

Zawsze chciałem napisać kompilator w wakacje!
Krzysztof Bartoszek

Napisz kompilator, czyli program, który na wejściu wczyta wyrażenie arytmetyczne, a na wyjściu wypisze obliczający je program w języku BF.

Język BF

Programy w języku BF wykonywane są przez maszynę, która operuje na prawostronnie nieskończonej taśmie bajtów, tzn. 8-bitowych liczb bez znaku, które mogą przyjmować wartości od 0 do 255 włącznie.

Początkowo taśma zawiera same zera, a głowica maszyny znajduje się nad pierwszą (skrajnie lewą) komórką taśmy. Program w języku BF składa się z ośmiu rodzajów operacji, z których każda oznaczana jest jednym znakiem.

- Operacja `+` zwiększa wartość w komórce o 1, a operacja `-` zmniejsza ją o 1.
- Operacja `<` przesuwa głowicę o jedną komórkę w lewo, a operacja `>` przesuwa głowicę o jedną komórkę w prawo. Wysunięcie głowicy poza taśmę skutkuje niezdefiniowanym zachowaniem.
- Operacja `,` wczytuje liczbę do aktualnej komórki ze standardowego wejścia.
- Operacja `.` wypisuje liczbę z aktualnej komórki na standardowe wyjście.
- Operacja `[` skacze za odpowiadający jej znak `]`, jeżeli w aktualnej komórce jest zero, a w przeciwnym przypadku nie robi nic.
- Operacja `]` zawsze skacze do odpowiadającego jej `[`. Dzięki temu, pary nawiasów `[i]` działają jak pętla `while`.

Więcej o języku BF możecie poczytać na Wikipedii (<https://pl.wikipedia.org/wiki/Brainfuck>). Różnice między używanym przez nas wariantem BF a oryginałem są następujące:

- W oryginale taśma ma stałą ograniczoną długość, a u nas jest (prawostronnie) nieskończona.
- W oryginale operacje `,` i `.` wczytują znaki ASCII, a u nas liczby. Przez to nie da się napisać programu wypisującego *Hello world*, ale za to da się łatwiej robić sensowną arytmetykę.

Język wyrażeń arytmetycznych

Wyrażenia arytmetyczne mogą składać się ze stałych (liczb z przedziału $[0, 255]$), nazw zmiennych (ciągów małych liter alfabetu łacińskiego), operatorów (dodawania, odejmowania i mnożenia oznaczanych odpowiednio znakami `+`, `-` i `*`) oraz nawiasów. Kolejność wykonywania działań jest taka jak zwykle. Wszystkie operacje na liczbach wykonujemy modulo 256. Aby ułatwić Wam parsowanie wyrażeń, wszystkie tokeny (tj. stałe, zmienne, operatory i nawiasy) są oddzielone pojedynczymi spacjami.

Wejście i wyjście

W pierwszym i jedynym wierszu wejścia znajduje się jedno wyrażenie arytmetyczne. Na wyjściu należy wypisać program w języku BF, który wczytuje wartość zmiennych obecnych w wyrażeniu, oblicza wartość wyrażenia i wypisuje ją na standardowe wyjście.

Wyprodukowany przez kompilator program powinien wczytać wartości zmiennych w kolejności alfabetycznej. Przykładowo, dla wyrażenia $(zmienna * 2) + a * b + 7$, program w BF powinien wczytać najpierw wartość `a`, potem wartość `b`, a potem wartość `zmienna`.

Przykłady

Wejście	Przykładowe wyjście
45 - 42	+++.

Wejście	Przykładowe wyjście
x	,.

Wejście	Przykładowe wyjście
x + 3	,+++.

Wejście	Przykładowe wyjście
x + y + 3	,>,[-<+>]<+++.

Wejście	Przykładowe wyjście
x + 254	,--.

Więcej przykładów wyrażeń (ale już bez przykładowych wyjść) znajdziecie w zakładce *Pliki*.

Ocenianie

Testy

Przykłady w zakładce *Pliki* są reprezentatywne, tzn. nie mamy testów bardziej złośliwych albo istotnie innych niż te przykłady. Wydajność wygenerowanych programów będziemy oceniać na losowych (ale takich samych dla wszystkich zgłoszeń) wartościach zmiennych. Innymi słowy, nie próbowaliśmy celowo generować przypadków w których potrzebne są specyficzne optymalizacje kompilatora.

Punkty za poprawność

Skompilowane programy, które będą zwracać niepoprawne odpowiedzi lub wykonają więcej niż 10^7 instrukcji, otrzymają 0 punktów. Programy, które zwrócą poprawną odpowiedź, otrzymają 40 punktów. Więcej punktów można dostać za wydajność.

Punkty za wydajność

Programy mogą otrzymać dodatkowe punkty za wydajność, zależnie od tego, jak wypadają na tle programów wygenerowanych przez nasze kompilatory. Przygotowaliśmy dwa kompilatory: kiepski oraz dobry.

- Programy wykonujące tyle operacji co wygenerowane przez kompilator kiepski (lub wolniej) otrzymają 0 punktów za wydajność.
- Programy wykonujące tyle operacji co wygenerowane przez kompilator dobry otrzymają 30 punktów za wydajność.
- Programy wykonujące 0 operacji otrzymają 60 punktów za wydajność.

Pomiędzy tymi trzema punktami będziemy interpolować liniowo.

Maszyna BF

W zakładce *Pliki* znajduje się implementacja maszyny wykonującej programy w języku BF. Można ją skompilować na przykład poleceniem: `g++ bfvm.cc -obfvm` Aby uruchomić program w BF, należy uruchomić `bfvm` podając jako argument ścieżkę do pliku z programem.