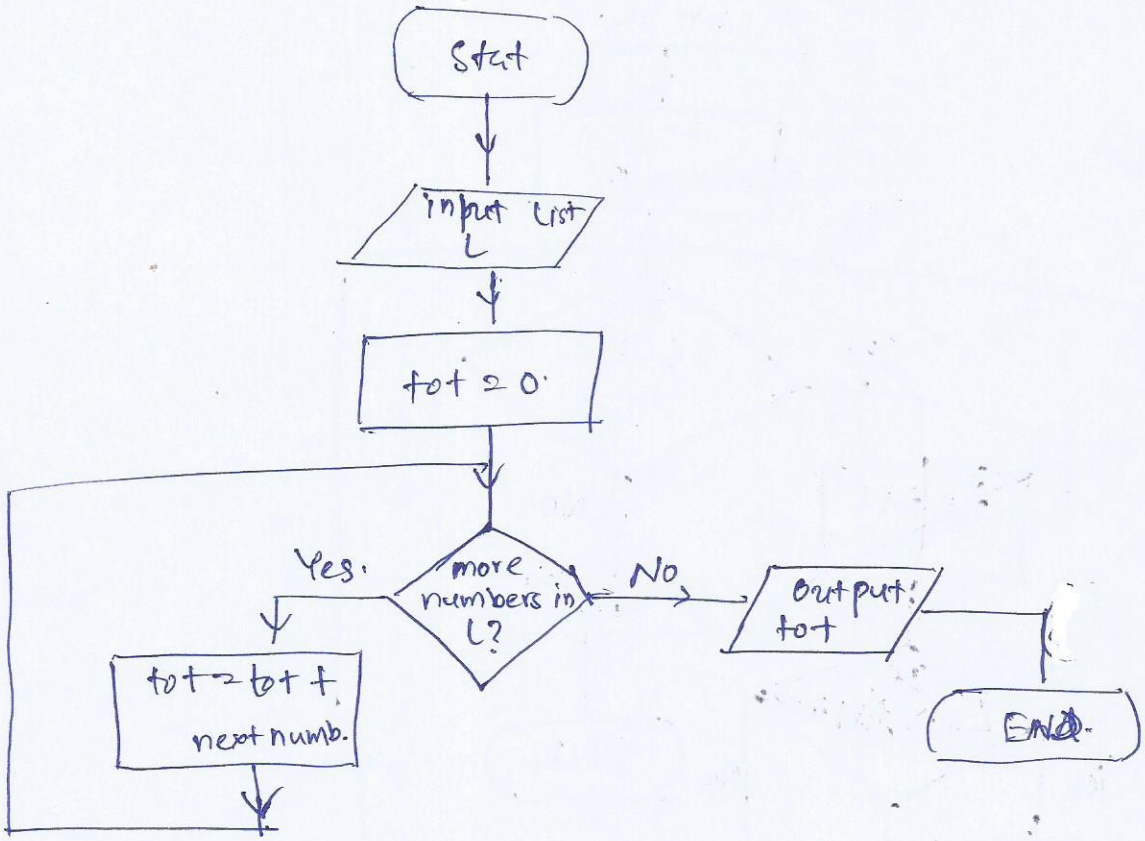
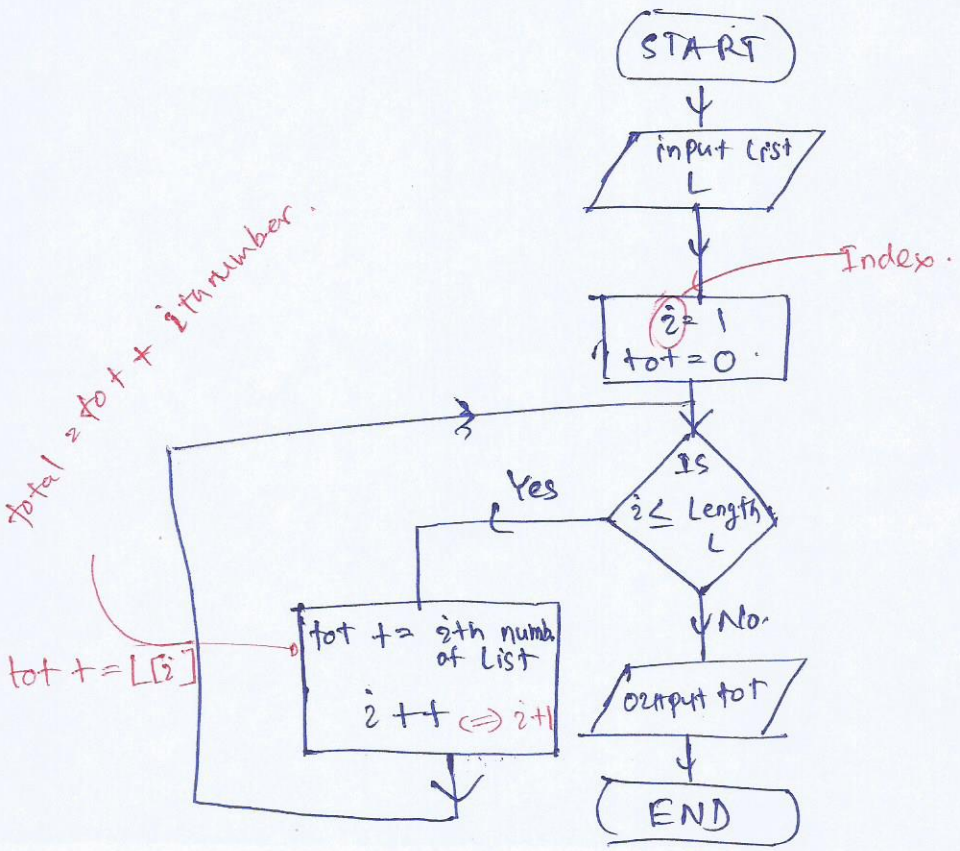
Example

more human readable. :-



more programamable :-



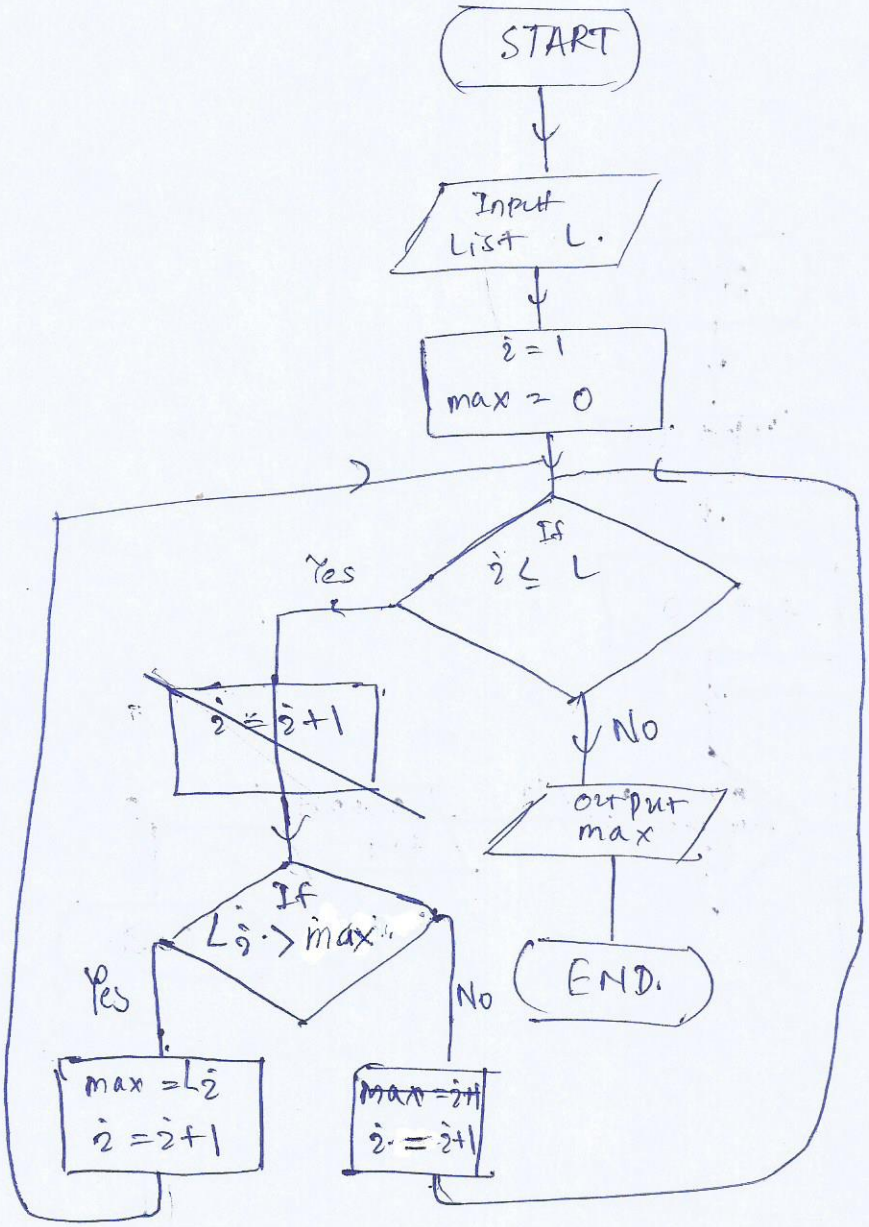
1st elem. 2nd elem 3rd elem

L = 2, 3, 5, 7, 8, 9

Length of L = 6

i	tot
1	0
2	2
3	5
4	10
5	17
6	25
7	34
8	END

Exm:- Max^m.



L = 2, 7, 3, 6, 9, 5 L = 6.

i	max
1	0
2	2
3	7
4	7
5	7
6	9.
7	9

END.

Pseudo Code for example -(1)

```
Calculate_total ( List L).
{
i = 1
tot = 0
Cond : if  (i > Length of List)
            go to Done.
       tot = tot + L[i]
       i = i + 1
       go to Cond:

Done: Print tot
}.
```

Done: Print tot

Simply writing using for loop :-

```
Calculate_total (List L) {
    tot = 0
    for (i = 1 ; i ≤ Length of L ; i = i + 1) {
                (1)      (2), (5), ...      (4), (7), ...
        tot = tot + L[i]
                    (3), (6), ...
    }
    Print tot.
}.
```

Or

```
for(i = Length of L ; i > 0 ; i--).
                          or
                        i = i - 1
```

Using while Loop :-

```
Calculate_total (list L) {
    tot = 0
    i = 1
    while (i ≤ length of L) {
        tot = tot + L[i]
        i++
    }
Print tot.
}.
```

Pseudo Code for example – (2)

Using for Loop:

```
find_max (list L) {
    max = 0.
    for (i = 1 ; i ≤ length of L ; i++) {
        if (L[i] > max ) {
            max = L[i].
    }
}
Print max.
}.
```

or

```
find_max (list L) {
    max = 0.
    for (i = 1; i ≤ Lenght of L; i++){
        max = return_fmax (max, L[i]).
```

```
return_max (num a, num b,
if a > b
return a
else
    return b
```

Lecture ①

Write pseudocode of a programme to,

(i) Inputs : a list of numbers & a single number.
L k

Output : If the number k is in the list L
return "TRUE"
otherwise "FALSE"

(ii) Inputs : a list of #s L, & a single number k.

Output : Print all the numbers in L which are greater than k.