

Time limit	1 s
Memory limit	64 MB

Kelas **HideNSeek** merupakan sebuah kelas yang memodelkan permainan Petak Umpet

Buatlah sebuah kelas **HideNSeek** yang memiliki atribut:

1. **seeker** dengan tipe data **String**, digunakan untuk menyimpan nama seeker.
2. **player** dengan tipe data **int**, digunakan untuk menyimpan banyak pemain selain seeker.
3. **playerFound** dengan tipe data **int**, digunakan untuk menyimpan jumlah pemain yang telah ditemukan oleh seeker.

Lengkapi dan submit file [HideNSeek.java](#)

Java 8 ↕

 [HideNSeek.java](#)

Time limit	1 s
Memory limit	64 MB

Kelas **CarDealer** merupakan sebuah kelas yang memodelkan dealer mobil

Buatlah sebuah kelas **CarDealer** yang memiliki atribut:

1. **brand** dengan tipe data **String**, digunakan untuk menyimpan nama brand mobil.
2. **carInDealer** dengan tipe data **int**, digunakan untuk menyimpan banyak mobil yang ada di dealer.
3. **carPrice** dengan tipe data **int**, digunakan untuk menyimpan harga mobil.
4. **profit** dengan tipe data **int**, digunakan untuk menyimpan total uang yang didapatkan dari menjual mobil.
5. **totalCar** dengan tipe data **int** dan merupakan static variable, digunakan untuk menyimpan total mobil yang ada di semua dealer.

Lengkapi dan submit file [CarDealer.java](#)

Java 8 ↕

 [CarDealer.java](#)

Time limit	1 s
Memory limit	64 MB

Ibnu ingin bermain petak umpet bersama teman-temannya. Namun, karena di luar sedang terjadi tornado, ia dan teman-temannya terpaksa bermain petak umpet secara virtual.

Buatlah program untuk Ibnu dan teman-temannya bermain petak umpet virtual. Beri program dengan nama file **GameSimulator.java** dan gunakan kelas/file **HideNSeek.java** yang sudah dibuat pada soal nomor 1 untuk mengerjakan soal berikut.

Program dimulai dengan memasukkan jumlah pemain, kemudian memasukkan siapa yang menjadi seeker. Selanjutnya program akan meminta input terus-menerus hingga semua pemain telah ditemukan. Input untuk membaca pemain ditemukan adalah jumlah pemain ditemukan saat ini + 1. Jika program mendapatkan input selain itu, maka program akan menganggap pemain lain belum ditemukan.

Format Input:

- Baris pertama merupakan jumlah pemain petak umpet
- Baris kedua merupakan pemain yang menjadi seeker
- N baris selanjutnya adalah angka yang mengindikasikan apakah pemain ditemukan atau tidak.
Jika pemain ditemukan input harus $x + 1$. Dimana x adalah jumlah pemain yang ditemukan saat ini. Selain input di atas, program akan menganggap pemain belum ditemukan. Program akan meminta input lagi.

Format Output:

- Setelah menerima input seeker, Program akan mengeluarkan output "Game dimulai dengan [jumlah pemain] pemain, seeker adalah [seeker]"
- Jika seeker menemukan pemain maka program akan mengeluarkan output "[n] Pemain ditemukan". Dimana n adalah jumlah pemain yang ditemukan
- Jika seeker telah menemukan semua pemain, Program akan mengeluarkan output "Semua pemain telah ditemukan, permainan selesai."

Contoh:

Input	Output	Penjelasan
5		
Ibnu	Game dimulai dengan 5 pemain, seeker adalah Ibnu	Game dimulai
5		Pemain ditemukan 0
1	1 Pemain ditemukan	Pemain ditemukan 1
2	2 Pemain ditemukan	Pemain ditemukan 2
4		Pemain ditemukan 2
3	3 Pemain ditemukan	Pemain ditemukan 3
4	4 Pemain ditemukan	Pemain ditemukan 4
	Semua pemain telah ditemukan, permainan selesai.	

Note:

- Permainan pasti dilakukan dengan minimal 2 orang
- Semua output diakhiri dengan newline
- Gunakan `scanner.nextLine()` untuk menerima input string yang diakhiri newline dan `Integer.parseInt(scanner.nextLine())` untuk menerima input integer yang diakhiri newline