6. Identyfikatory

1 Zadanie

1.1 Zliczanie komentarzy

Szablon programu należy uzupełnić o definicję funkcji `find_comments()`, która czyta ze standardowego wejścia ciąg znaków stanowiący program w języku C. Funkcja zlicza komentarze blokowe (`/* ... */`) i jednoliniowe (`// ...`) w przeczytanym tekście i zwraca uzyskane liczby do funkcji `main()` przy użyciu parametrów.

Zagnieżdżone komentarze nie są liczone, czyli np. następujący fragment kodu:

```
/*// tekst*/
```

jest uważany za jeden komentarz blokowy.

Można założyć, że wszystkie komentarze blokowe są prawidłowo zamknięte.

- **Wejście**
  1
  `linie tekstu`
- **Wyjście**
  Liczba komentarzy blokowych i liniowych
- **Przykład:**
  Wejście:

  ```
  1
  int main() { // comment
      printf ("Hello\n"); /* and another */
      return 0;
      /* and more
      and more ...
      */
  }
  ```

  Wyjście: 2 1

1.2 Znajdowanie identyfikatorów

Szablon programu należy uzupełnić o definicję funkcji `find_idents()`, która czyta ze standardowego wejścia ciąg znaków stanowiący program w języku C. Funkcja zlicza **unikalne** identyfikatory zawarte w przeczytanym tekście i zwraca uzyskaną liczbę do funkcji `main()`.

Definicja identyfikatora jest taka jak w języku C, czyli jest to ciąg liter i cyfr zaczynający się od litery; znak podkreślnika jest uważany za literę, czyli na przykład identyfikator `_a` jest legalny.

Uwaga 1: W programie wejściowym mogą znajdować się stałe znakowe, napisowe (ciągi znaków ujęte w cudzysłowy) oraz komentarze (jak w zadaniu o komentarzach). Powinny one być pomijane przy liczeniu identyfikatorów.

Uwaga 2: Można założyć, że w podanych tekstach wejściowych nie ma dyrektyw preprocesora.

Uwaga 3: Dla ujednolicenia templatka programu zawiera słowa kluczowe, które należy wyłączyć z listy identyfikatorów.

- **Wejście**
  2
  `linie tekstu`
- **Wyjście**
  liczba unikalnych identyfikatorów zawartych w tekście nie będących słowami kluczowymi
- **Przykład:**
  Wejście:

```
2
int main() { // comment
    printf ("Hello\n"); /* and another */
    printf ("Greetings\n");
    return 0;
}
```

  Wyjście: 2