1. `def rearrange\_even\_odd(arr):`: Ini adalah deklarasi fungsi yang bernama `rearrange\_even\_odd`. Fungsi ini akan menerima satu parameter, yaitu list `arr`, yang akan diubah urutannya.

2. `if len(arr) <= 1:`: Ini adalah kondisi pertama di dalam fungsi. Jika panjang list `arr` kurang dari atau sama dengan 1 (artinya hanya terdapat 1 elemen atau tidak ada elemen), maka fungsi akan mengembalikan list tersebut tanpa mengubah urutannya, karena tidak ada yang perlu diatur ulang.

3. `first\_element = arr[0]`: Fungsi ini mengambil elemen pertama dari list `arr` dan menyimpannya dalam variabel `first\_element`.

4. `rest\_elements = arr[1:]`: Fungsi ini mengambil semua elemen selain elemen pertama dari list `arr` dan menyimpannya dalam variabel `rest\_elements`.

5. `if first\_element % 2 == 0:`: Ini adalah kondisi kedua di dalam fungsi. Fungsi ini memeriksa apakah `first\_element` adalah bilangan genap dengan memeriksa apakah sisa bagi (`%`) dari `first\_element` dengan 2 adalah 0. Jika ya, maka `first\_element` adalah bilangan genap, dan fungsi akan mengembalikan list yang berisi `first\_element` diikuti oleh hasil dari pemanggilan rekursif dari `rearrange\_even\_odd(rest\_elements)`. Ini berarti `first\_element` akan berada di depan elemen-elemen ganjil.

6. `else:`: Jika `first\_element` bukan bilangan genap (artinya ganjil), maka fungsi akan mengembalikan hasil dari pemanggilan rekursif dari `rearrange\_even\_odd(rest\_elements)` diikuti oleh `[first\_element]`. Ini berarti elemen pertama akan tetap di belakang, setelah elemen-elemen genap.

Selanjutnya, program menggunakan fungsi ini untuk mengatur ulang list `input\_list`:

- `input\_list = [3, 1, 9, 2, 8, 5, 4, 7, 6, 10]`: Ini adalah list input awal yang berisi campuran bilangan ganjil dan genap.

- `result = rearrange\_even\_odd(input\_list)`: Fungsi `rearrange\_even\_odd` dipanggil dengan `input\_list` sebagai argumen, dan hasilnya disimpan dalam variabel `result`.

- `print("Urutan setelah diatur ulang:", result)`: Hasil yang telah diatur ulang disimpan dalam `result`, dan program mencetak hasilnya dengan pesan "Urutan setelah diatur ulang:". Hasilnya adalah list `input\_list` yang elemen-elemen genap berada di depan elemen-elemen ganjil sesuai dengan urutan yang telah dijelaskan dalam fungsi `rearrange\_even\_odd`.