Started on	Thursday, 24 April 2025, 1:34 AM
State	Finished
Completed on	Thursday, 24 April 2025, 1:57 AM
Time taken	23 mins 8 secs
Marks	400.00/440.00
Grade	**10.00** out of 11.00 (**91**%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: MisiRahasia.c

Suatu hari, Tuan Gro memberikan misi rahasia kepada para nimons. Tiap nimon diberikan 1 buah misi rahasia yang memiliki nilai keuntungan tertentu jika berhasil diselesaikan. Setelah menjalankan misi, para nimon melaporkan data misi mereka kepada Tuan Gro. Gro kemudian menyimpan data misi tersebut ke dalam sebuah file teks dengan nama `input.txt`. Setiap baris pada file ini menyimpan informasi satu misi dengan format berikut:

```
<NimonID> <Success(0/1)> <MissionValue>
```

Adapun *mark* yang menandakan *end of file* (EOF) adalah record berikut:

```
-1 0 0
```

Berikut adalah contoh isi teks file laporan:

```
1 1 60
2 1 20
3 0 100
4 1 30
-1 0 0
```

Tuan Gro meminta bantuan anda untuk menuliskan *summary report* berdasarkan laporan misi. Anda diminta mengeluarkan banyak misi keseluruhan, banyak misi yang berhasil dikerjakan, dan total *mission value* yang diperoleh. Total *mission value* dijumlahkan hanya dari misi yang berhasil saja.

Berikut adalah template driver yang digunakan pada soal ini. Perhatikan bahwa file `MisiRahasia.c` wajib meng-*include* header `tester.h` dan memanggil fungsi `init()` di awal `main()` seperti pada contoh driver, kecuali jika ingin menjalankan file di komputer lokal.

Batasan Masukan File:

- 1 ≤ `NimonID` ≤ 1000, dan dipastikan unik
- 0 ≤ `Success` ≤ 1
- 1 ≤ `MissionValue` ≤ 1000

Format Keluaran:

- Baris pertama berisi string `"Total Misi: "` dan sebuah bilangan bulat yang menyatakan banyak misi keseluruhan.
- Baris kedua berisi string `"Misi Berhasil: "` dan sebuah bilangan bulat yang menyatakan banyak misi yang berhasil dikerjakan.
- Baris ketiga berisi string `"Total Nilai Misi: "` dan sebuah bilangan bulat yang menyatakan total mission value yang diperoleh.

Contoh Masukan dan Keluaran:

No	Masukan Pada File	Keluaran

1.	1 1 60
	2 1 20
	3 0 100
	4 1 30
	-1 0 0
		`Total Misi: 4`
		`Misi Berhasil: 3`
		`Total Nilai Misi: 110`

Pastikan setiap output diakhiri oleh **endline ("\n")**!

C

MisiRahasia.c

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 1.68 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: MisiRahasia.c

Suatu hari, Tuan Gro memberikan misi rahasia kepada para nimons. Tiap nimon diberikan 1 buah misi rahasia yang memiliki nilai keuntungan tertentu jika berhasil diselesaikan. Setelah menjalankan misi, para nimon melaporkan data misi mereka kepada Tuan Gro. Gro kemudian menyimpan data misi tersebut ke dalam sebuah file teks dengan nama `input.txt`. Setiap baris pada file ini menyimpan informasi satu misi dengan format berikut:

```
<NimonID> <Success(0/1)> <MissionValue>
```

Adapun *mark* yang menandakan *end of file* (EOF) adalah record berikut:

```
-1 0 0
```

Berikut adalah contoh isi teks file laporan:

```
1 1 60
2 1 20
3 0 100
4 1 30
-1 0 0
```

Tuan Gro memiliki sebuah file bernama `query.txt` yang berisi *Q* buah kueri. Untuk setiap kueri, ia ingin mengetahui ID dan Mission Value dari nimons yang berada pada urutan ke-*k* setelah dilakukan pengurutan data misi nimon berdasarkan prioritas tertentu.

Untuk itu, data misi para nimon harus diurutkan terlebih dahulu berdasarkan prioritas berikut, dari yang paling tidak penting hingga paling penting (pengurutan dilakukan secara menaik):

1. Nimon yang berhasil, memiliki prioritas yang lebih tinggi dibandingkan nimon yang gagal menyelesaikan misi.
2. Nimon yang memiliki mission value lebih tinggi, memiliki prioritas yang lebih tinggi dibandingkan nimon dengan mission value yang lebih rendah.
3. Nimon yang memiliki ID lebih tinggi, memiliki prioritas yang lebih tinggi dibandingkan nimon dengan ID lebih rendah.

Adapun format untuk isi file `query.txt` adalah sebagai berikut:

- Baris pertama adalah sebuah bilangan bulat *Q* yang menyatakan banyak kueri.
- Sebanyak Q baris berikutnya berisi sebuah bilangan bulat k_i yang menyatakan kueri ke-i.

Berikut adalah template driver yang digunakan pada soal ini. Perhatikan bahwa file `MisiRahasia.c` wajib meng-*include* header `tester.h` dan memanggil fungsi `init()` di awal `main()` seperti pada contoh driver, kecuali jika ingin menjalankan file di komputer lokal.

Batasan Masukan File input.txt:

- 1 ≤ `Banyak Nimon` ≤ 1000
- 1 ≤ `NimonID` ≤ 1000, dan dipastikan unik
- 0 ≤ `Success` ≤ 1
- 1 ≤ `MissionValue` ≤ 1000

Batasan Masukan File query.txt:

- 1 ≤ *Q* ≤ 1000
- 1 ≤ *k* ≤ 1000

Format Keluaran:

- Q baris berisi dua buah bilangan bulat A dan B (1 ≤ *A*, *B* ≤ 1000) yang berturut-turut mewakili ID dan *mission value* nimon terbesar ke-k_i.

Batasan File:

- 1 ≤ `Banyak Nimon` ≤ 1000
- 1 ≤ `NimonID` ≤ 1000
- 0 ≤ `Success` ≤ 1
- 1 ≤ `MissionValue` ≤ 1000

Contoh Masukan dan Keluaran:

No	Isi input.txt	Isi query.txt	Keluaran
1.	1 1 60 2 1 20 3 0 100 4 1 30 -1 0 0	2 1 2	3 100 2 20

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	100 s
Memory limit	64 MB

Nama File: GabungNimons.c

Tuan Gro memiliki dua file berisi data hasil panen pisang dari dua kelompok nimons. Setiap file berisi informasi hasil panen dari beberapa nimon yang sudah diurut berdasarkan **ID Nimon (menaik)**.

Setiap baris dalam file memiliki format berikut:
`<NimonID> <JumlahPisang>`

Setiap file diakhiri dengan mark berikut:
`-1 99`

Tuan Gro meminta bantuan Anda untuk **menggabungkan isi kedua file tersebut dan menampilkannya ke terminal**, dengan data tetap dalam urutan menaik berdasarkan NimonID.

Format Masukan:

Masukan berupa 2 file bernama:

- `file_a.txt` (nama file pertama)
- `file_b.txt` (nama file kedua)

Format Keluaran:

Cetak ke layar isi gabungan dari kedua file dalam urutan menaik berdasarkan NimonID. Mark (`-1 99`) **tidak perlu ditampilkan** dalam output.

Batasan:

- 1 ≤ NimonID ≤ 1000
- 1 ≤ JumlahPisang ≤ 100
- Kedua file sudah **terurut menaik berdasarkan NimonID**
- Tidak ada duplikat NimonID antar file
- Minimal berisi mark (`-1 99`) pada setiap input file

Contoh Masukan dan Keluaran:

No	Masukan	Keluaran	Keterangan
1.		1 20 2 30 3 50 4 45	isi file_a.txt 2 30 4 45 -1 99 isi file_b.txt 1 20 3 50 -1 99

Pastikan setiap output diakhiri oleh **endline ("\n")**!

Notes:

- Gunakan fungsi `fscanf` dan `printf`, serta `fopen`, `fclose` untuk pemrosesan file.

GUNAKAN STRUKTUR KODE SEBAGAI BERIKUT SAAT MELAKUKAN SUBMISI KE OLYMPIA

```c
#include <stdio.h>
#include "tester.h" // Comment this when developing in local
int main () {
    init(); // Comment this when developing in local

    /* Write your code here */
    return 0;
}
```

C

GabungNimons.c

Question **4**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: akademinimons.c

Di Laboratorium Gro, para Nimons tidak hanya bertugas untuk membantu eksperimen jahat, tetapi mereka juga memiliki sistem pendidikan sendiri yang disebut "Akademi Nimons". Setiap Nimon memiliki Nomor Identitas Mahasiswa (NIM) unik dan mengikuti berbagai kursus.

Professor Neroifa, sebagai kepala akademik, membutuhkan sistem untuk mengolah nilai-nilai para Nimons ini. Setiap Nimon mengikuti beberapa ujian dalam setiap kursus, dan Profesor Neroifa perlu tahu rata-rata nilai mereka, nilai tertinggi dan terendah, serta status kelulusan mereka (nilai minimal 60), persentase Nimons yang lulus dan tidak lulus, serta siapa Nimon dengan nilai rata-rata tertinggi.

Sebagai asisten baru Profesor Neroifa, tugas kamu adalah mengimplementasikan file header akademinimons.h pada file akademinimons.c. Untuk pengujian program tersebut, kamu bisa menggunakan file berikut driver.c.

Format Masukan:

Sudah dijelaskan pada akademinimons.h untuk format file dan juga driver.c untuk pengujian.

Format Keluaran:

Sudah dijelaskan pada akademinimons.h untuk format file**.**

Contoh Masukan dan Keluaran:

No	Masukan (Isi file)	Keluaran	Keterangan
1.	13522001|John Doe|85,92,78,88,95 13522002|Jane Smith|65,70,58,62,75 13522003|Bob Johnson|45,50,55,48,52 13522004|Alice Williams|92,95,98,90,94 13522005|Charlie Brown|72,68,75,70,78	13522001|John Doe|87.60|95|78|PASS 13522002|Jane Smith|66.00|75|58|PASS 13522003|Bob Johnson|50.00|55|45|FAIL 13522004|Alice Williams|93.80|98|90|PASS 13522005|Charlie Brown|72.60|78|68|PASS 5 4 80.00% 1 20.00% 13522004|Alice Williams|93.80	NIM 13522011: Nilai rata-rata = 87.60 Nilai tertinggi = 95 Nilai terendah = 78 Lulus, karena 87.60 >= 60 NIM 13522002: Nilai rata-rata = 66.00 Nilai tertinggi = 75 Nilai terendah = 58 Lulus, karena 66.00 >= 60 NIM 13522003: Nilai rata-rata = 50.00 Nilai tertinggi = 55 Nilai terendah = 45 Tidak lulus, karena 55.00 < 60 NIM 13522004: Nilai rata-rata = 93.80 Nilai tertinggi = 98 Nilai terendah = 90 Lulus, karena 93.80 >= 60 NIM 13522005: Nilai rata-rata = 72.60 Nilai tertinggi = 78 Nilai terendah = 68 Lulus, karena 72.60 >= 60 JumlahTotalNimons = 5 JumlahLulus, % Lulus = 4, 80% JumlahTidakLulus, % Tidak Lulus = 1, 20% Nilai Rata2 Tertinggi diraih 13522004 sebesar 93.80.

Pastikan setiap output diakhiri oleh **endline ("\n")**!

C

akademinimons.c

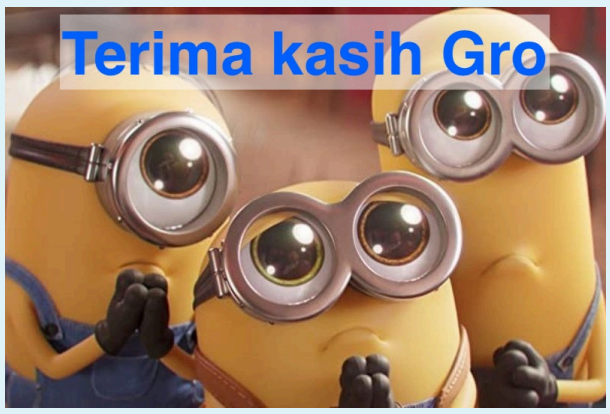
Question **5**

Not answered

Marked out of 40.00

Time limit	10 s
Memory limit	64 MB

Karena Asisten sangat baik hatinya, asisten menyarankan Gro untuk tidak mempersulit Nimons dalam mengerjakan task ini. Oleh karena itu, Gro akan menganggap task ini sebagai task bonus yang tidak wajib dikerjakan.



Nama File: server.c

Gro sedang sibuk mengembangkan teknologi komunikasi baru untuk markas rahasia Nimons. Dia ingin supaya setiap pesan dari alat komunikasi para Nimons diproses seperti HTTP Request!

Untuk itu, Gro meminta bantuan Kebin, Stewart, Pop, dan Toto untuk membuat server sederhana. Karena mereka belum siap menghadapi jaringan asli, mereka hanya akan memproses permintaan lewat file `request.txt`. Nantinya, pesan yang diterima akan diproses dan dibalas ke file `response.txt`.

Kamu ditugaskan oleh Dr. Neroifa untuk membantu Nimons menyelesaikan prototipe ini. Tugasmu adalah membaca file request, memahami isi perintahnya, dan menyusunnya kembali sebagai respons yang rapi!

Lengkapilah file `server.h`. Program akan menerima input berupa file HTTP request dan output disimpan di `response.txt` berupa HTTP response yang rapi. Tidak perlu membuat header tambahan.

Format Masukan:

- File `request.txt` di dalamnya terdapat HTTP request dengan format standar:
  - Baris pertama adalah: `METHOD PATH HTTP/1.1` (contoh: `POST /submit HTTP/1.1`)
  - Diikuti oleh beberapa header (masing-masing 1 baris), seperti: `Content-Type`, `Host`, `Content-Length`, dst.
  - Baris kosong (`\n`) untuk memisahkan header dengan body.
  - Diikuti body (jika ada).

Batasan:

- Panjang setiap baris maksimal 255 karakter.
- Body maksimal 500 karakter.

Format Keluaran:

File `response.txt` yang berisi:

- Baris status: `HTTP/1.1 200 OK`
- Header: `Content-Type: text/plain`
- Diikuti dengan informasi hasil parsing, seperti:
  - Method
  - Path
  - Content-Type (jika tidak ditemukan, tampilkan `N/A`)
  - Body (jika ada)

Contoh Masukan dan Keluaran:

No	Masukan (request.txt)	Keluaran (response.txt)

1.	`POST /submit HTTP/1.1`
	`Host: nimons.lab`
	`Content-Type: application/json`
	`Content-Length: 35`
	
	`{"name":"Pop","mission":"banana"}`
		`HTTP/1.1 200 OK`
		`Content-Type: text/plain`
		
		`Method: POST`
		`Path: /submit`
		`Content-Type: application/json`
		`Body:`
		`{"name":"Pop","mission":"banana"}`

Keterangan

- Gunakan `fgets` untuk membaca file baris demi baris, dan `strstr` atau `strncmp` untuk mendeteksi baris tertentu.
- Pastikan setiap output diakhiri oleh **endline ("\n")**