

# Latihan Soal Algo CS UAS

## 1. Recursion: Ganjil Genap

Buatlah sebuah fungsi rekursif yang menerima input bilangan bulat N. Fungsi tersebut harus bekerja dengan aturan sebagai berikut:

- Jika N adalah bilangan **ganjil**, maka hasilnya adalah N ditambah dengan hasil fungsi rekursif dari (N-1).
- Jika N adalah bilangan **genap** (dan bukan 0), maka hasilnya adalah hasil fungsi rekursif dari (N/2).
- Jika  $N \leq 0$ , maka kembalikan nilai 0.

Tugas Anda adalah mencetak hasil akhir dari pemanggilan fungsi tersebut terhadap input yang diberikan.

### Format Input

Satu baris berisi sebuah bilangan bulat N.

### Format Output

Sebuah bilangan bulat hasil perhitungan rekursif.

### Constraints

- $0 \leq N \leq 100$

### Sample Input

7

### Sample Output

11

(Penjelasan: 7 ganjil  $\rightarrow 7 + f(6)$ . 6 genap  $\rightarrow f(3)$ . 3 ganjil  $\rightarrow 3 + f(2)$ . 2 genap  $\rightarrow f(1)$ . 1 ganjil  $\rightarrow 1 + f(0)$ . Total:  $7 + 3 + 1 = 11$ )

## 2. Sorting: Mengurutkan Usia

Anda diberikan data sekelompok orang, di mana setiap orang memiliki Nama dan Umur. Anda diminta untuk mengimplementasikan struct (atau class) untuk menampung data tersebut, kemudian mengurutkan data berdasarkan aturan berikut:

1. Berdasarkan **Umur** secara menaik (*ascending*).
2. Jika Umur sama, urutkan berdasarkan **Nama** secara abjad (*ascending / leksikografis*).

### Format Input

Baris pertama berisi bilangan bulat N, menyatakan jumlah orang.

N baris berikutnya masing-masing berisi string Nama dan bilangan bulat Umur yang dipisahkan spasi.

### Format Output

N baris data yang telah diurutkan, dengan format: Nama Umur.

### Constraints

- $1 \leq N \leq 1.000$
- Panjang nama 1 - 20 karakter.

### Sample Input

```
3
Caca 20
Budi 22
Andi 20
```

### Sample Output

```
Andi 20
Caca 20
Budi 22
```

### 3. File Processing: Penjumlahan Terfilter

Diberikan sebuah file teks bernama angka.txt yang berisi sekumpulan bilangan bulat yang dipisahkan oleh spasi. Tugas Anda adalah membuat program untuk:

1. Membaca seluruh angka dari file tersebut.
2. Menghitung jumlah total elemen (K) yang ada di dalam file.
3. Menjumlahkan hanya angka-angka yang **habis dibagi oleh K**.

#### Format Input (Isi file angka.txt)

File berisi angka-angka integer. Polanya adalah: Angka pertama N, diikuti N angka. Kemudian angka M, diikuti M angka, dan seterusnya sampai akhir file (EOF).

#### Format Output

Cetak hasil penjumlahan untuk setiap kasus di baris baru.

#### Constraints

- Banyaknya elemen dalam file  $\leq 100$ .
- Nilai elemen  $\leq 1.000$ .

#### Sample Input (Isi file angka.txt)

```
3
1 5 7
4
4 6 2 6
5
10 12 5 20 5
```

#### Sample Output

```
0
4
40
```

#### Penjelasan Sample:

1. **Kasus 1:** N=3. Angka: 1, 5, 7. Tidak ada yang habis dibagi 3. Total = 0.
2. **Kasus 2:** N=4. Angka: 4, 6, 2, 6. Hanya angka **4** yang habis dibagi 4. Total = 4.
3. **Kasus 3:** N=5. Angka: 10, 12, 5, 20, 5. Yang habis dibagi 5 adalah 10, 5, 20, 5. Total =  $10+5+20+5 = 40$ .

#### 4. Binary Search: Cari Posisi

Diberikan sebuah array A yang berisi N bilangan bulat yang sudah terurut secara menaik (sorted ascending). Anda diminta mencari keberadaan sebuah angka target T menggunakan algoritma Binary Search.

Jika angka ditemukan, cetak posisi indeksnya dengan format tertentu. Jika tidak ditemukan, cetak "Data tidak ditemukan".

##### Format Input

Baris pertama berisi bilangan bulat N (jumlah elemen).

Baris kedua berisi N bilangan bulat yang terurut.

Baris ketiga berisi bilangan bulat T (target yang dicari).

##### Format Output

Jika ditemukan, cetak dengan format: "%d - [%d]" (Angka Target, Indeks). Indeks dimulai dari 0.

Jika tidak ditemukan, cetak string biasa.

##### Constraints

- $1 \leq N \leq 100.000$
- Elemen array unik (tidak ada duplikat).

##### Sample Input

```
7
2 5 8 12 19 24 30
19
```

##### Sample Output

```
19 - [4]
```

---