

ZADANIE

Osiągalne (treść, rozwiązanie i ocena jakości)

Otwarto: poniedziałek, 10 stycznia 2022, 08:30

Wymagane do: wtorek, 25 stycznia 2022, 20:00

Wprowadzenie

Powiemy, że w grafie skierowanym węzeł **b** jest osiągalny z węzła **a**, jeśli:

- **a** i **b** są tym samym węzłem lub
- istnieje krawędź z węzła **a** do węzła **c** i węzeł **b** jest osiągalny z węzła **c**.

Polecenie

Napisz program, który czyta opis grafu skierowanego w podzbiorze języka DOT i pisze liczbę węzłów osiągalnych z węzła o nazwie "**start**".

Postać danych

Język danych jest opisany gramatyką w rozszerzonej notacji BNF:

```
<graf> ::= DIGRAPH LEWY { ID STRZALKA ID } PRAWY
```

Symbolami końcowymi gramatyki są:

- **DIGRAPH**

ciąg liter opisany wyrażeniem regularnym:

```
[Dd][Ii][Gg][Rr][Aa][Pp][Hh]
```

gdzie **[Dd]** oznacza małą lub wielką litera **d** itd.,

- **LEWY**

znak '{',

- **PRAWY**

znak '}',

- **STRZALKKA**

ciąg znaków "->",

- **ID**

ciąg wielkich lub małych liter i cyfr, zaczynający się od litery, który nie jest żadnej z postaci:

- **[Dd][Ii][Gg][Rr][Aa][Pp][Hh]**
- **[Ee][Dd][Gg][Ee]**
- **[Gg][Rr][Aa][Pp][Hh]**
- **[Nn][Oo][Dd][Ee]**
- **[Ss][Tt][Rr][Ii][Cc][Tt]**
- **[Ss][Uu][Bb][Gg][Rr][Aa][Pp][Hh]**

i po którym w danych występuje znak inny niż litera lub cyfra.

W danych, przed i po symbolach końcowych, mogą wystąpić, w dowolnej liczbie, spacje i końce wiersza.

Symbol **ID** reprezentuje nazwę węzła.

Jeden z węzłów grafu ma nazwę "**start**".

Postać wyniku

Wynikiem programu jest jeden wiersz z zapisaną dziesiętnie liczbą węzłów osiągalnych z węzła o nazwie "**start**".

Przykłady

Do treści zadania dołączone są pliki **.in** z przykładowymi danymi i pliki **.out** z wynikami wzorcowymi.

- Dla danych [przykład1.in](#) poprawny wynik to [przykład1.out](#).
- Dla danych [przykład2.in](#) poprawny wynik to [przykład2.out](#).
- Dla danych [przykład3.in](#) poprawny wynik to [przykład3.out](#).

Walidacja i testy

- Rozwiązania podlegają walidacji, wstępnie badającej zgodność ze specyfikacją.

Walidacja sprawdza działanie programu na przykładach dołączonych do treści zadania.

Pomyślne przejście walidacji jest warunkiem dopuszczenia programu do testów poprawności. Program, który walidacji nie przejdzie, dostaje zerową ocenę poprawności.

- Walidacja i testy są prowadzone na komputerze **students**.
- Programy są kompilowane poleceniem:

```
gcc @opcje nazwa.c -o nazwa
```

gdzie **nazwa.c** to nazwa pliku z kodem źródłowym, a plik **opcje** ma zawartość:

```
-std=c17
-pedantic
-Wall
-Wextra
-Wformat-security
-Wduplicated-cond
-Wfloat-equal
-Wshadow
-Wconversion
-Wjump-misses-init
-Wlogical-not-parentheses
-Wnull-dereference
-Wvla
-Werror
-fstack-protector-strong
-fsanitize=undefined
-fno-sanitize-recover
-g
-fno-omit-frame-pointer
-O1
```

Opcje **-std=c17**, **-pedantic** wskazują, że kompilator ma dbać o zgodność kodu z aktualnym standardem języka C.

Dzięki opcjom **-Wall**, **-Wextra** kompilator zgłosi zauważone usterki.

Opcje **-Wformat-security**, **-Wduplicated-cond**, **-Wfloat-equal**, **-Wshadow**, **-Wconversion**, **-Wjump-misses-init**, **-Wlogical-not-parentheses**, **-Wnull-dereference** umożliwiają wykrywanie dodatkowych usterek.

Opcja **-Wvla** sprawia, że użycie tablic zmiennej długości jest uznawane za usterkę.

Opcja **-Werror** wskazuje, że kompilator ma uznać usterki za błędy.

Dzięki opcji **-fstack-protector-strong**, podczas wykonania programu zostaną wykryte niektóre błędne odwołania do pamięci na stosie.

Opcje **-fsanitize=undefined**, **-fno-sanitize-recover** umożliwiają wykrywanie operacji, które mają efekt nieokreślony.

Opcje **-g**, **-fno-omit-frame-pointer** poprawiają jakość komunikatów o błędach wykonania.

Opcja **-O1** włącza optymalizacje, co zwiększa prawdopodobieństwo ujawnienia się błędów.

Wymagane są wszystkie wymienione opcje kompilatora. Nie będą do nich dodawane żadne inne.