* Function with argument with Return type.

Algorithm

step1: staut

step2: input   a = 3 ; b = 6

step3: sum = add (a,b)

step4: Display sum

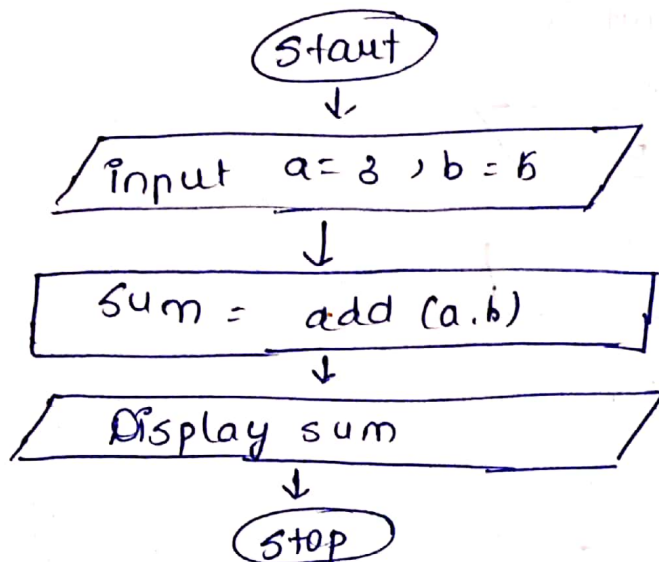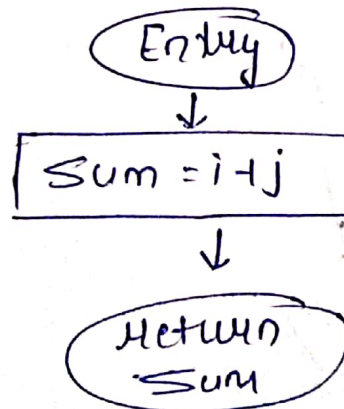step5: stop.

aad (int i, int j)

step1: entuy

step2: sum = i + j

step3: Return sum

Flow chaut

add (int i, int j)

Functions with no arguments and no return type.

Algorithm:
step1: start
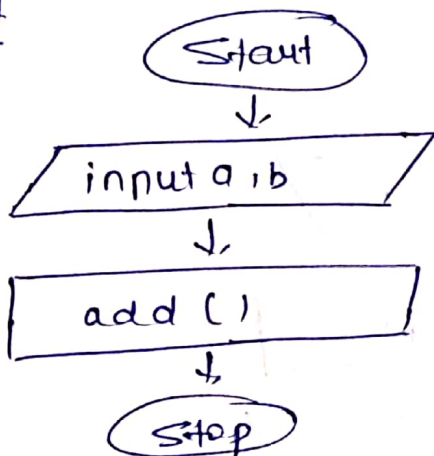step2: input a,b
step3: add ()
step4: stop.

add ()
step1: Entry
step2: Input i=3; j=6
step3: sum i+j
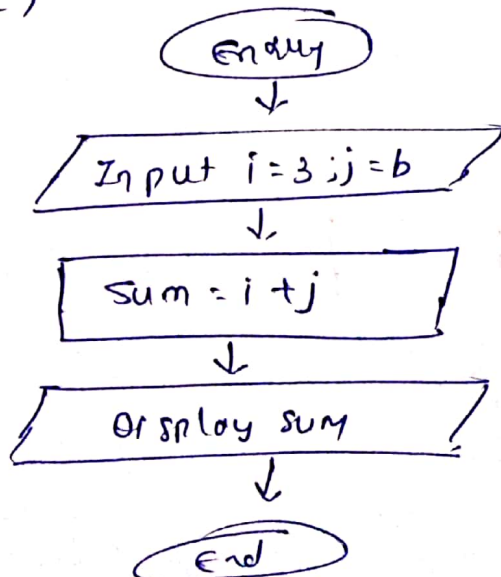step4: Display sum
step5: End.

flowchart

```
        ( Start )
            ↓
     / input a,b /
            ↓
     [  add ()  ]
            ↓
        ( Stop )
```

add ()

```
        ( Entry )
            ↓
   / Input i=3; j=b /
            ↓
     [  Sum = i+j  ]
            ↓
   / Display sum /
            ↓
        ( End )
```

* Functions with no arguments, with return type.

Algorithm

Step 1 : Start
Step 2 : sum = add()
Step 3 : Display sum
Step 4 : Stop.

add()

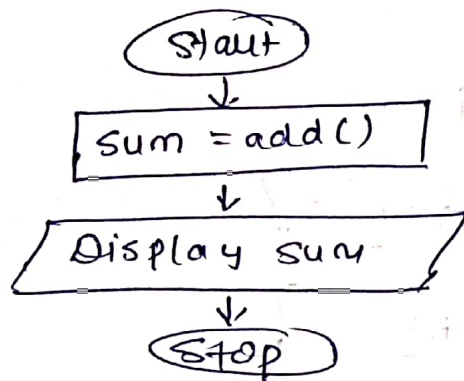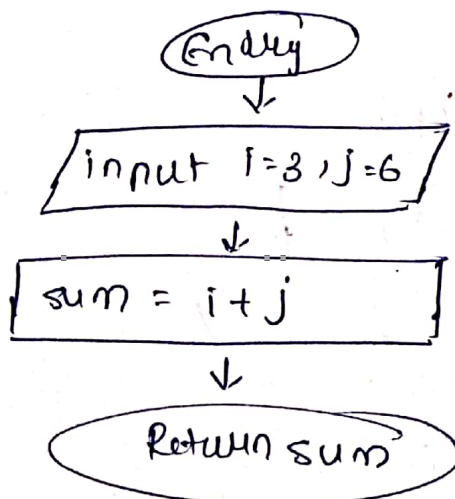Step 1 : Entry
Step 2 : Input i = 3; j = 6
Step 3 : sum = i + j
Step 4 : return sum

Flowchart

```
        ( Start )
           ↓
   | sum = add() |
           ↓
  / Display sum /
           ↓
        ( Stop )
```

add()

```
        ( Entry )
           ↓
  / input i = 3, j = 6 /
           ↓
   | sum = i + j |
           ↓
   ( Return sum )
```

* Functions with argument no return type.

Algorithm

Step1: Start
Step2: Input a=3; b=6
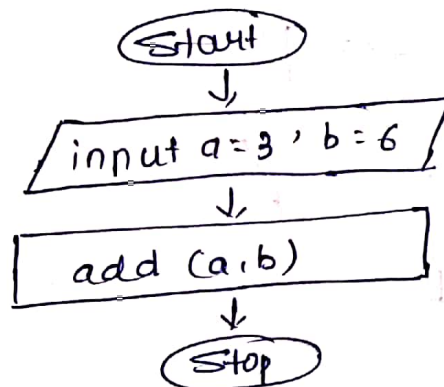Step3: add (a,b)
Step4: stop.

add (int i, int j)
Step1: Entry
Step2: sum = i+j
Step3: Display sum
Step4: End.

Flowchart

```
        ( Start )
            ↓
  / input a=3, b=6 /
            ↓
  |   add (a,b)   |
            ↓
        ( Stop )
```

add (int i, int j)

```
        ( Entry )
            ↓
  |   sum = i+j   |
            ↓
  / Display  sum /
            ↓
        ( End )
```