MALACHOVIACUS
INFORMATICUS

Płock, 7 kwietnia 2017 r.

Instrukcje dla zawodników

Arkusze otwieramy na wyraźne polecenie komisji. Wszystkie poniższe instrukcje zostaną odczytane i wyjaśnione.

1. Arkusz składa się z **3 zadań**.
2. Każde zadanie składa się z **wprowadzenia** oraz kilku **pytań**.
3. Liczba punktów możliwa do uzyskania za każde pytanie podana jest przy jego treści. Suma tych punktów w każdym zadaniu wynosi 20.
4. Przed udzieleniem odpowiedzi na pytania **przeczytaj dokładnie** wprowadzenie oraz treści poleceń.
5. Po przeczytaniu treści zadania, spróbuj rozwiązać zadanie **na brudno**, na przeznaczonych do tego pustych arkuszach. Kiedy uznasz, że twoja odpowiedź jest poprawna, przepisz ją **czytelnie** na arkusz odpowiedzi. Nieczytelne odpowiedzi oraz arkusze zapełnionione skreśleniami **nie będą oceniane**.
6. Do zapisu odpowiedzi używaj wyłącznie **długopisu lub pióra** z **czarnym lub niebieskim** tuszem. Do wykonywania rysunków możesz użyć ołówka.
7. Każdy arkusz odpowiedzi powinien zawierać odpowiedź, lub jej część, na **tylko jedno** zadanie.
8. Na pierwszej stronie każdego arkusza odpowiedzi, w prawym górnym rogu, zapisz czytelnie **swój kod** oraz **numer zadania**.
9. Czas na rozwiązanie zadań wynosi **120 minut**.

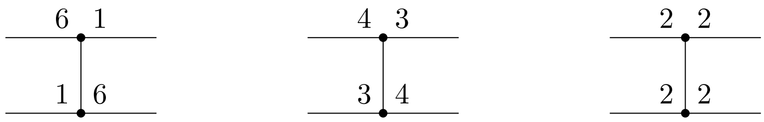
Powodzenia!

Zadanie 1

Sieci porównujące

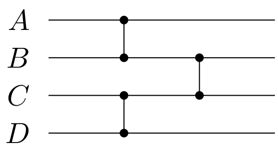
Sieć porównująca dla n liczb składa się z n poziomych **przewodów**. W lewym końcu każdego z przewodów umieszczamy jedną liczbę naturalną. Liczba ta przemieszcza się po swoim przewodzie w prawo aż dotrze do prawego końca.

Parę przewodów możemy połączyć pionowym **porównywaczem**, który porównuje liczby wchodzące do niego oboma przewodami, po czym zamienia je miejscami tak, aby w górym przewodzie znajdowała się mniejsza z dwóch liczb.



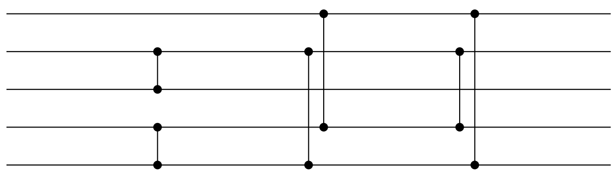
Rysunek 1: Przykład zachowania porównywaczy. Jeżeli górny przewód zawiera mniejszą z dwóch liczb (lub obie liczby są równe), nic się nie dzieje. Jeżeli górnym przewodem porusza się większa z liczb, obie liczby zamieniane są miejscami.

Kolejność działania porównywaczy ma znaczenie: jeżeli dwa porównywacze dotykają tego samego przewodu, to pierwszy wykonuje swoje działanie porównywacz po lewej. Jeżeli jednak dwa z nich łączą zupełnie inne pary przewodów, to mogą one działać jednocześnie. Rysujemy je wtedy na rysunku bliżej siebie.

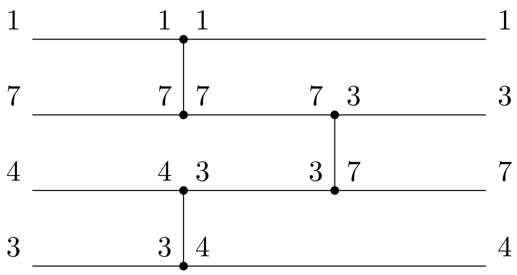


Rysunek 2: Porównywacze AB i CD nie wspódzielą żadnych przewodów, mogą więc wykonywać swoje działania jednocześnie. Porównywacz BC natomiast dzieli przewód B z AB oraz C z CD, stąd musi zaczekać na zakończenie działania obu z nich.

Każdy z porównywaczy potrzebuje 1 jednostki czasu, aby wykonać swoje działanie. Czas potrzebny na przemieszczanie liczb wzdłuż przewodów jest tak mały, że będziemy pomijać go w obliczeniach.



Rysunek 3: Powyższa sieć składa się z 6 porównywaczy, jednak potrzebuje ona jedynie 3 jednostek czasu na wykonanie wszystkich porównań – w każdej z 3 par porównywaczy, oba mogą działać jednocześnie.



Rysunek 4: Przykład działania sieci porównującej.

PYTANIA

PYTANIE 1 Spójrz na rysunek 4, który pokazuje przykład działania sieci porównującej. Jak wyglądałoby działanie tej sieci, gdyby zamiast liczb 1, 7, 4 i 3, umieścić w przewodach (kolejno od góry) liczby 8, 5, 7 i 2? Narysuj sieć i opisz odpowiednimi wartościami końce przewodów i porównywacze, jak na rysunku powyżej. [3 punkty]

PYTANIE 2 Narysuj sieć porównującą zawierającą 4 przewody, która dla każdego zestawu 4 liczb ułożonych w dowolnej kolejności wybierze i umieści w górnym przewodzie liczbę o najmniejszej wartości. Kolejność pozostałych liczb nie ma znaczenia. [4 punkty]

PYTANIE 3 Na podstawie odpowiedzi na poprzednie pytanie, narysuj sieć porównującą zawierającą 3 przewody, która każdy zestaw trzech liczb ułoży w kolejności od najmniejszej (w górnym przewodzie) do największej (w dolnym). [5 punktów]

PYTANIE 4 Narysuj taką sieć porównującą zawierającą 4 przewody, która każdy zestaw 4 liczb ułoży w kolejności od najmniejszej (w górnym przewodzie) do największej (w dolnym) w jak najkrótszym czasie. Ile czasu będzie potrzeba twojej sieci? [8 punktów]

Uwaga! Za ostatnie pytanie otrzymasz punkty nawet jeżeli twoje rozwiązanie nie będzie idealne. Jednak im bliżej prawdziwej najmniejszej możliwej ilości czasu, tym więcej punktów dostaniesz.

Zadanie 2

Programy dla procesora

W tym zadaniu będziesz pisać programy dla procesora podobnego do tego opisanego w materiałach przygotowawczych. Dla przypomnienia:

- pamięć komputera składa się z dowolnej liczby komórek, numerowanych kolejno od 0, z których każda jest w stanie przechować jedną liczbę całkowitą, dowolnie dużą lub małą, także ujemną,
- początkowa wartość komórki 0 to zawsze 0, początkowe wartości pozostałych komórek możemy ustalić dowolnie,
- instrukcje programu ponumerowane są kolejnymi liczbami naturalnymi od 0,
- program kończy działanie, jeżeli kolejna instrukcja do wykonania nie istnieje (t.j. była ostatnią instrukcją programu lub wykonany został skok do nieistniejącej instrukcji).
- wynikiem działania programu jest **zawsze** wartość w komórce 0 po zakończeniu działania programu.

Procesor w tym zadaniu obsługuje 3 rodzaje operacji arytmetycznych:

- `add a, b, c`, dodawanie,
- `sub a, b, c`, odejmowanie,
- `mod a, b, c`, reszta z dzielenia.

Dla każdej z powyższych operacji, b i c to wartości na których wykonujemy operacje (liczby, jak `#3`, lub adresy komórek, jak `@2`), natomiast a jest adresem komórki, w której umieszczony zostanie wynik operacji.

Wspierane są również następujące operacje skoku:

- `br l`, bezwarunkowy skok do instrukcji l,
- `brz a, l`, jeżeli wartość w komórce a jest równa 0,
- `brg a, l`, jeżeli wartość w komórce a jest większa lub równa 0.

Zwróć uwagę, że niektóre programy mogą dla pewnych początkowych wartości komórek pamięci nigdy nie zakończyć swojego działania. Na przykład program:

```
0: brg 1, 0
```

zakończy swoje działanie wtedy i tylko wtedy, gdy początkowa wartość komórki 1 będzie mniejsza od 0. W przeciwnym razie program nigdy się nie zakończy.

Poniższe pytania zawierają tabele z przykładami. Kolumny oznaczone poprzez `@1` oraz `@2` zawierają początkowe wartości w komórkach 1 i 2, pozostałe komórki mają wartość 0. Kolumna Wynik zawiera oczekiwany wynik działania programu lub słowo *brak*, jeżeli program nie powinien zakończyć działania.

PYTANIA

PYTANIE 1 Napisz program, który **nigdy** nie zakończy swojego działania, niezależnie od początkowych wartości komórek pamięci.

[1 punkt]

PYTANIE 2 Napisz program, który zakończy swoje działanie tylko i wyłącznie wtedy, gdy początkowa wartość komórki 1 będzie **większa** od 0. Wynikiem działania programu powinno być wtedy 0. [2 punkty]

`@1`	WYNIK
1	0
0	*brak*
-1	*brak*

PYTANIE 3 Napisz program, którego wynikiem będzie różnica początkowych wartości komórek 1 i 2. Program powinien zakończyć działanie tylko i wyłącznie wtedy, gdy wynik odejmowania będzie większy lub równy 0, w przeciwnym razie program nie powinien się nigdy zakończyć. [3 punkty]

`@1`	`@2`	WYNIK	UWAGI
5	3	2	
5	5	0	
5	8	*brak*	wynik odejmowania jest ujemny.

PYTANIE 4 Napisz program, którego wynikiem będzie 1, jeżeli początkowa wartość komórki 2 jest dzielnikiem początkowej wartości komórki 1 (t.j. ich reszta z dzielenia wynosi 0). W przeciwnym razie, wynikiem działania programu powinno być 0. [4 punkty]

`@1`	`@2`	WYNIK	UWAGI
9	3	1	3 jest dzielnikiem 9
10	3	0	3 nie jest dzielnikiem 10

PYTANIE 5 Napisz program, którego wynikiem będzie liczba dzielników liczby znajdującej się początkowo w komórce 1. Jeżeli wartość w komórce 1 jest ujemna, wynikiem działania programu powinno być 0. [7 punktów]

`@1`	WYNIK	UWAGI
12	6	12 posiada 6 dzielników: 1, 2, 3, 4, 6, 12
1	1	1 posiada tylko 1 dzielnik: 1
-5	0	wartość w komórce 1 jest ujemna

PYTANIE 6 Zmień swój program z poprzedniego pytania, aby jego wynikiem było 1, jeżeli początkowa wartość komórki 1 jest dodatnia i jest liczbą pierwszą. W przeciwnym razie wynikiem działania programu powinno być 0.

[3 punkty]

`@1`	WYNIK	UWAGI
12	0	liczba 12 nie jest pierwsza
13	1	liczba 13 jest pierwsza
-5	0	wartość w komórce 1 jest ujemna
1	0	liczba 1 **nie jest pierwsza**

ZADANIE 3

Funkcje operujące na listach

W tym zadaniu będziemy zapisywać wzory funkcji, które operują na **listach** liczb naturalnych. Lista to pewien skończony, uporządkowany zestaw elementów, na przykład $[1, 0, 6]$.

Skończony, to jest posiadający koniec – wszystkie elementy dowolnej listy możemy wypisać na kartce papieru. W szczególności lista może być pusta, co zapisujemy $[]$.

Uporządkowany, to jest taki, w którym kolejność elementów ma znaczenie – stąd lista $[1, 0, 6]$ nie jest tą samą listą co $[6, 1, 0]$.

Mamy do dyspozycji kilka podstawowych funkcji, które działają na listach:

- $L(n)$ – funkcja **tworząca listy**, której wartością jest lista liczb od 1 do n, na przykład $L(3) = [1, 2, 3]$, $L(0) = []$. W ogólności:

$$L(n) = [1, 2, \ldots, n]$$

- $S(l)$ – funkcja **sumująca**, której wartością jest suma elementów listy l. Na przykład $S([1, 0, 6]) = 7$. Suma elementów pustej listy to 0.

- $F(f, l)$ – funkcja **filtrująca**, której wartością jest lista tych elementów l, dla których funkcja f przyjmuje wartości różne od 0. Kolejność elementów powinna być zachowana. Na przykład:

$$f(x) = x$$
$$F(f, [1, 0, 6]) = [1, 6]$$

- $M(f, l)$ – funkcja **mapująca**, której wartością jest lista składająca się z wartości funkcji f dla kolejnych elementów listy l. To jest:

$$M(f, [l_1, \ldots, l_n]) = [f(l_1), \ldots, f(l_n)]$$

  Na przykład:

$$f(x) = 2x$$
$$F(f, [1, 0, 6]) = [2, 0, 12]$$

Używając tych podstawowych funkcji, możemy budować bardziej złożone funkcje działające na listach liczb naturalnych. Na przykład gdybyśmy chcieli opisać wzorem funkcję $Z(l)$, która usuwa z listy l wszystkie zera, moglibyśmy napisać:

$$f(x) = x$$
$$Z(l) = F(f, l)$$

Uwaga! W odpowiedziach zapisz wzory **wszystkich** użytych funkcji, innych niż L, S, F, M, chyba że w treści pytania napisano inaczej. W swoich wzorach możesz używać operacji arytmetycznych, jak dodawanie, odejmowanie, mnożenie i dzielenie, chyba że napisano inaczej.

PYTANIA

PYTANIE 1 Dana jest lista $[7, 2, 3]$. Wypisz dwie różne przykładowe listy, które zawierają te same elementy co lista $[7, 2, 3]$, jednak są różne od tej listy.
[2 punkty]

PYTANIE 2 Zapisz wzór funkcji $s(n)$, której wartością jest suma liczb od 1 do n. Nie używaj żadnych operacji arytmetycznych. [2 punkty]

PYTANIE 3 Zapisz wzór funkcji $d(l)$, której wartością jest długość (liczba elementów) listy l. Pusta lista ma długość 0.
[3 punkty]

PYTANIE 4 Zapisz wzór funkcji $z(l)$, której wartością jest liczba zer w danej liście.
[4 punkty]

PYTANIE 5 Funkcja $r(a, b)$, której wzoru nie musisz podawać, zwraca resztę z dzielenia a przez b, na przykład $r(8, 3) = 2$. Zapisz wzór funkcji $P(l)$, której wartością będą parzyste elementy listy l. [4 punkty]

PYTANIE 6 Wszystkie dotychczas zapisywane przez nas funkcje posiadały nazwy, jak S, d, r czy f. Funkcje mogą jednak być **anonimowe**, czyli nie posiadać nazwy. Zapisujemy je wtedy przy użyciu strzałki, po lewej stronie podając argument funkcji, a po prawej jej wartość. Na przykład $x \to 3x + 1$. Funkcja ta różni się od $f(x) = 3x + 1$ jedynie tym, że nie posiada nazwy: przyjmuje te same argumenty i dla tych samych argumentów zwraca te same wartości.

Dzięki temu, na przykład, dwa wzory użyte w treści zadania do opisania funkcji Z możemy zastąpić jednym: $Z(l) = F(x \to x, l)$.

Używając anonimowych funkcji, zapisz wzór funkcji $m(a, b)$, której wartością będzie iloczyn liczb a i b. [5 punktów]