Laporan Praktikum Struktur Data Pekan 6

DOSEN PENGAMPU:

Dr. Wahyudi S.T, M.T

OLEH:

Wahyu Khairi

2311531009

UNIVERSITAS ANDALAS

FAKULTAS TEKNOLOGI INFORMASI

JURUSAN INFORMATIKA

2023/2024

1. **TUJUAN**
2. Mengenal apa itu sorting algorithm dalam bahasa pemrograman java
3. Memahami langkah langkah menjalankan program menggunakan sorting algorithm
4. Memahami kapan sorting algorithm dapat digunakan dalam sebuah program
5. **KAJIAN TEORI**
6. **Sorting Algorithm**

Algoritma pengurutan mengatur elemen daftar dalam urutan tertentu. Urutan numerik dan leksikografis, baik menaik maupun menurun, adalah urutan yang paling sering digunakan. Pengurutan yang efisien sangat penting untuk mengoptimalkan efisiensi algoritma lain, seperti algoritma pencarian dan penggabungan, yang memerlukan data dalam bentuk daftar yang terurut. Selain itu, pengurutan sering digunakan untuk mengkanonisasi data dan menghasilkan output yang mudah dibaca oleh manusia.

Dua kondisi harus dipenuhi oleh output algoritma pengurutan secara formal: Input adalah permutasi (urutan ulang, tetapi tetap mempertahankan semua elemen asli) dari input, dan output berada dalam urutan monotonik (setiap elemen tidak lebih kecil atau lebih besar dari elemen sebelumnya, sesuai dengan urutan yang diinginkan).

Pengurutan adalah topik yang menarik dalam ilmu komputer, dan banyak algoritma pengurutan yang dapat digunakan. Beberapa algoritma pengurutan yang umum digunakan meliputi:

1. Shell Sort

Selain disebut sebagai metode Shell, Shellsort adalah algoritma pengurutan perbandingan yang dilakukan di tempat. Ini dapat dianggap sebagai generalisasi dari pengurutan dengan bubble atau penyisipan.

Metode Shellsort mengurutkan pasangan elemen yang sangat jauh satu sama lain sebelum secara bertahap mengurangi jarak antara elemen yang akan dibandingkan. Dengan memulai dengan elemen yang sangat jauh, algoritma ini dapat memindahkan sejumlah elemen yang tidak berada di tempatnya dengan lebih cepat daripada proses pertukaran tetangga terdekat yang sederhana.

1. Quick Sort

QuickSort adalah algoritma pengurutan yang menggunakan metode Divide and Conquer untuk memilih elemen sebagai pivot dan membagi array yang diberikan seputar pivot yang dipilih dengan menempatkan pivot pada posisi yang tepat dalam array yang terurut. Bagaimana cara kerja QuickSort? Partisi adalah proses utama dalam QuickSort. Tujuan partisi adalah untuk menempatkan pivot, atau elemen apa pun yang dapat dipilih sebagai pivot, pada posisi yang benar dalam array yang terurut, dan menempatkan semua elemen yang lebih kecil di sebelah kiri pivot dan semua elemen yang lebih besar di sebelah kanan pivot. Partisi dilakukan secara rekursif di setiap sisi pivot setelah pivot ditempatkan pada posisi yang benar, dan ini mengurutkan array pada akhirnya.

1. Merge Sort

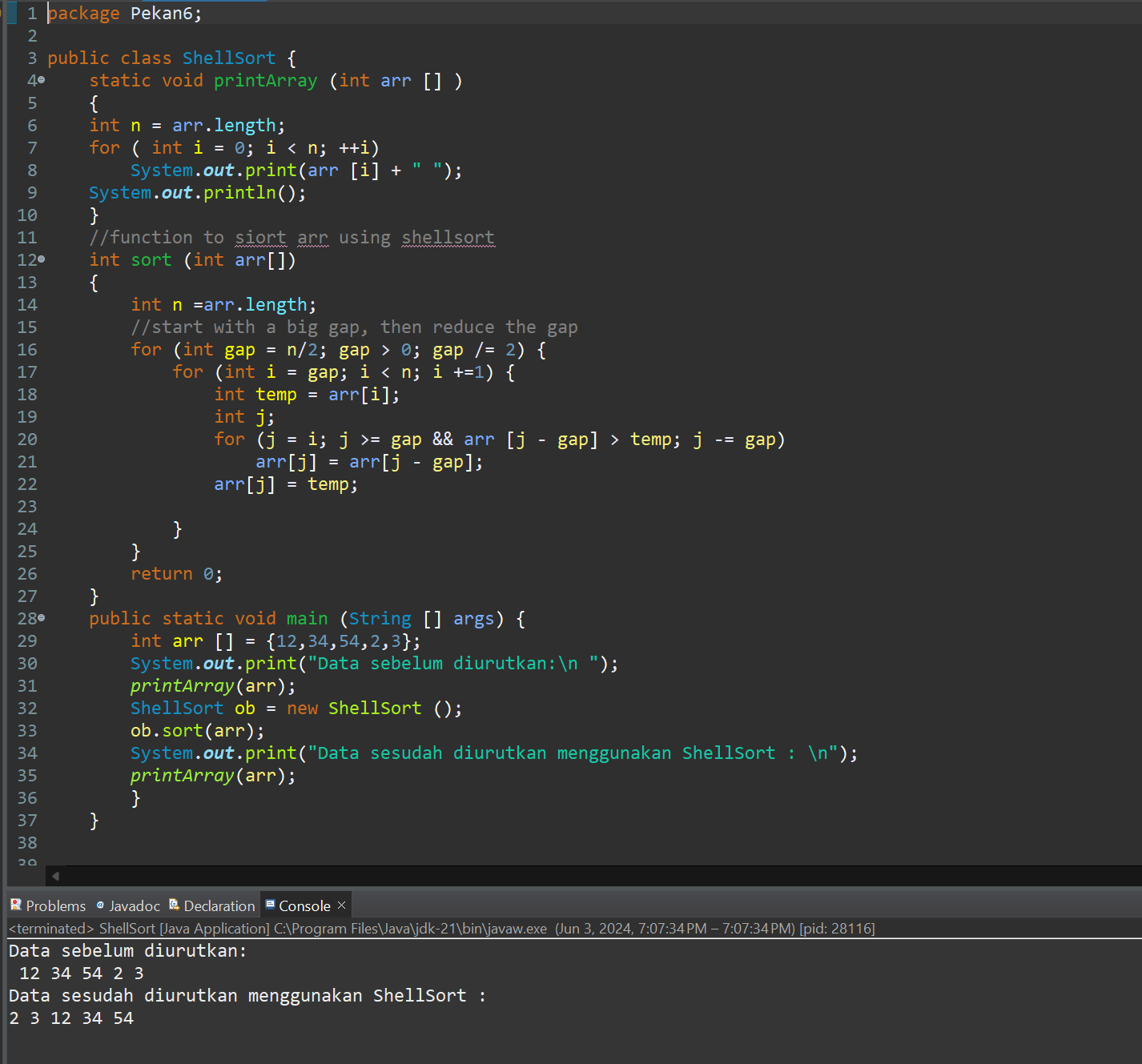
Merge Sort adalah algoritma pengurutan yang menggunakan pendekatan divide-and-conquer, yang bekerja dengan membagi array input menjadi subarray yang lebih kecil, mengurutkan setiap subarray, dan kemudian menggabungkan bagian-bagian yang sudah terurut kembali untuk mendapatkan array yang terurut.

1. **LANGKAH PENGERJAAN**
2. **shellSort**
3. Metode sort(int arr[]):

Metode ini menerima parameter berupa array integer arr. Algoritma Shell Sort diimplementasikan di dalam metode ini. Pertama, kita mengambil panjang array (n). Kemudian, kita melakukan iterasi dengan mengurangi nilai gap sebanyak setengah dari panjang array hingga gap menjadi 1. Pada setiap iterasi, kita membandingkan elemen saat ini dengan elemen yang berjarak gap dari elemen saat ini. Jika elemen yang berjarak gap lebih besar dari elemen saat ini, kita tukar posisi keduanya. Proses ini diulang hingga seluruh array terurut.

1. Metode main(String[] args):

Metode ini merupakan metode utama yang akan dijalankan ketika program dijalankan. Di dalam metode ini, kita membuat array dengan elemen [12, 34, 54, 2, 3]. Kemudian, kita memanggil metode sort(arr) untuk mengurutkan array. Terakhir, kita mencetak array yang sudah terurut.

****

1. **quickSort**
2. Metode swap(int arr[], int i, int j):

Metode ini menerima tiga parameter: array integer arr, indeks i, dan indeks j. Metode ini bertugas menukar posisi elemen pada indeks i dan j dalam array.

1. Metode partition(int arr[], int low, int high):

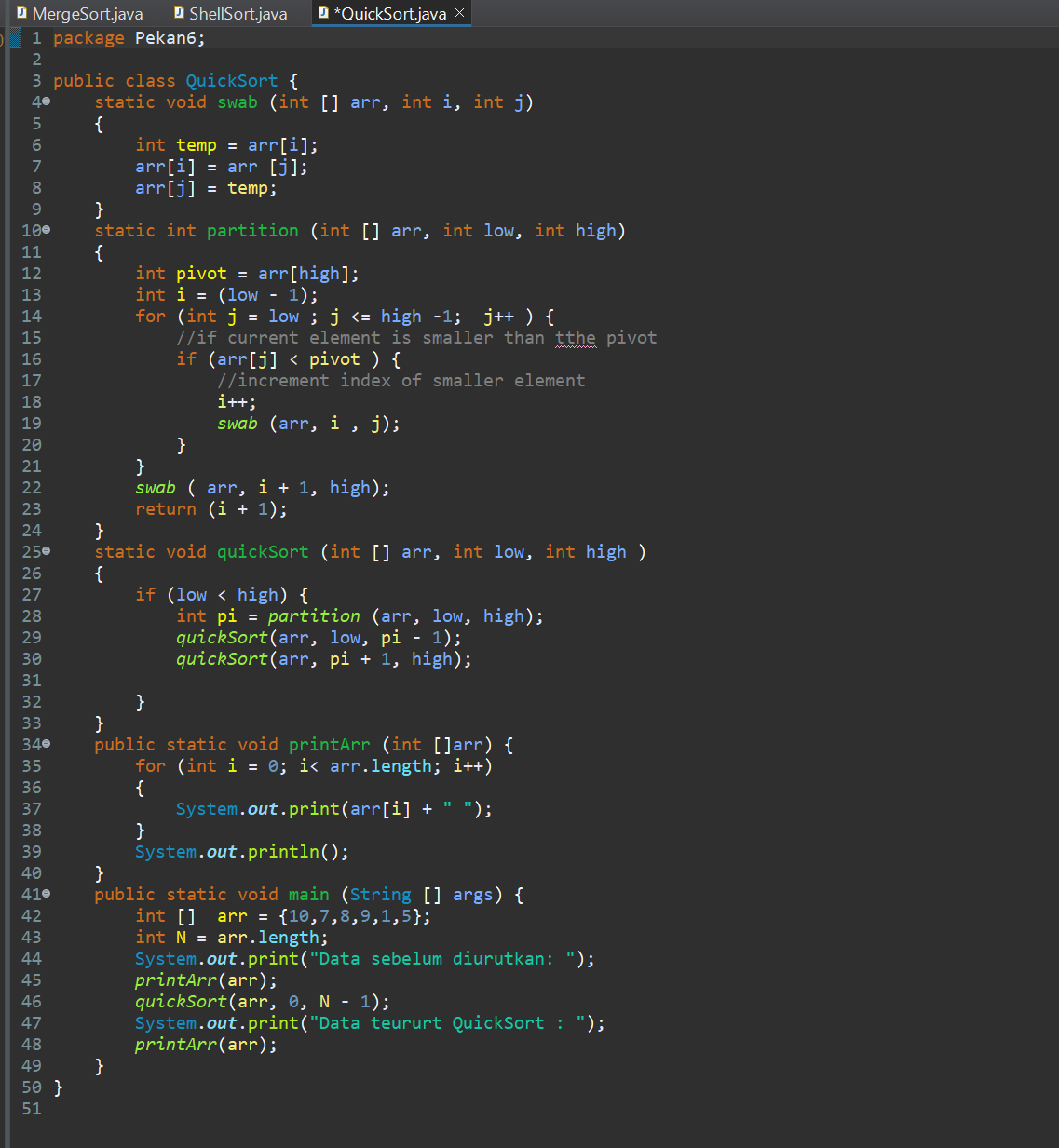
Metode ini menerima tiga parameter: array integer arr, indeks low, dan indeks high. Metode ini membagi array menjadi dua subarray berdasarkan elemen pivot. Elemen pivot dipilih dari indeks high. Elemen-elemen yang lebih kecil dari pivot ditempatkan di sebelah kiri pivot, dan elemen-elemen yang lebih besar ditempatkan di sebelah kanan pivot. Metode ini mengembalikan indeks pivot yang sudah terurut.

1. Metode quickSort(int arr[], int low, int high):

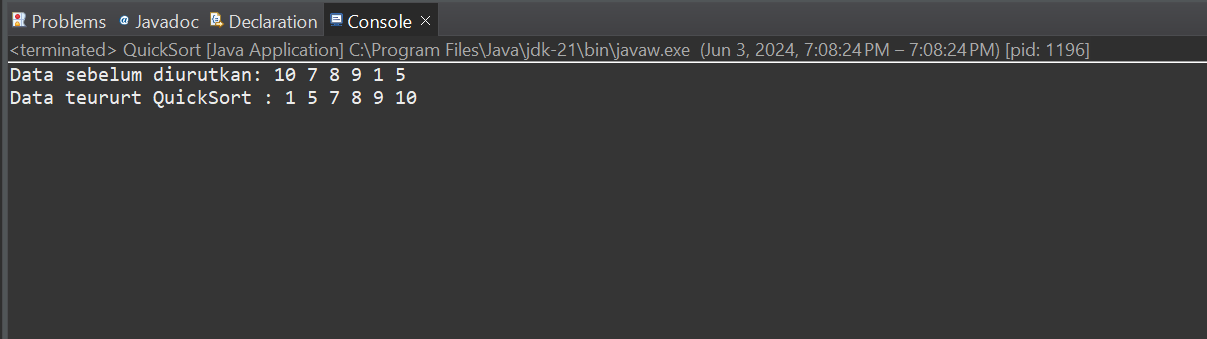
Metode ini menerima tiga parameter: array integer arr, indeks low, dan indeks high. Metode ini secara rekursif membagi dan mengurutkan array menggunakan algoritma QuickSort. Base case terjadi ketika low kurang dari high.

1. Metode main(String[] args):

Metode ini merupakan metode utama yang akan dijalankan ketika program dijalankan. Di dalam metode ini, kita membuat array dengan elemen [10, 7, 8, 9, 1, 5]. Kemudian, kita memanggil metode quickSort(arr, 0, n - 1) untuk mengurutkan array. Terakhir, kita mencetak array yang sudah terurut.

****

**OUTPUT :**

****

1. **mergeSort**
2. Metode merge(int arr[], int l, int m, int r):

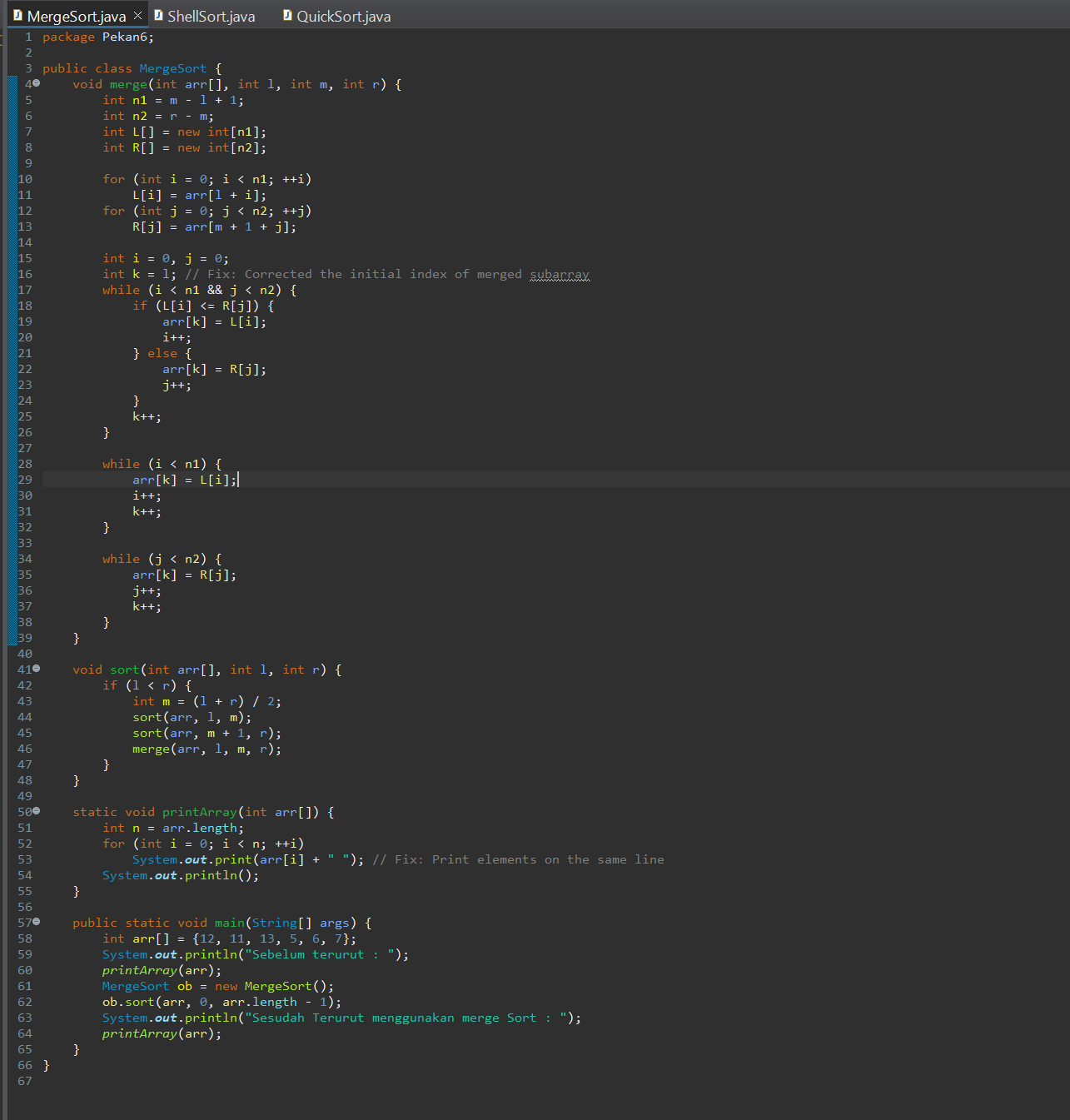
Metode ini menerima empat parameter: array integer arr, indeks l, indeks m, dan indeks r. Metode ini menggabungkan dua subarray terurut (dari indeks l hingga m dan dari indeks m+1 hingga r) menjadi satu array terurut. Proses penggabungan dilakukan dengan membandingkan elemen-elemen dari kedua subarray dan menempatkannya dalam urutan yang benar.

1. Metode sort(int arr[], int l, int r):

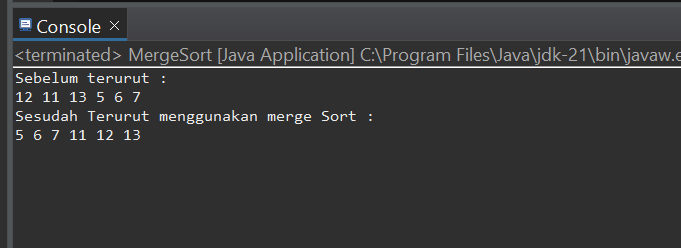
Metode ini menerima tiga parameter: array integer arr, indeks l, dan indeks r. Metode ini secara rekursif membagi dan mengurutkan array menggunakan algoritma Merge Sort. Base case terjadi ketika l kurang dari r.

1. Metode main(String[] args):

Metode ini merupakan metode utama yang akan dijalankan ketika program dijalankan. Di dalam metode ini, kita membuat array dengan elemen [12, 11, 13, 5, 6, 7]. Kemudian, kita memanggil metode sort(arr, 0, arr.length - 1) untuk mengurutkan array. Terakhir, kita mencetak array yang sudah terurut.

****

**OUTPUT :**

****