

[Dashboard](#) / [My courses](#) / [ITB_IF2110_1_2526](#) / [Praktikum 8](#) / [Post Praktikum 8](#)

Started on	Sunday, 16 November 2025, 11:00 PM
State	Finished
Completed on	Sunday, 16 November 2025, 11:59 PM
Time taken	58 mins 49 secs
Marks	366.00/400.00
Grade	9.15 out of 10.00 (92%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: reversewords.c

Anda diberikan sebuah sekuens yang terdiri atas n buah kata, dipisahkan oleh satu atau lebih *whitespace*. Sekuens ini diakhiri oleh **MARK** berupa "." (titik) yang bukan merupakan bagian dari kata manapun.

Tugas anda adalah membalikkan setiap karakter pada setiap kata dengan tetap mempertahankan urutan kata awal. Setelah memproses semua kata, keluarkan setiap kata terbalik dengan dipisahkan oleh *new line* ("\\n").

Format Masukan:

- Baris pertama berisi sebuah sekuens yang terdiri atas n buah kata dan dipisahkan oleh *whitespace(s)* ($1 \leq n \leq 1000$).

Format Keluaran:

- Keluaran berupa n buah baris, dengan baris ke- i berisi kata ke- i yang sudah dibalik.

Contoh Masukan dan Keluaran:

No	Masukan	Keluaran
1.	hello world abc def .	olleh dlrow cba fed
2.	example test case.	elpmaxe tset esac

- Pastikan setiap output diakhiri oleh **endline** ("\\n")!
- Gunakan ADT mesin kata dan mesin karakter pada file [berikut](#).

C

 [reversewords.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.54 MB
2	10	Accepted	0.00 sec, 1.51 MB
3	10	Accepted	0.00 sec, 1.54 MB
4	10	Accepted	0.00 sec, 1.50 MB
5	10	Accepted	0.00 sec, 1.54 MB
6	10	Accepted	0.00 sec, 1.54 MB
7	10	Accepted	0.00 sec, 1.50 MB
8	10	Accepted	0.00 sec, 1.50 MB
9	10	Accepted	0.00 sec, 1.49 MB
10	10	Accepted	0.00 sec, 1.51 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Berikut diberikan ADT [berikut](#) yang berisi mesin kata dan mesin karakter. Lengkapi dan kumpulkan kembali file [berikut](#)!

C

 [palindrome.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.60 MB
2	10	Accepted	0.00 sec, 1.58 MB
3	10	Accepted	0.00 sec, 1.53 MB
4	10	Accepted	0.00 sec, 1.60 MB
5	10	Accepted	0.00 sec, 1.64 MB
6	10	Accepted	0.00 sec, 1.58 MB
7	10	Accepted	0.00 sec, 1.59 MB
8	10	Accepted	0.00 sec, 1.64 MB
9	10	Accepted	0.00 sec, 1.55 MB
10	10	Accepted	0.00 sec, 1.57 MB

Question **3**

Partially correct

Mark 66.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: problem.c

Anda diminta mengimplementasikan **grammar parser** -- dengan menggunakan [mesinkata.h](#) dan [mesinkarakter.h](#) yang telah anda implementasikan sebelumnya -- untuk memeriksa apakah sebuah kalimat (dibaca dari *stdin* menggunakan mesin kata) valid terhadap definisi grammar berikut.

- **<Kalimat> ::= <Subjek> <Predikat> <Objek> { , <Objek> }***
- **<Subjek> ∈ { "aku", "kamu", "dia" }**
- **<Predikat> ∈ { "beli", "ambil" }**
- **<Objek>** adalah satu kata alfabet (A–Z/a–z), tanpa spasi di dalamnya.

Yang Harus Diimplementasikan

Lengkapi fungsi berikut pada berkas [problem.c](#):

```
int cfg_parser() {
    STARTWORD();
    // TODO: Implementasi fungsi cfg_parser
}
```

Output Fungsi

- Kembalikan **1** jika kalimat valid sesuai grammar.
- Kembalikan **0** jika tidak valid.

Contoh

Input : "aku beli buku."
Output : 1

Input : "aku beli buku, pensil, penggaris."
Output : 1

Input : "dia ambil tas , buku."
Output : 1

Input : "Dia ambil tas , buku, penggariss ."
Output : 1

Input : "aku beli ." Output : 0 // tidak ada objek
Input : "beli buku." Output : 0 // tidak ada subjek
Input : "aku beli , pensil." Output : 0 // koma langsung setelah predikat
Input : "aku beli buku, ." Output : 0 // koma menjadi token terakhir sebelum MARK

Catatan

- Urutan token harus persis: Subjek → Predikat → Objek (opsional diikuti urutan **,** **Objek** berulang).
- Objek dianggap valid jika seluruh karakternya alfabet (kecuali koma opsional di posisi paling akhir token).
- Parser disarankan dibuat dengan **state transition / FSM** agar mudah menangani kasus koma menempel vs koma terpisah.

Hint Singkat

- Buat state: **S_EXPECT_SUBJECT**, **S_EXPECT_PREDICATE**, **S_EXPECT_FIRST_OBJECT**, **S_EXPECT_SEPARATOR_OR_END**, **S_EXPECT_OBJECT_AFTER_COMMA**.
- Deteksi *token* koma: **w.Length == 1 && w.TabWord[0] == ','**.
- Deteksi objek dengan koma menempel: jika karakter terakhir **' , '**, pastikan sisanya alfabet dan tandai “butuh objek lagi”.
- Kalimat valid jika saat mencapai **EndWord == true** dan program sedang berada pada state **S_EXPECT_SEPARATOR_OR_END** dan minimal satu objek sudah terbaca.

C

 [parser.c](#)

Score: 66

Blackbox

Score: 66

Verdict: Time limit exceeded

Evaluator: Exact

No	Score	Verdict	Description
1	6	Accepted	0.00 sec, 1.49 MB
2	6	Accepted	0.00 sec, 1.57 MB
3	6	Accepted	0.00 sec, 1.64 MB
4	6	Accepted	0.00 sec, 1.57 MB
5	6	Accepted	0.00 sec, 1.55 MB
6	6	Accepted	0.00 sec, 1.55 MB
7	6	Accepted	0.00 sec, 1.57 MB
8	6	Accepted	0.00 sec, 1.56 MB
9	6	Accepted	0.00 sec, 1.56 MB
10	0	Time limit exceeded	1.00 sec, 1.46 MB
11	6	Accepted	0.00 sec, 1.60 MB
12	0	Time limit exceeded	1.00 sec, 1.49 MB
13	6	Accepted	0.00 sec, 1.59 MB
14	0	Time limit exceeded	1.00 sec, 1.51 MB
15	0	Wrong answer	0.00 sec, 1.57 MB

Question **4**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: telegram.c

Para Minions sedang bekerja di kantor pos kuno Gru. Mereka menemukan mesin tua pengirim pesan bernama Telegram 3000. Mesin ini bisa mengirim pesan super cepat lewat kabel, tapi hanya jika pesan ditulis dengan format telegram klasik.

Sayangnya, mesin itu tidak bisa menghitung biaya pengiriman secara otomatis. Tugas Anda adalah membantu para Minions menulis program untuk **membaca isi telegram** dan **menghitung total biayanya**.

Telegram dikirim sebagai deretan kata yang dibaca menggunakan **mesinkata.h** dan **mesinkarakter.h**. Tanda baca ditulis dalam bentuk kata:

- **KOMA** untuk tanda koma (,)
- **TITIK** untuk tanda titik (.)

Aturan Pengiriman Telegram

- Setiap kata dipisahkan oleh satu atau lebih spasi.
- Pita karakter diakhiri dengan karakter titik (.).
- Tanda baca tidak dihitung sebagai kata berbayar.
- Kata reguler (panjang < 10 karakter) dikenakan biaya Rp1000.
- Kata panjang (panjang ≥ 10 karakter) dikenakan biaya Rp1500.
- Jika total kata (termasuk tanda baca) > 10, Minions mendapatkan diskon 10%.

Output Program

- Isi telegram setelah tanda baca dikonversi (**KOMA** → , dan **TITIK** → .)
- Jumlah kata reguler dan kata panjang
- Total biaya yang harus dibayar

Contoh Kasus

No	Masukan	Keluaran	Keterangan
1	SEGERA PULANG KOMA ADIK SAKIT TITIK.	SEGERA PULANG, ADIK SAKIT. 4 0 4000	4 kata reguler (SEGERA , PULANG , ADIK , SAKIT) × 1000 = 4000 Tidak ada kata panjang. Total biaya = 4000.
2	APA KABAR TITIK SAYA SENANG MENERIMA TAWARANNYA KOMA SEGERA KIRIM PROPOSAL KE KANTOR BANDUNG TITIK.	APA KABAR. SAYA SENANG MENERIMA TAWARANNYA, SEGERA KIRIM PROPOSAL KE KANTOR BANDUNG. 11 1 11250	11 kata reguler × 1000 = 11000 1 kata panjang (TAWARANNYA) × 1500 = 1500 Total = 12500 – 10% diskon = 11250
3	TITIK KOMA A KOMA TITIK TITIK KOMA KOMA TITIK TITIK TITIK.	., A, ., ., . 1 0 900	1 kata reguler (A) × 1000 = 1000 Total kata termasuk tanda baca = 12 → diskon 10% Total biaya = 1000 – 10% = 900

Catatan

- Tanda baca menempel pada kata sebelumnya dan dipisahkan satu spasi dengan kata berikutnya.

Kumpulkan hasil implementasi file [telegram.c](#) ini

C

 [telegram.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	8	Accepted	0.00 sec, 1.50 MB
2	8	Accepted	0.00 sec, 1.57 MB
3	8	Accepted	0.00 sec, 1.56 MB
4	8	Accepted	0.00 sec, 1.53 MB
5	8	Accepted	0.00 sec, 1.55 MB
6	8	Accepted	0.00 sec, 1.54 MB
7	8	Accepted	0.00 sec, 1.53 MB
8	8	Accepted	0.00 sec, 1.54 MB
9	8	Accepted	0.00 sec, 1.52 MB
10	8	Accepted	0.00 sec, 1.45 MB
11	8	Accepted	0.00 sec, 1.64 MB
12	12	Accepted	0.00 sec, 1.65 MB

[◀ Praktikum 8](#)

Jump to...

[Praktikum 9 ▶](#)