

## 1. QUESTION 1

### Nama file: penduduk.py

Buatlah program yang digunakan untuk membaca masukan jumlah penduduk desa dari sebuah kecamatan di suatu kota. Program pertama-tama akan membaca input N yang merupakan jumlah desa dalam kecamatan. N diasumsikan selalu valid. Selanjutnya, program akan meminta input jumlah penduduk setiap desa sebanyak N (input jumlah penduduk desa dianggap selalu valid, yaitu  $> 0$ ) dan menghitung jumlah penduduk kecamatan (total penduduk desa).

### Contoh input/output:

Input	Output	Keterangan
3 3000 5000 1000	9000	N = 3
5 7000 5000 4000 1000 3000	20000	N = 5

## 2. QUESTION 2

### Nama file: nilaiekstrim.py

Buatlah sebuah program yang digunakan untuk membaca sebuah integer N (asumsikan  $0 < N \leq 100$ ).

Program kemudian membaca N buah integer dan menyimpan setiap integer ke dalam suatu array. Selanjutnya, program menerima masukan sebuah nilai integer, misalnya X, dan menampilkan:

- Jika X ada di array, apakah X adalah nilai maksimum (tuliskan "maksimum") atau nilai minimum (tuliskan "minimum") atau keduanya. Jika bukan nilai maksimum atau minimum, menuliskan "N#A".
- Jika X tidak ada di array, tuliskan "X tidak ada".

### Contoh input/output:

Input	Output	Keterangan
6 1 -1 2 4 -1 2 -1	minimum	N = 6 Isi array = {1,-1,2,4,-1,2} X = -1 maksimum = 4 minimum = -1
5 3 2 2 4 2 3	N#A	N = 5 Isi array = {3,2,2,4,2} X = 3 maksimum = 4 minimum = 2
5 1 1 1 1 1 1	maksimum minimum	N = 5 Isi array = {1,1,1,1,1} X = 1 maksimum = 1 minimum = 1
6 1 -1 2 4 -1 2 8	8 tidak ada	N = 6 Isi array = {1,-1,2,4,-1,2} X = 8 X tidak ada dalam array

### 3. QUESTION 3

Nama file: bujursangkar.py

Buatlah program yang digunakan untuk menghitung luas bujur sangkar (persegi). Program menerima input sisi (dalam bentuk integer). Input sisi harus divalidasi sehingga selalu  $> 0$ . Apabila sisi yang dimasukkan  $\leq 0$ , program akan mengeluarkan pesan error. Setelah sisi yang dimasukkan valid, program menghasilkan luas sisi dengan rumus: sisi \* sisi

**Contoh input/output:**

Input	Output
10	100
-1	Sisi harus $> 0$

#### 4. QUESTION 4

**Nama file: caricharacter.py**

Buatlah sebuah program yang digunakan untuk membaca sebuah integer positif N. Nilai N harus divalidasi sampai didapatkan nilai N yang benar, yaitu  $0 < N \leq 100$ . Jika masukan N salah, tuliskan pesan "Masukan salah. Ulangi!". Setelah didapatkan nilai N yang benar, program meminta masukan N buah character.

Selanjutnya, program menerima masukan sebuah character, misalnya CC, dan menghasilkan:

- Jika CC = 'S' atau CC = 's' (kependekan dari "small"), tuliskan urutan pertama ditemukannya character huruf kecil dan tuliskan hurufnya. Jika tidak ada character huruf kecil, tuliskan "Tidak ada huruf kecil".
- Jika CC = 'L' atau CC = 'l' (el, kependekan dari "large"), tuliskan urutan pertama ditemukannya character huruf besar dan tuliskan hurufnya. Jika tidak ada character huruf besar, tuliskan "Tidak ada huruf besar".
- Jika CC = 'X' atau CC = 'x', tuliskan urutan pertama ditemukannya character selain huruf dan tuliskan character-nya. Jika tidak ada character selain huruf, tuliskan "Semua huruf".
- Jika CC selain selain {'S','s','L','l','X','x'} tuliskan: "Tidak diproses".

**Petunjuk:**

- Simpan nilai-nilai character dalam array of character.
- Padanan standar ASCII untuk character huruf kecil 'a' s.d. 'z' adalah angka desimal 97 s.d. 122.
- ☐ ☐ ☐ Padanan standar ASCII untuk character huruf besar 'A' s.d. 'Z' adalah angka desimal 65 s.d. 90.

**Contoh input/output:**

Interaksi Input	Output	Keterangan
<u>-1</u> Masukan salah. Ulangi! <u>8</u> <u>A</u> <u>B</u> <u>a</u> <u>C</u> <u>%</u> <u>#</u> <u>7</u> <u>*</u> <u>-</u> <u>S</u>	3 a	N = 8 (divalidasi 1 kali) CC = 'S' Character huruf kecil pertama muncul di urutan ke-3 yaitu 'a'
<u>3</u> <u>A</u> <u>B</u> <u>C</u> <u>L</u>	1 A	N = 3 CC = 'L' Character huruf besar pertama muncul di urutan pertama yaitu 'A'
<u>3</u> <u>a</u> <u>#</u> <u>\$</u> <u>L</u>	Tidak ada huruf besar	N = 3 CC = 'L' Tidak ada huruf besar
<u>3</u> <u>A</u> <u>4</u> <u>\$</u> <u>3</u>	Tidak diproses	N = 3 X = 99 Tidak ada yang diproses

## 5. QUESTION 5

### Nama file: berat.py

Buatlah program yang digunakan untuk membaca masukan berat tubuh mahasiswa (dalam bentuk angka) dalam suatu kelas, sampai pengguna mengetikkan -999 (-999 tidak termasuk nilai yang diproses). Harus diperiksa bahwa nilai yang dimasukkan berada dalam range 30-200 (30 dan 200 termasuk). Jika ada data nilai yang salah, maka data tersebut diabaikan (tidak termasuk dalam pemrosesan).

Selanjutnya program menuliskan ada berapa banyak mahasiswa keseluruhan, berapa banyak mahasiswa dengan berat  $\leq 50$ , berapa banyak mahasiswa dengan berat  $\geq 100$ , dan berapa rata-rata berat mahasiswa. Berat rata-rata dituliskan tanpa koma (dibulatkan, gunakan fungsi **round**).

Semua data nilai yang salah diabaikan (tidak termasuk yang diproses). Jika pengguna dari awal mengetikkan -999 (artinya tidak ada data yang dimasukkan) atau tidak ada data yang valid, tuliskan pesan: "Data kosong".

### Contoh input/output:

Input	Output	Keterangan
50 290 100 -90 -999	2 1 1 75	2 data berat mahasiswa salah, yaitu 290 dan -90 diabaikan  Banyak mahasiswa = 2  Berat $\leq 50 = 1$  Berat $\geq 100 = 1$  Rata-rata = 75
150 40 60 -999	3 1 1 83	Banyak mahasiswa = 3  Berat $\leq 50 = 1$  Berat $\geq 100 = 1$  Rata-rata = 83
-999	Data kosong	Tidak ada data
-90 -999	Data kosong	Tidak ada data valid