

Soal Praktikum #2

Perulangan dan Array

Tim Materi Berpikir Komputasional 2024/2025

30 Oktober 2024

Petunjuk

1. Kerjakan modul ini sesuai dengan materi yang diujikan (Perulangan dan Array). Tidak perlu menggunakan materi yang belum diujikan.
2. Perhatikan penamaan file terutama untuk ekstensi file (*.py). File tanpa ekstensi beresiko tidak dapat dibuka oleh asisten sehingga kode program tidak dapat dikoreksi (nilai 0)
3. Pastikan program lulus compile dan dapat dijalankan.
4. Untuk setiap file source code program berikan identitas, minimum:

NIM>Nama :
Tanggal :
Deskripsi :

5. Seluruh file kode program di-compress dengan nama **P02_NIM.zip** sebelum dikumpulkan.
6. Kecuali dituliskan secara khusus, Anda dapat menganggap masukan user sesuai dengan kehendak program.
7. Penulisan kode sebaiknya menggunakan indentasi yang baik dan menambahkan komentar (kegunaan sebuah variabel, percabangan, pengulangan, fungsi dan prosedur) sehingga mempermudah proses pencarian kesalahan pada program (debugging)
8. Kecurangan berupa copy-paste kode program dari peserta atau sumber lain akan memperoleh sanksi tegas.
9. Dilarang meng-capture atau menyebarkan sebagian dan/atau seluruh soal ini. Pelanggaran akan dikenakan sanksi sesuai SOP yang berlaku.
10. Jika ada perbedaan antara instruksi di sini dan instruksi asisten, ikuti instruksi asisten.
11. Selamat Mengerjakan!

Problem 1

Simpan dengan nama file: **P02_NIM.01.py**.

Mobil Tuan Leo sedang berada di sebuah tempat parkir, dan semua orang sedang mencarinya. Untuk membantu pencarian, Tuan Leo memberikan petunjuk berupa jumlah digit dari nomor plat mobilnya serta jumlah total digit pada nomor tersebut.

Sebagai salah satu pencari, bantu diri Anda sendiri untuk menemukan mobil Tuan Leo!

- Input program adalah nomor plat mobil Tuan Leo dalam format "DXXXXX...YYY...", dengan "XXXXX..." berupa digit angka dan "YYY..." berupa huruf.
- Plat nomor Tuan Leo selalu diawali dengan huruf "D".

Petunjuk: Untuk mengonversi string angka menjadi bilangan, Anda dapat menggunakan fungsi `int(angka)`. Misalnya, `int("1")` akan menghasilkan bilangan 1 (integer).

Catatan: Hati-hati saat menggunakan `int()` dengan input yang bukan angka, karena akan memunculkan pesan kesalahan `ValueError: invalid literal for int() with base 10: ""`.

Jumlah digit pada masukan selalu sesuai dengan yang diperlukan (masukan selalu valid).

Test Case 1

Masukkan nomor plat mobil: <u>D1234ABC</u> Masukkan jumlah digit: <u>10</u> Masukkan banyak digit: <u>4</u> Mobil Tuan Leo ditemukan!
--

Test Case 2

Masukkan nomor plat mobil: <u>D14RE</u> Masukkan jumlah digit: <u>8</u> Masukkan banyak digit: <u>2</u> Bukan mobil Tuan Leo!
--

Test Case 3

Masukkan nomor plat mobil: <u>B154YUK</u> Masukkan jumlah digit: <u>10</u> Masukkan banyak digit: <u>3</u> Bukan mobil Tuan Leo!

Penjelasan Test Case 3:

Plat nomor Tuan Leo seharusnya diawali dengan huruf D.

Problem 2

Simpan dengan nama file: **P02_NIM.02.py**.

Nona Sal merupakan seorang auditor internal yang diminta untuk mengecek penomoran invoice yang dikeluarkan oleh departemen di sebuah perusahaan. Berikut merupakan aturan penomoran invoice yang dikeluarkan dan diterima oleh departemen tersebut:

- Dua digit pertama adalah nomor dari departemen yang mengeluarkan invoice tersebut.
- Tiga digit selanjutnya adalah nomor urut invoice yang dikeluarkan oleh departemen tersebut.
- Dua digit selanjutnya adalah nomor dari departemen tujuan invoice tersebut.

Di perusahaan tersebut, terdapat 3 departemen yang bisa mengeluarkan dan tidak bisa menerima invoice, yaitu departemen 01, 09, dan 14 dari 30 departemen yang ada dengan contoh penomoran yang benar seperti 01/001/02. Saat melakukan internal audit, Nona Sal menemukan banyak penomoran invoice yang salah di basis data akibat kesalahan input oleh departemen. Karena sebentar lagi akan diadakan audit eksternal, Nona Sal meminta bantuan Anda untuk membantu memperbaiki penomoran invoice yang salah dan menandakan nomor invoice yang tidak valid untuk dihapus. Buatlah program yang dapat membantu Nona Sal untuk memperbaiki penomoran invoice yang salah tersebut!

Test Case 1

Masukkan jumlah invoice yang telah dikeluarkan oleh departemen 01: <u>208</u>
Masukkan jumlah invoice yang telah dikeluarkan oleh departemen 09: <u>228</u>
Masukkan jumlah invoice yang telah dikeluarkan oleh departemen 14: <u>198</u>
Masukkan nomor invoice yang akan dicek: <u>104/12/14</u>
Nomor invoice yang valid: 14/104/12

Penjelasan Test Case 1:

Pada nomor invoice yang dicek, departemen pengirim adalah 14, nomor urut invoice adalah 104, departemen penerima adalah 12. Karena seluruh informasi pada nomor invoice tersebut valid, maka nomor tersebut bisa diperbaiki dan dianggap valid.

Test Case 2

Masukkan jumlah invoice yang telah dikeluarkan oleh departemen 01: <u>124</u>
Masukkan jumlah invoice yang telah dikeluarkan oleh departemen 09: <u>031</u>
Masukkan jumlah invoice yang telah dikeluarkan oleh departemen 14: <u>295</u>
Masukkan nomor invoice yang akan dicek: <u>03/07/001</u>
Nomor invoice tersebut tidak valid.

Test Case 3

Masukkan jumlah invoice yang telah dikeluarkan oleh departemen 01: <u>236</u>
Masukkan jumlah invoice yang telah dikeluarkan oleh departemen 09: <u>348</u>
Masukkan jumlah invoice yang telah dikeluarkan oleh departemen 14: <u>676</u>
Masukkan nomor invoice yang akan dicek: <u>09/033/01</u>
Nomor invoice tersebut tidak valid.

Problem 3

Simpan dengan nama file: **P02_NIM_03.py**.

Nona Deb memiliki banyak koleksi buku di beberapa rak, namun sayangnya jumlah buku di setiap rak tidak teratur. Buku-bukunya tersebar dengan jumlah yang tidak konsisten dari rak ke rak, membuatnya sulit untuk dicari. Untuk merapikan tampilan raknya, Nona Deb ingin menyusun ulang buku-bukunya dalam bentuk tangga yang rapi (terurut naik atau turun) dan memiliki perbedaan maksimal 1 buku dibandingkan rak sebelumnya. Selain itu, setiap rak harus memiliki setidaknya satu buku setelah ditata ulang.

Bantu Nona Deb merapikan jumlah buku di setiap raknya! Sebagai catatan, kombinasi yang valid adalah kombinasi yang memaksimalkan selisih buku antara rak pertama dengan rak yang terakhir

Test Case 1

Jumlah rak yang dimiliki: 5
Jumlah buku di rak ke-1: 1
Jumlah buku di rak ke-2: 2
Jumlah buku di rak ke-3: 4
Jumlah buku di rak ke-4: 1
Jumlah buku di rak ke-5: 2
Ingin diurutkan Menaik/Menurun? Menaik
Urutan buku yang baru adalah [1, 2, 2, 2, 3].

Penjelasan Test Case 1:

Total buku yang dimiliki adalah 10 dan belum terurut menaik sehingga harus disusun ulang. Sehingga setelah disusun ulang menjadi [1, 2, 2, 2, 3].

Test Case 2

Jumlah rak yang dimiliki: 4
Jumlah buku di rak ke-1: 6
Jumlah buku di rak ke-2: 1
Jumlah buku di rak ke-3: 4
Jumlah buku di rak ke-4: 2
Ingin diurutkan Menaik/Menurun? Menaik
Urutan buku yang baru adalah [2, 3, 4, 4].

Test Case 3

Jumlah rak yang dimiliki: 5
Jumlah buku di rak ke-1: 5
Jumlah buku di rak ke-2: 1
Jumlah buku di rak ke-3: 3
Jumlah buku di rak ke-4: 3
Jumlah buku di rak ke-5: 1
Ingin diurutkan Menaik/Menurun? Menurun
Urutan buku yang baru adalah [4, 3, 3, 2, 1].