYYYYZZZZZ
20	YYYZZYYYYZZZZZ
50	YYZZYYYYZZZZYZZYYYYZZZZZ
64	YYYZZYYYZZZZZ
100	YYYZZZYZYZZZZZYZZYYYYZZZZZ
127	YZYZZZYZZZZZYZZZZYYYYZZZZYZZYZZZZYYYZZYYYYZZZZZ
128	YYZYZZZYZZZZZ
144	YYYYZZZZYZZYZZYYYYZZZZZ
199	YZYZZZYZZZZYZZYZZYZZZZYZZYZZZYZZZZZ
256	YYYZYZZZZZ
1000	YYZYZZZYZZYZZZZYZZYZZYZZZZYZZYZZZYZZZZYZZYZZZZZYZZYZZZZZZZ
1009	YZYYYYZZZZYZZYZZZZZZYZZYZZYZZZZYZZYZZZYZZZZYYYZZZZZYZZYZZYZZZZZ
1024	YYYZZYZZYZZZZZ
10000	YYYYZZZZYZZYZZZZZYZZYZZYZZZZYZZYZZZZZYZZYZZZYZZZYZZYZZZZZ
65536	YYYYYZZZZZZ

- Program, który zarezerwował pamięć funkcjami `malloc`, `realloc` itp. ma obowiązek zwolnić ją funkcją `free`. Jeśli tego nie robi, występuje zjawisko wycieku pamięci, które uznajemy za błąd. W wykryciu tego i innych błędów może pomóc program `valgrind`. By z niego skorzystać, kompilujemy swój program z dodatkową opcją `-g`, np. poleceniem:

```
gcc -std=c89 -pedantic -Wall -Wextra -Werror -g liliczyby.c -o liliczyby
```

Spowoduje to dołączenie do programu wykonywalnego informacji pomagających w lokalizacji błędów. Tak skompilowany program uruchamiamy pod kontrolą `valgrind` poleceniem:

```
valgrind --leak-check=full ./liliczyby
```

Opcja `--leak-check=full` wskazuje, że chcemy, między innymi, wykryć wycieki pamięci i znaleźć ich źródło.

Na zakończenie wykonania programu przez `valgrind`, na wyjście diagnostyczne wypisywany jest raport. Jeżeli nie wykryto błędów, może on mieć postać np.

```
==46974== Memcheck, a memory error detector
==46974== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==46974== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==46974== Command: ./liliczyby
==46974==
==46974==
==46974== HEAP SUMMARY:
==46974==      in use at exit: 0 bytes in 0 blocks
==46974==    total heap usage: 97 allocs, 97 frees, 9,712 bytes allocated
==46974==
==46974== All heap blocks were freed -- no leaks are possible
==46974==
==46974== For counts of detected and suppressed errors, rerun with: -v
==46974== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Wartość inna niż 0 po `ERROR SUMMARY` informuje, że wykryto błąd. W raporcie będą też wskazówki pomagające w lokalizacji błędów.

- Rozwiązania do testów będą kompilowane poleceniem:

```
gcc -std=c89 -pedantic -Wall -Wextra -Werror -g nazwa.c -o nazwa
```

Wszystkie wymienione opcje kompilatora są obowiązkowe i nie wolno dodawać do nich żadnych innych.

- Podczas testów rozwiązania będą uruchamiane pod kontrolą programu `valgrind`. Jeżeli wykryje on błąd, np. wyciek pamięci, to przyjmujemy, że program testu nie przeszedł nawet, jeśli jego wynik będzie prawidłowy.
- Tekstowy zapis liliczyby można przekształcić na wyrażenie arytmetyczne, zastępując kończące `z` przez `-0`, każdą parę znaków `yz` przez `y0z`, parę `zy` przez `z+y` a następnie każdy znak `y` przez `2^(` i każdy znak `z` przez `)`.

Za pomocą programu `sed` zastępującego wzorce w tekście oraz kalkulatora wyrażeń arytmetycznych `bc` możemy poznać dziesiętny zapis wartości liliczyby. Np. polecenie:

```
echo YZZZZZZZZZZZZZZZZZZZZ | sed 's:Z$:-0:g; s:YZ:Y0Z:g; s:ZY:Z+Y:g; s:Y:2^( :g; s:Z:):g' | bc
```

wypisze:

W rozwiązaniu zadania nie wolno korzystać z programów `sed` i `bc`, ale mogą one pomóc w testowaniu.

- Rozwiązanie zadania wymaga zastosowania rekursji i dynamicznych struktur danych.
- Iloczyn dwóch liczb można wyznaczyć, korzystając ze wzoru:

$$\left(\sum_{i=0}^m 2^{a_i} \right) \left(\sum_{j=0}^n 2^{b_j} \right) = \sum_{i=0}^m \sum_{j=0}^n 2^{a_i+b_j}$$

Będzie do tego potrzebne dodawanie.


Obliczenie sumy znormalizowanych liczb wymaga scalenia uporządkowanych rosnąco list cyfr. W przypadku, gdy ta sama cyfra występuje w obu dodawanych liczbach, realizujemy przeniesienie.

By scalić listy cyfr potrzebujemy porównania. Cyfry liczb porównujemy w kolejności od najbardziej znaczących.

Pracę nad rozwiązaniem proponujemy zacząć od programu, który czyta i wypisuje liczbę. Następnie zmieniamy go, kolejno, w program czytający dwie liczby i wypisujący wynik ich porównania, ich sumę i na koniec ich iloczyn.

przykład1.in
przykład1.out
przykład2.in
przykład2.out
przykład3.in
przykład3.out

Status przesłanego zadania

Numer próby	To jest próba nr 1.
Status przesłanego zadania	Przesłane do oceny
Stan oceniania	Ocenione
Termin oddania	środa, 11 styczeń 2017, 10:00
Pozostały czas	Zadanie zostało złożone 1 dzień 12 godz. przed terminem
Ostatnio modyfikowane	poniedziałek, 9 styczeń 2017, 21:29
Przesyłane pliki	 kk385830-liliczyby.c
Komentarz do przesłanego zadania	► Komentarze (0)

Informacja zwrotna

Ocena 9,00 / 10,00

Ocenione dnia środa, 11 styczeń 2017, 14:33

Ocenione przez



Eryk Kopczyński

NAWIGACJA



Kokpit

■ Strona główna

Strony

Moje kursy

IPP.INFO.I.16/17

POWI.INFO.I.16/17


WPI.INFO.I.16/17

WPI.LAB.INFO.I.16/17

Uczestnicy

 Odznaki

 Kompetencje

 Oceny

Główne składowe


Zadanie 0 (treningowe)

Zadanie 1

Zadanie 2

Zadanie 3

 **Zadanie 3: Liliczby**

 Zadanie 3: poprawność

Zadanie 4 (poprawkowe)

PO.INFO.I.16/17

MD.INFO.I.16/17

ADMINISTRACJA



Administracja kursem

Jesteś zalogowany(a) jako Krzysztof Kowalczyk (Wyloguj)

WPI.LAB.INFO.I.16/17

Moodle, wersja 3.2.2+ | moodle@mimuw.edu.pl