

Zadanie programistyczne 2

Zadanie składa się z dwóch części: A i B. W wariancie **uproszczonym (max 75 punktów)** należy rozwiązać część A; w wariancie **pełnym** - obie części A i B.

Uwaga: wykonanie części B jest istotnie prostsze, jeśli ma się działającą część A.

Część A (max 75 punktów)

Napisz program symulujący wykonanie podanej maszyny Turinga (z możliwymi znakami "0", "1", "#") dla podanych początkowych zawartości taśmy.

PRZYKŁAD NA ROZGRZEWKĘ

Dla wejścia

20

/ To jest komentarz. W powyższej linii jest liczba wszystkich linii wejścia - dla Państwa wygody.

// To też jest komentarz (tym razem zaczyna się od dwóch ukośników; co nam szkodzi).

// W poniższych liniach są wszystkie reguły przejść dla pewnej maszyny Turinga.

// Maszyna ta patrzy na znak na pozycji startowej, i jeśli jest on zerem, kopiuje go "o 1 w lewo".

```
START # -> STOP # right
START 0 -> KOPIUJ 0 left
START 1 -> STOP 1 right
```

```
// W stanie KOPIUJ _musimy_ zobaczyć znak # (patrz opis założeń w części "WEJŚCIE")
KOPIUJ # -> STOP 0 left
```

// Koniec opisu maszyny. Przechodzimy do opisu przykładowych taśm wejściowych.

// Tu maszyna wykona kopiowanie

\$ 011##

// A tu nie zmienia nic

\$ ###1#00#11

poprawnym wyjściem będzie

0011

1#00#11

WEJŚCIE

Program powinien czytywać dane wejściowe na jeden z dwóch sposobów (wybór pozostawiam Państwu):

- bezpośrednio ze standardowego wejścia
- lub z pliku, którego nazwa jest jedynym argumentem uruchomienia programu

Pierwsza linia zawiera jedną liczbą całkowitą, stanowiącą liczbę wszystkich linii wejścia.

Każda następna linia jest jednej z następujących postaci:

- linia pusta - należy ją zignorować (pozwalamy na nią, aby pliki wejściowe były czytelniejsze)
- linia rozpoczynająca się od znaku "/" - należy ją również zignorować (z przyczyn jak wyżej; takie linie mają funkcję komentarzy)
- linia rozpoczynająca się od znaku "\$" oraz spacji, po których następuje opis jednej przykładowej zawartości taśmy na początku pracy maszyny; początkowo głowica maszyny wskazuje na **pierwszy** z podanych znaków; wszystkie pozostałe znaki na taśmie (zarówno po lewej, jak po prawej od opisanego fragmentu) mają wartość "#";
- wszystkie pozostałe linie wejścia opisują reguły przejścia maszyny Turinga i mają postać

$S_b c_b \rightarrow S_e c_e M$

gdzie symbole mają takie samo znaczenie, jak w slajdzie 2 wykładu 4, a więc:

- S_b , S_e są nazwami stanów maszyny odpowiednio przed i po wykonaniu danego przejścia; nazwy takie składają się z wielkich i małych liter alfabetu łacińskiego, cyfr oraz znaku podkreślenia (); stan początkowy nosi nazwę "START", zaś końcowy "STOP"
- c_b , c_e opisują zawartość danej komórki taśmy odpowiednio przed i po wykonaniu danego przejścia
- M opisuje kierunek ruchu taśmy na koniec wykonania danego przejścia; jest to jeden z napisów "left" albo "right".
- symbol " \rightarrow " właściwie nie pełni żadnej funkcji, ale umieszczamy go w wejściu dla czytelności.

Dodatkowo, wejście spełnia następujące warunki:

- wszystkie linie rozpoczynające się od znaku "#" następują po wszystkich liniach opisujących reguły (jednak linie puste oraz linie z komentarzami mogą się z powyższymi dowolnie przeplatać)
- dane wejściowe zawiera maksymalnie 1000 linii
- każda linia zawiera maksymalnie 100 znaków
- podane wejście nigdy spowoduje przesunięcia głowicy dalej niż o 10000 pozycji od pozycji początkowej (w lewo bądź w prawo)
- maszyna nie otrzymuje sprzecznych poleceń, a dokładnie:
 - dla dowolnej pary (S_b , c_b) wejście zawiera co najwyżej jedną regułę postaci $S_b c_b \rightarrow [...]$
- maszyna otrzymuje wszystkie potrzebne do wykonania polecenia, a dokładnie:
 - jeśli kiedykolwiek w trakcie wykonania maszyna znajdzie się w stanie S mając na głowicy znak c , to wejście zawiera dokładnie jedną regułę postaci $S c \rightarrow [...]$

Uwaga. Dla ułatwienia, nie wymagam jakiegokolwiek zachowania programu, gdyby te założenia były złamane. Proszę po prostu założyć ich prawdziwość, i nie przejmować się innymi przypadkami.

WYJŚCIE

Dla każdej otrzymanej na wejściu przykładowej zawartości taśmy, symulator powinien wypisać jedną linię, opisującą zawartość taśmy po zakończeniu wykonania maszyny.

Opis ten powinien obejmować dokładnie odcinek taśmy pomiędzy pierwszym a ostatnim znakiem różnym od "#" (włącznie).

WIĘCEJ PRZYKŁADÓW

Dwa bardziej rozbudowane opisy maszyn Turinga (odpowiadające tym ze slajdów 4 oraz 6 z wykładu 4), wraz z kilkoma przykładami początkowej zawartości taśmy i oczekiwanym wyjściem, zamieszczam poniżej.
(Tzn. pod opisem części B).

Część B (max 25 punktów)

Wykorzystując język opisu maszyn Turinga z części A, napisz maszynę, która podwaja każdy znak w pierwszym "słowie" napotkanym na prawo od startowej pozycji głowicy. Mówiąc o "słowie", traktujemy znak "#" jako spację - tzn. "słowem" nazywamy dowolny fragment taśmy niezawierający znaku "#", ale otoczony tym znakiem po obu stronach.

Ściślej, wymagamy tylko tego, aby pierwsze słowo (od lewej) na taśmie po zakończeniu działania maszyny stanowiło podwojenie (znak po znaku) pierwszego słowa na taśmie (od lewej) przed jej wykonaniem.

Maszyna ma prawo dowolnie zamazać dowolny obszar taśmy na prawo od wypisanego podwojenia. Ma też prawo wypisać to podwojenie w dowolnym miejscu (byle by w limicie 10000 znaków na lewo/prawo od pozycji startowej).

ZAŁOŻENIA

Podtrzymujemy założenia dotyczące wejścia z części A, to znaczy:

- na lewo od pozycji startowej głowicy występują wyłącznie znaki "#"
- dalej niż 100 pozycji na prawo od pozycji startowej głowicy również występują wyłącznie znaki "#"

WYMAGANIA

Kod maszyny powinien spełniać założenia z części A dotyczące wejścia (linie długości max 100 znaków, nazwy stanów składające się z liter, cyfr i znaku _, itp.). Chodzi o to, żeby symulator maszyny Turinga był w stanie uruchomić kod opracowany w części B.

PRZYKŁAD

Dla początkowej zawartości taśmy opisanej linią

```
$ ###0101#
```

maszyna powinna produkować taśmę, na której pierwszym słowem jest 00110011.

Wynik może zawierać to słowo z pewnym przesunięciem (np. pierwszy znak "0" z tego słowa może znajdować się na pozycji startowej, albo 4 znaki przed nią, albo 15 znaków za nią, itp.)

Wynik może też zawierać więcej słów oprócz podwojonego, byle by były one po nim (np. następujący stan taśmy, zaczynając od pozycji startowej, będzie poprawny: ###00110011##0#1#0# i dalej już same "#").

Natomiast wynik ###00110011**1**##0#1#0# (i dalej same "#") nie jest poprawny, ponieważ pierwsze słowo w nim jest dłuższe od oczekiwanego o dwa dodatkowe znaki "1" (zaznaczone na czerwono).

UWAGI

Jeśli będą Państwo dysponować poprawnym symulatorem dla części A, to za jego pomocą można łatwo testować część B. (W rozważanym przykładzie symulator powinien wypisać 00110011 lub 00110011#*i dalej cokolwiek*)

Gdyby ktoś z Państwa miał pomysł na uprzyjemnienie sobie części B poprzez lekkie rozszerzenie języka obsługiwanego przez swój symulator w części A, to jest szansa, że się zgodzę - proszę o kontakt.

Przykład maszyny Turinga: inkrementacja w systemie unarnym (wykład 4, slajd 4)

```
26
// poczatek dzialania maszyny
START # -> A # right
START 0 -> A 0 right
START 1 -> B 1 right

A # -> A # right
A 0 -> A 0 right
A 1 -> B 1 right

B # -> C 1 right
B 0 -> C 1 right
B 1 -> B 1 right

// zakonczenie dzialania maszyny
C # -> STOP # right
C 0 -> STOP 0 right
C 1 -> STOP 1 right

// PRZYKLADOWE POCZATKOWE ZAWARTOSCI TASMY
// oczekiwany wynik: 00##1111
$ ###00##111#
// oczekiwany wynik: 001110##1
$ ###001100##1#
// oczekiwany wynik: 11111
$ ###11011#
```

Przykład maszyny Turinga: palindromy (wykład 4, slajd 6)

```
46
// poczatek dzialania maszyny
START # -> R # right
START 0 -> RZ # right
START 1 -> R0 # right

R # -> R # right
R 0 -> RZ # right
R 1 -> R0 # right

RZ # -> CZ # left
RZ 0 -> RZ 0 right
RZ 1 -> RZ 1 right

R0 # -> C0 # left
R0 0 -> R0 0 right
R0 1 -> R0 1 right
```

```
// zakonczenie dzialania maszyny, ciag jest palindromem nieparzystej dlugosci
CZ # -> STOP # left
CZ 0 -> TMP # left
// zakonczenie dzialania maszyny, ciag nie jest palindromem
CZ 1 -> STOP 1 left

// zakonczenie dzialania maszyny, ciag jest palindromem nieparzystej dlugosci
CO # -> STOP # left
// zakonczenie dzialania maszyny, ciag nie jest palindromem
CO 0 -> STOP 0 left
CO 1 -> TMP # left

// zakonczenie dzialania maszyny, ciag jest palindromem parzystej dlugosci
TMP # -> STOP # left
TMP 0 -> L 0 left
TMP 1 -> L 1 left

L # -> R # right
L 0 -> L 0 left
L 1 -> L 1 left

// PRZYKLADOWE POCZATKOWE ZAWARTOSCI TASMY
// oczekiwany wynik: 10
$ ###1##10#
// oczekiwany wynik: 10
$ ##0110##10#
// oczekiwany wynik: 11100#0
$ ##111100#0#
```