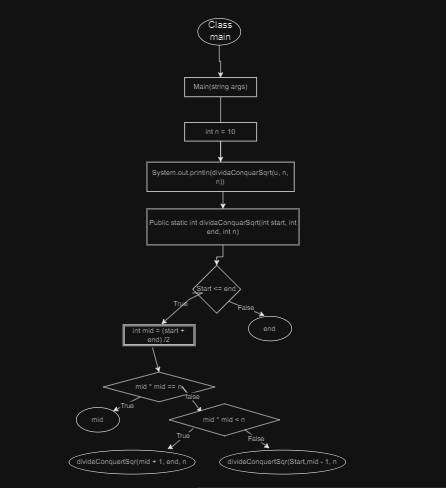
Latihan :

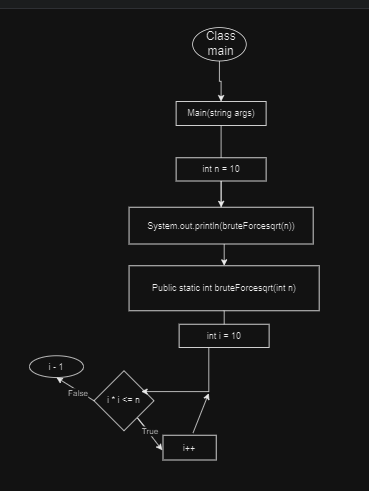
1. Buatlah flowchart untuk menghitung nilai akar dari suatu bilangan dengan algoritma Brute Force dan Divide Conquer! *Jika bilangan tersebut bukan merupakan kuadrat sempurna, bulatkan angka ke bawah.*

Divide :



Start - input bilangan n - apakah n >= 0? - ya- Output divideConquerSqrt(0, n, n) Output "Invalid input" - end - tidak-end

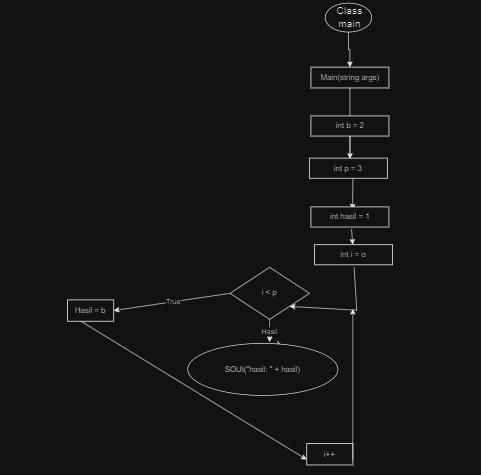
**Brute force:**

****

Start - input bilangan n - inisalisasi variable I = 0 - apakah i8i <= n? - ya I = i+1-kembali ke “apakah i\*I <= n? - tidak-output i-1

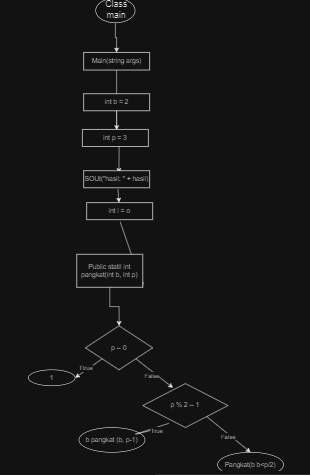
1. Buatlah flowchart untuk menghitung hasil pangkat dari inputan suatu bilangan dengan algoritma Brute Force dan Divide Conquer

Brute :



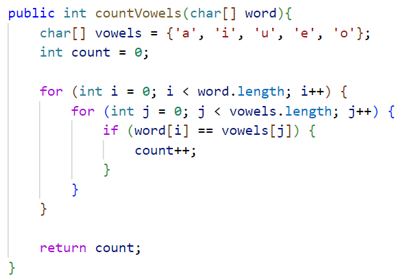
Start V Input: basis (b), pangkat (p) V Inisialisasi hasil = 1 V Ulangi p kali: hasil = hasil \* b V Output: hasil End

Divide :



Start V Input: basis (b), pangkat (p) V Jika p = 0, return 1 V Jika p ganjil, return b \* pangkat(b, p-1) V Jika p genap, return pangkat(b\*b, p/2) V Output: hasil End

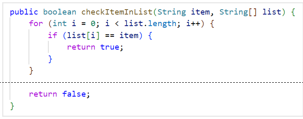
1. Tentukan notasi Big O yang sesuai dari kode program berikut!



Jawab :

Notasi Big O dari kode program tersebut adalah O(n), di mana n adalah panjang array `word`. Hal ini disebabkan oleh iterasi melalui array `word` yang dilakukan sebanyak `word.length` kali. Iterasi melalui array `vowels` tidak mempengaruhi kompleksitas secara signifikan karena jumlah elemennya tetap dan relatif kecil (5 elemen).

1. Tentukan notasi Big O yang sesuai dari kode program berikut!



Jawab :

Notasi Big O dari kode program di atas adalah O(n), di mana n adalah panjang array `list`. Hal ini disebabkan oleh iterasi melalui array `list` dalam loop `for`.

Di dalam loop, satu perbandingan dilakukan untuk memeriksa kesamaan antara `list[i]` dan `item`, menghasilkan operasi konstan.