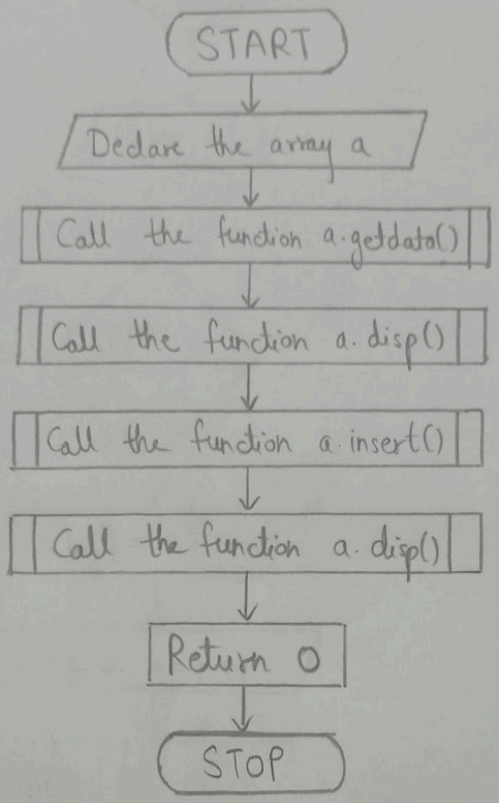
a[j] = a[j-1]

STEP ② - Declare n = n+1

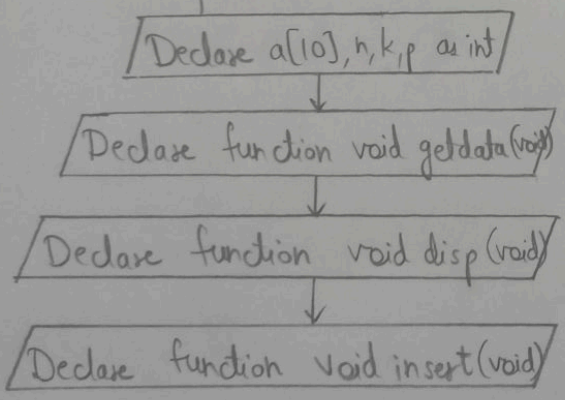
STEP ③ - Declare a[k-1] = p.

Flowchart

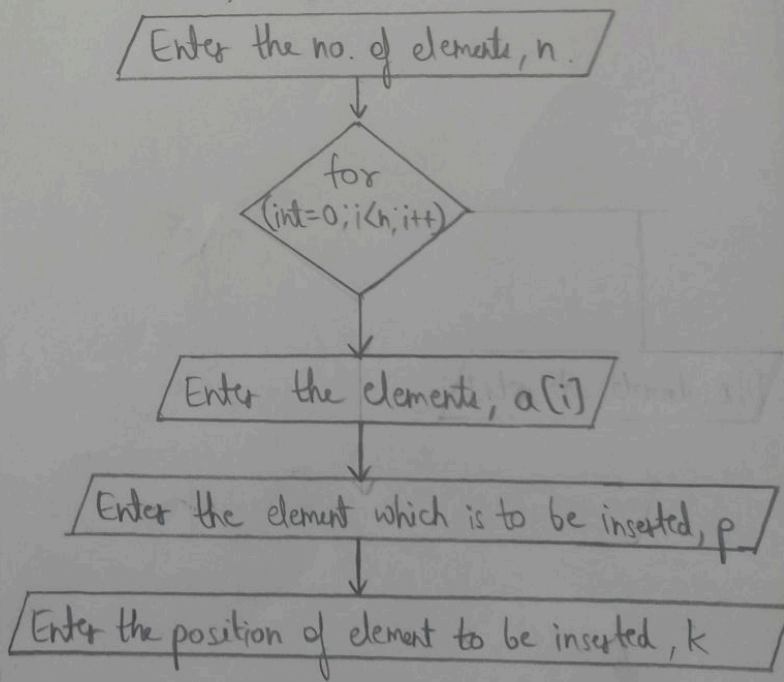


1

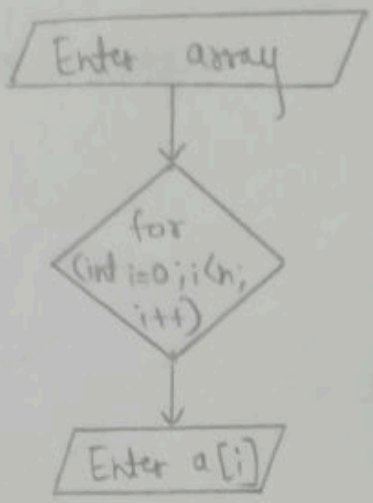
Class array



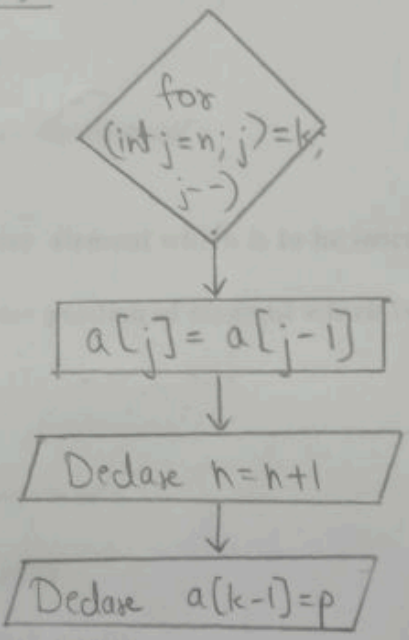
Void array :: getdata (void)



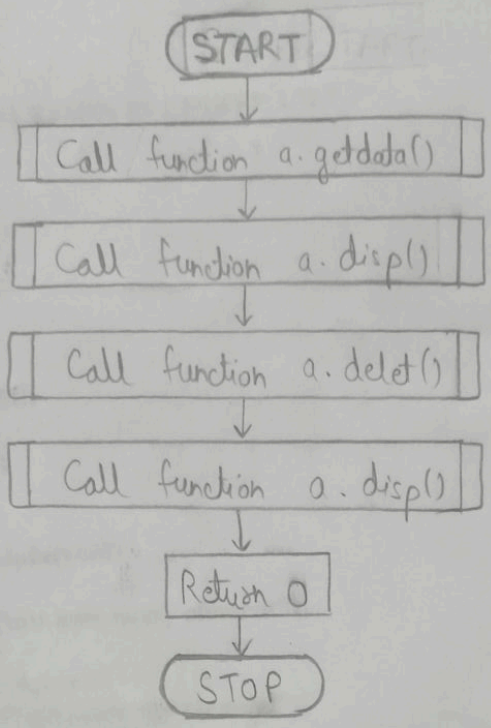
Void array :: disp(void)



Void array :: insert(void)

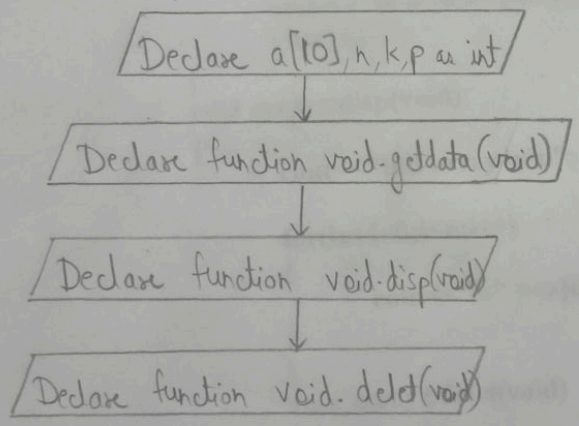


Flowchart

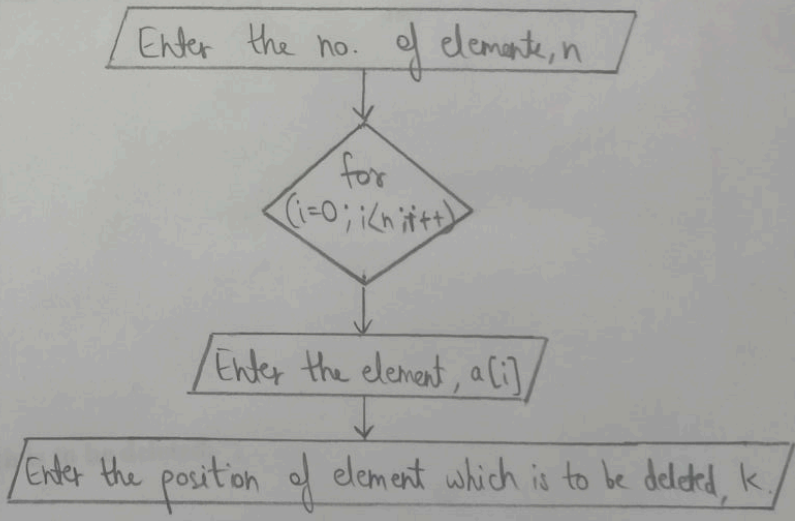


2

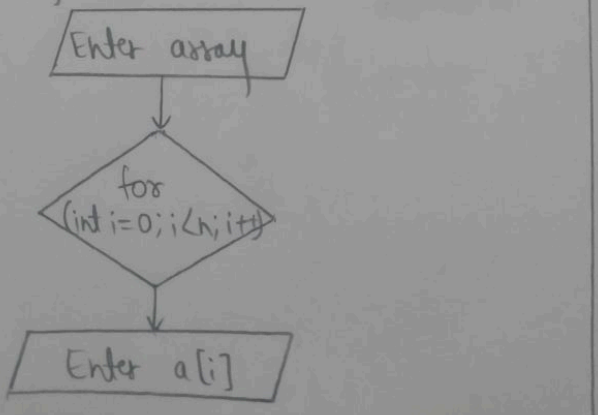
Class array



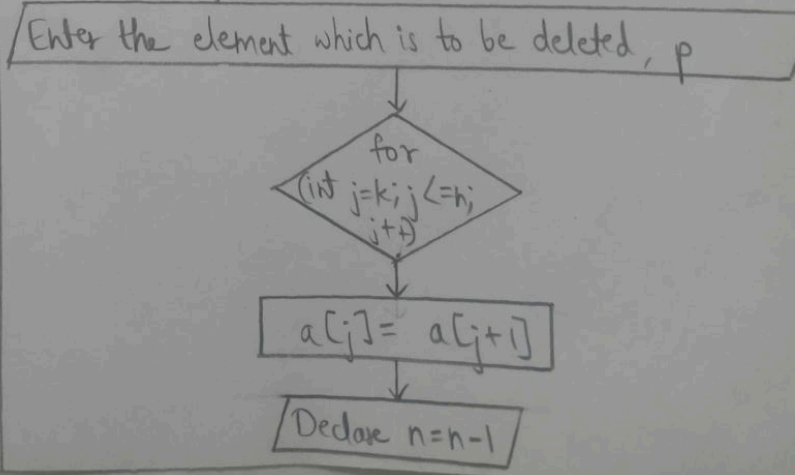
void array :: getdata (void)



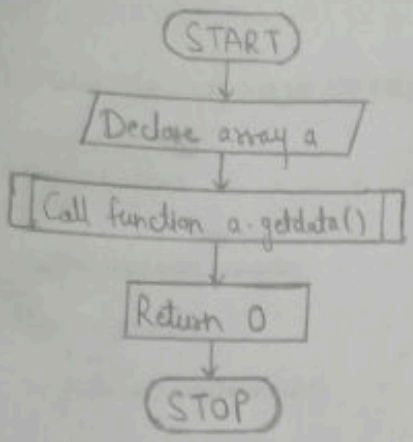
void array :: disp(void)



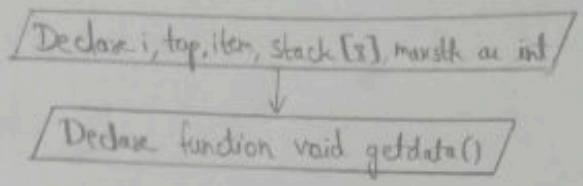
void array :: delet (void)



Flowchart

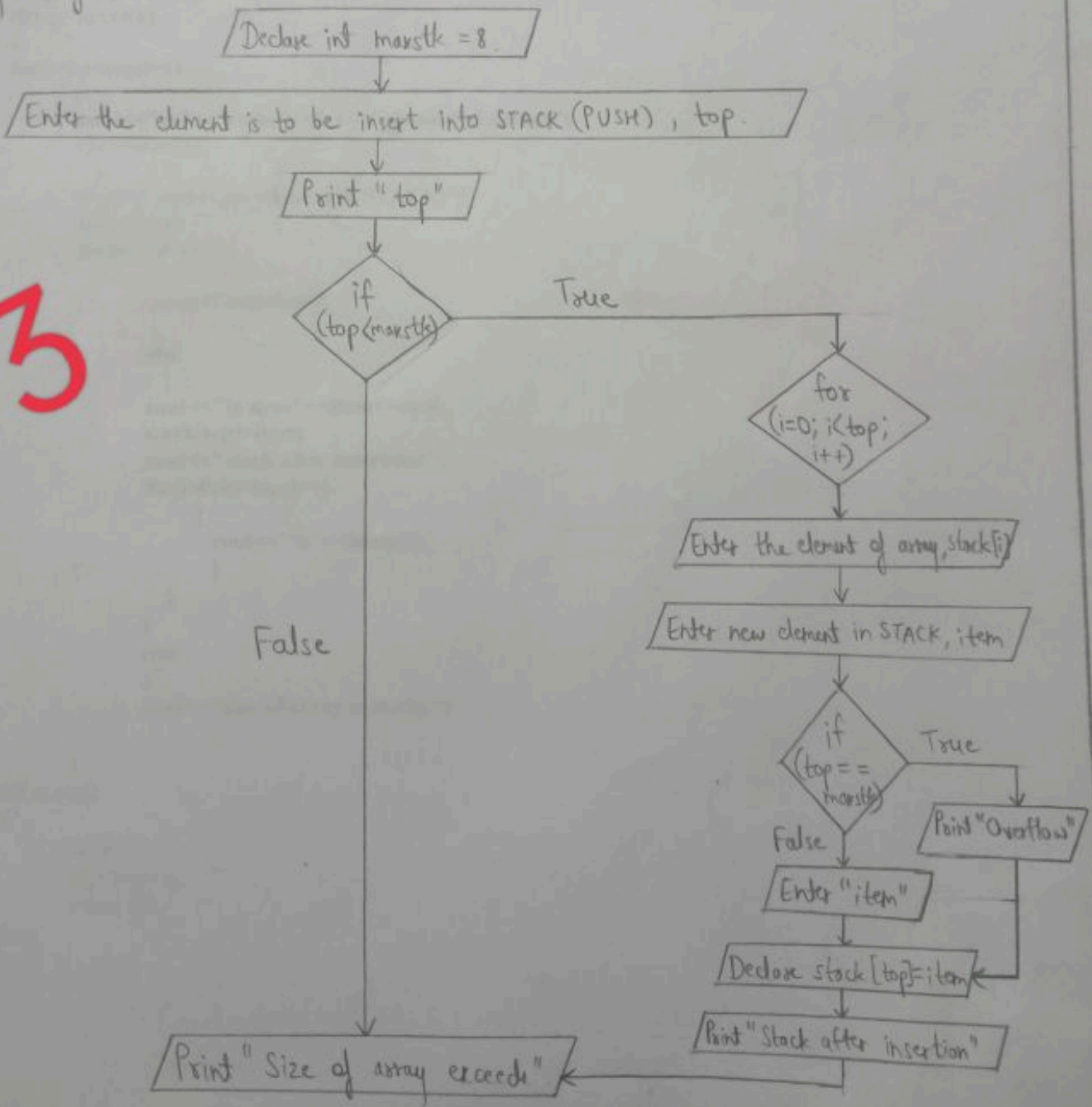


class array

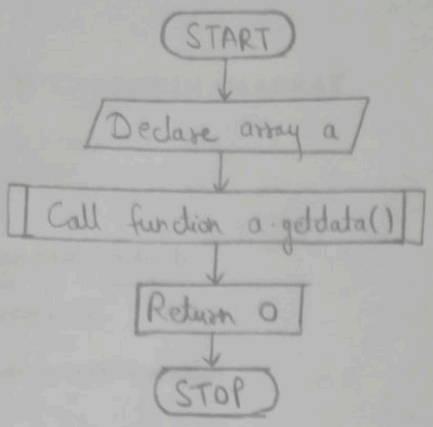


void array :: getdata (void)

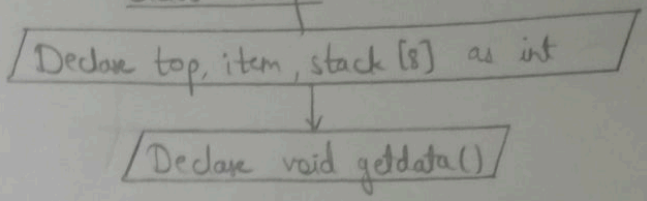
3



Flowchart

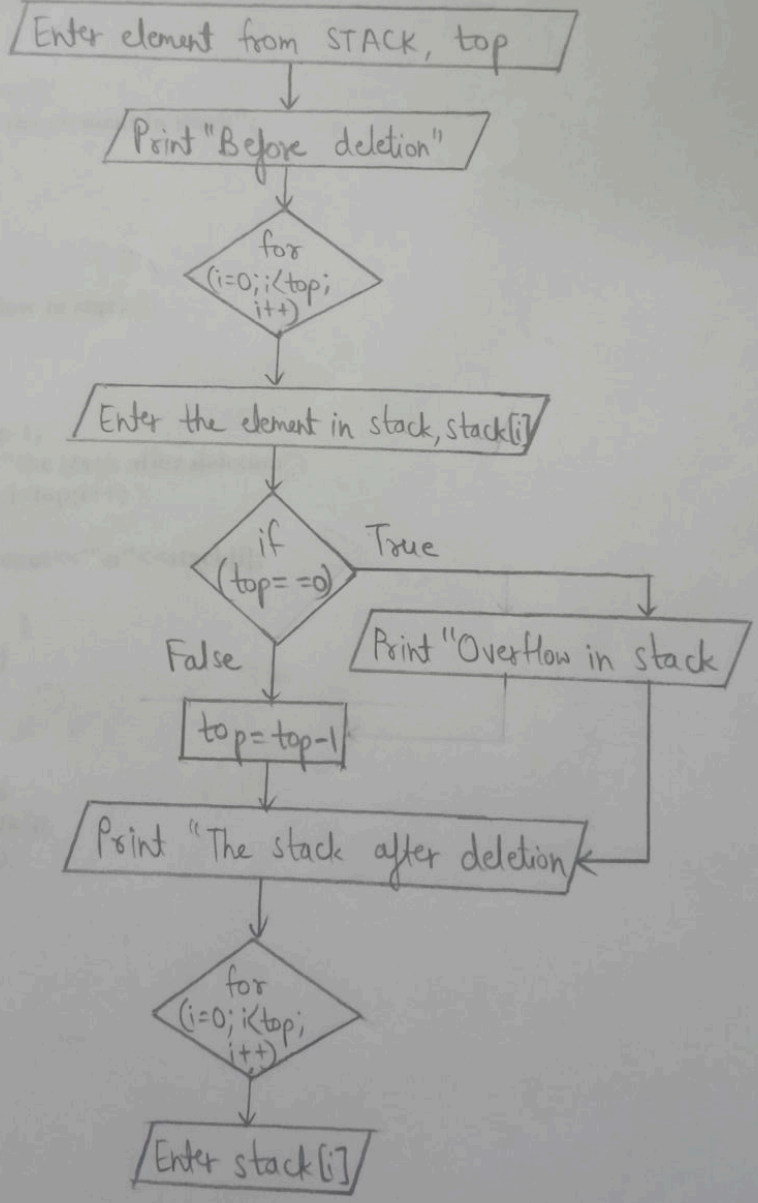


class array

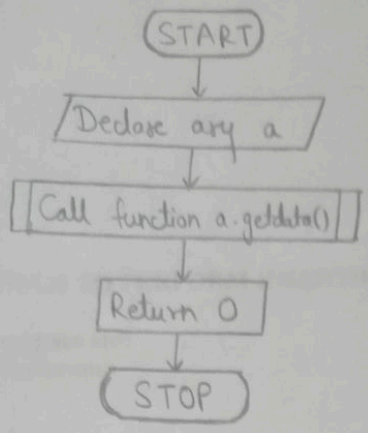


void array :: getdata(void)

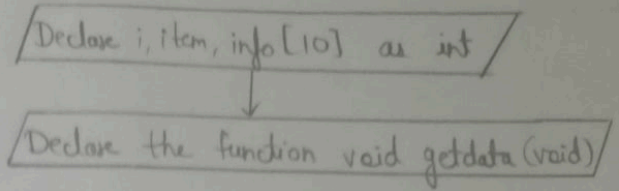
4



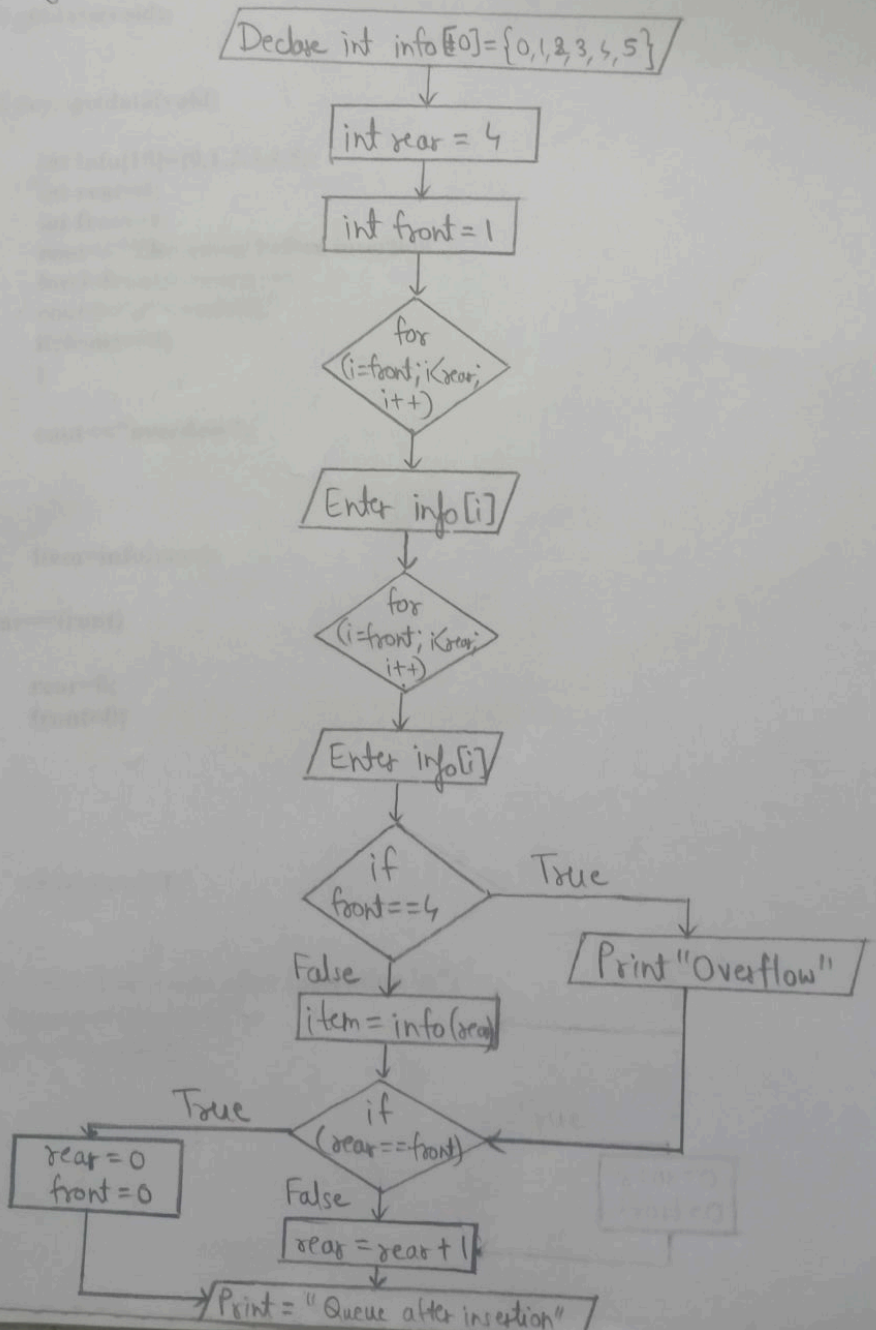
Flowchart



class ary

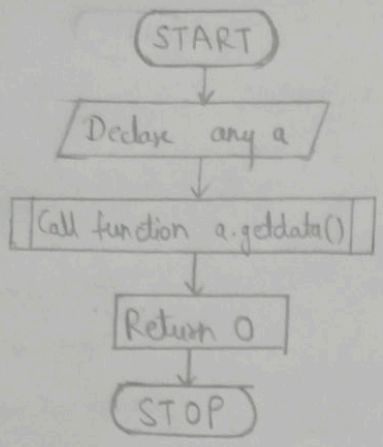


void ary :: getdata(void)

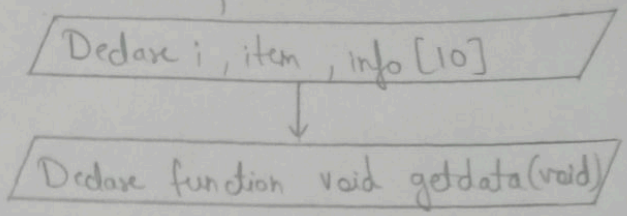


5

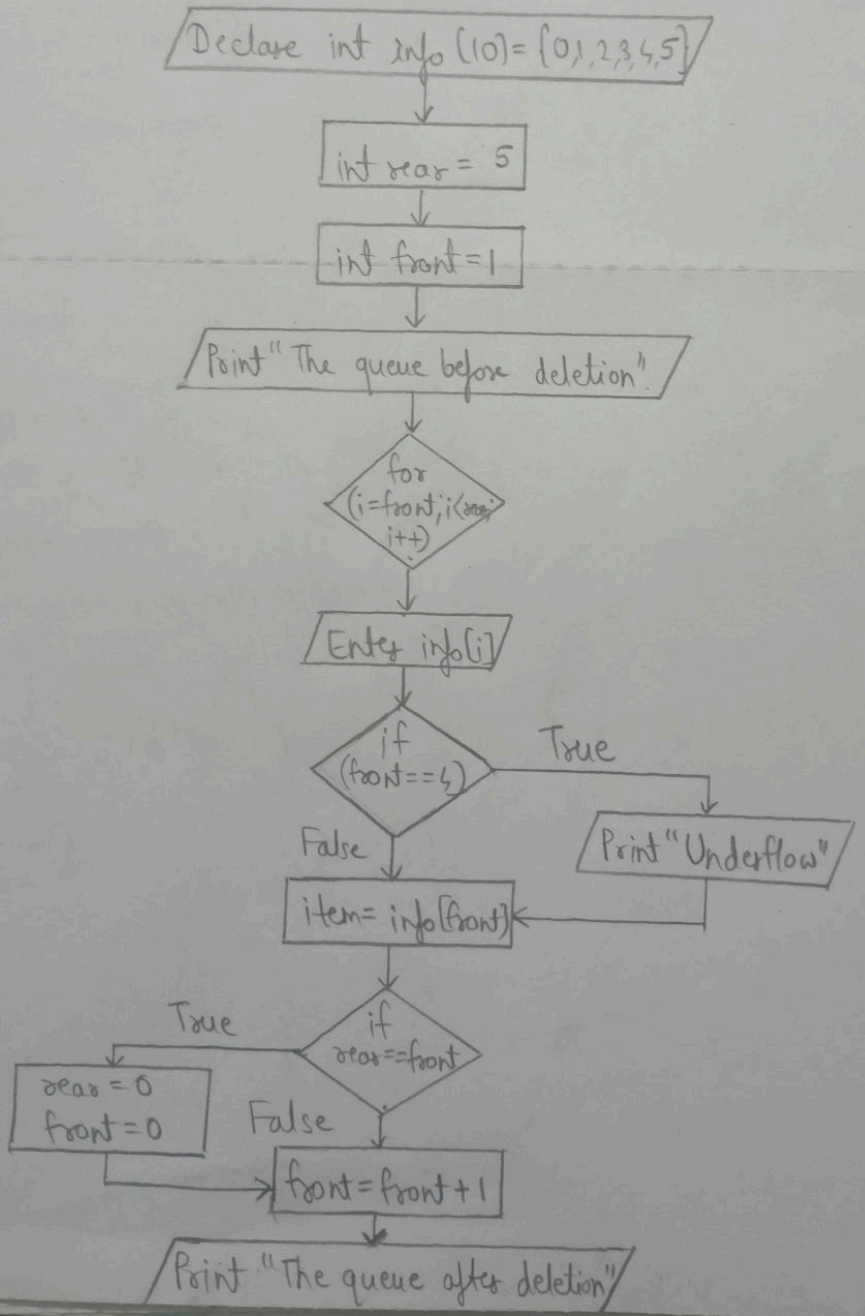
Flowchart



class ary

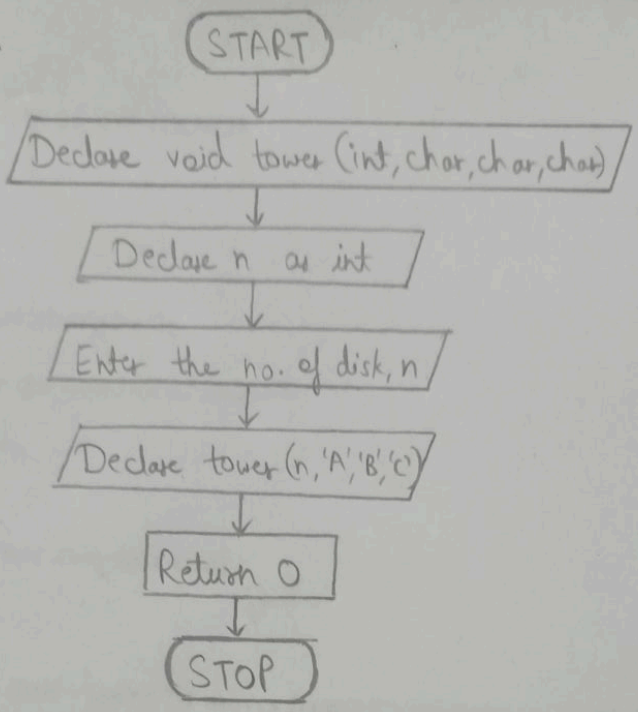


void ary :: getdata (void)



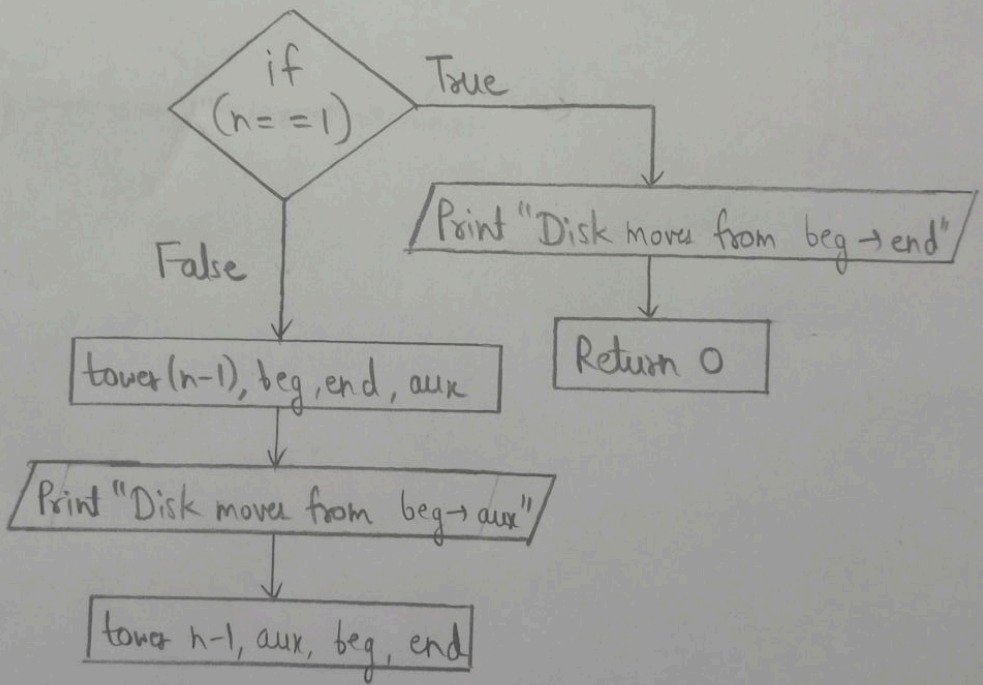
6

Flowchart



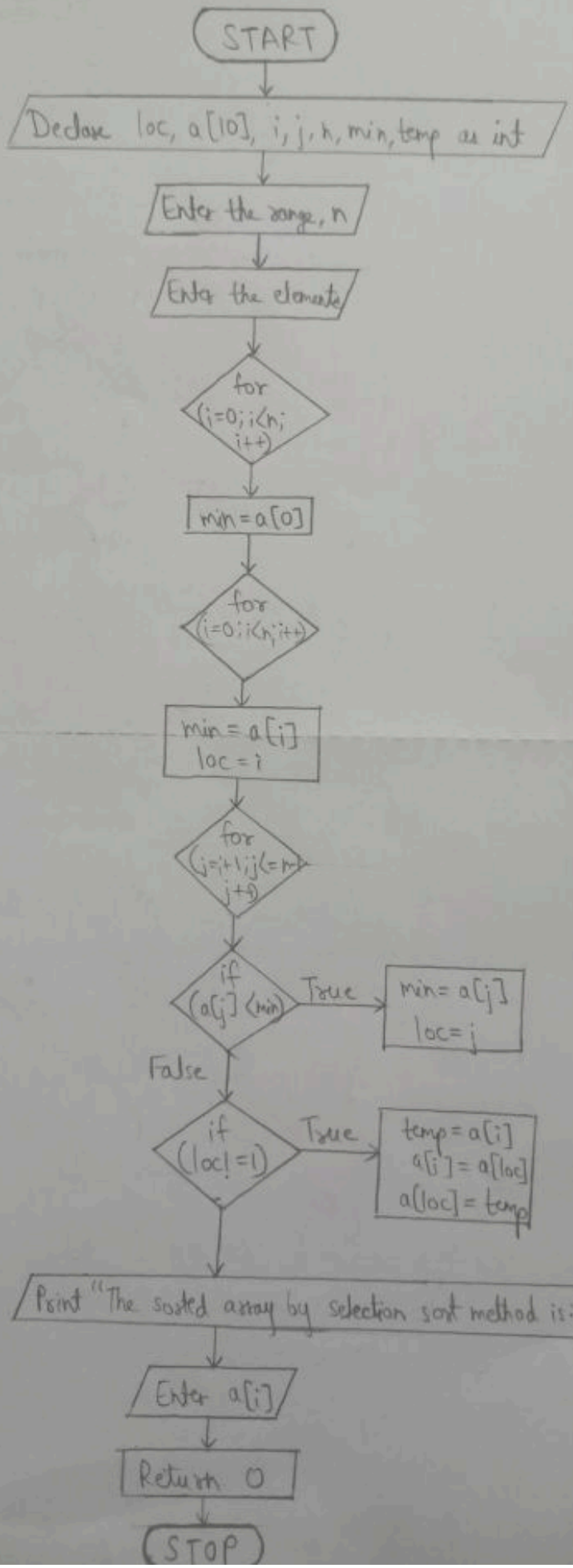
7

int tower (int n, char, beg, char, char end)

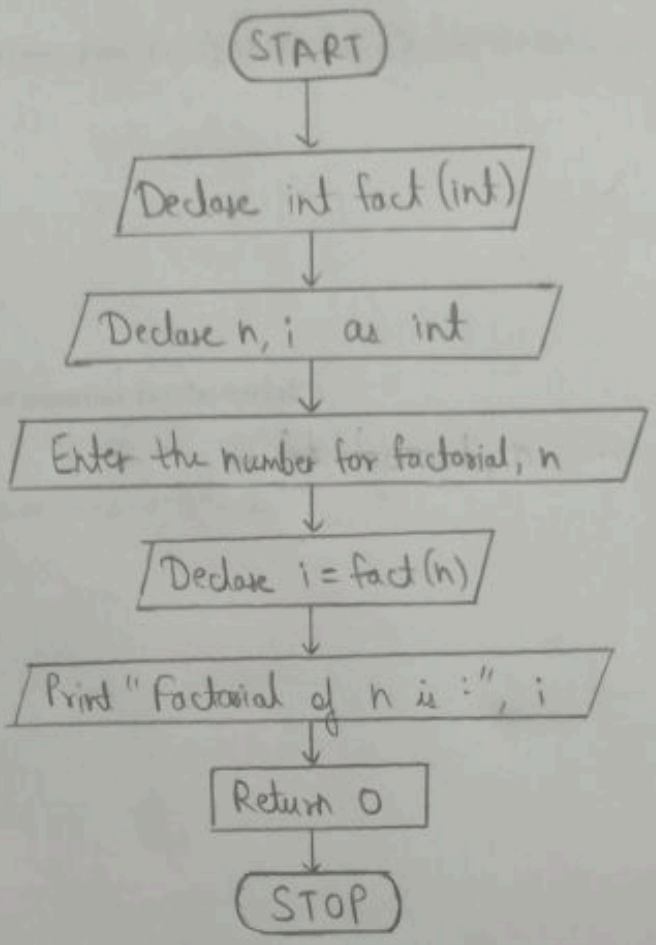


Flowchart

8

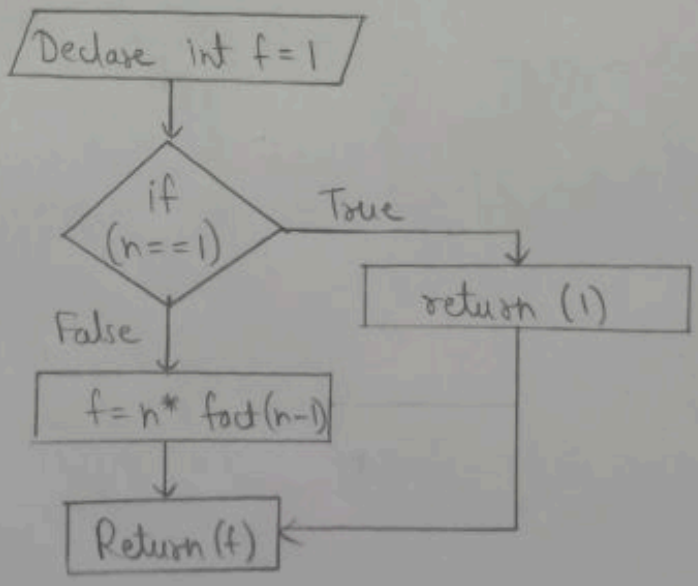

START
Declare loc, a[10], i, j, n, min, temp as int
Enter the range, n
Enter the elements
for (i=0; i<n; i++)
min = a[0]
for (i=0; i<n; i++)
min = a[i]
loc = i
for (j=i+1; j<=n; j++)
if (a[j] < min)
True
min = a[j]
loc = j
False
if (loc != i)
True
temp = a[i]
a[i] = a[loc]
a[loc] = temp
Print "The sorted array by selection sort method is:"
Enter a[i]
Return 0
STOP

Flowchart

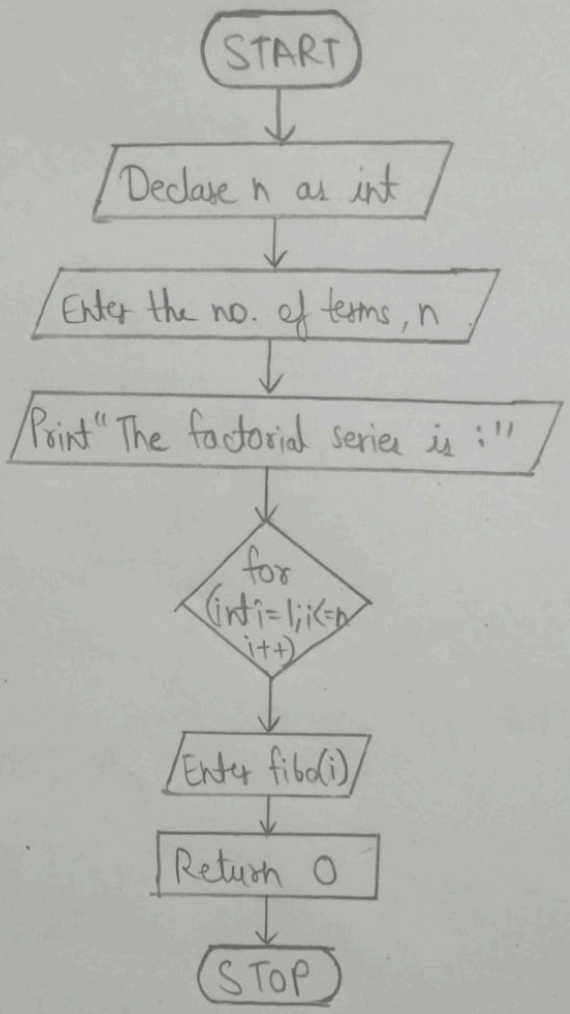


9

int fact (int n)

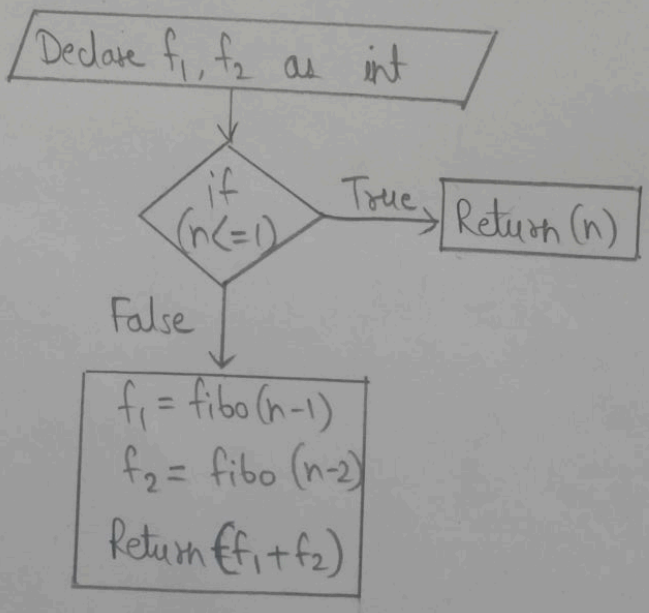


Flowchart

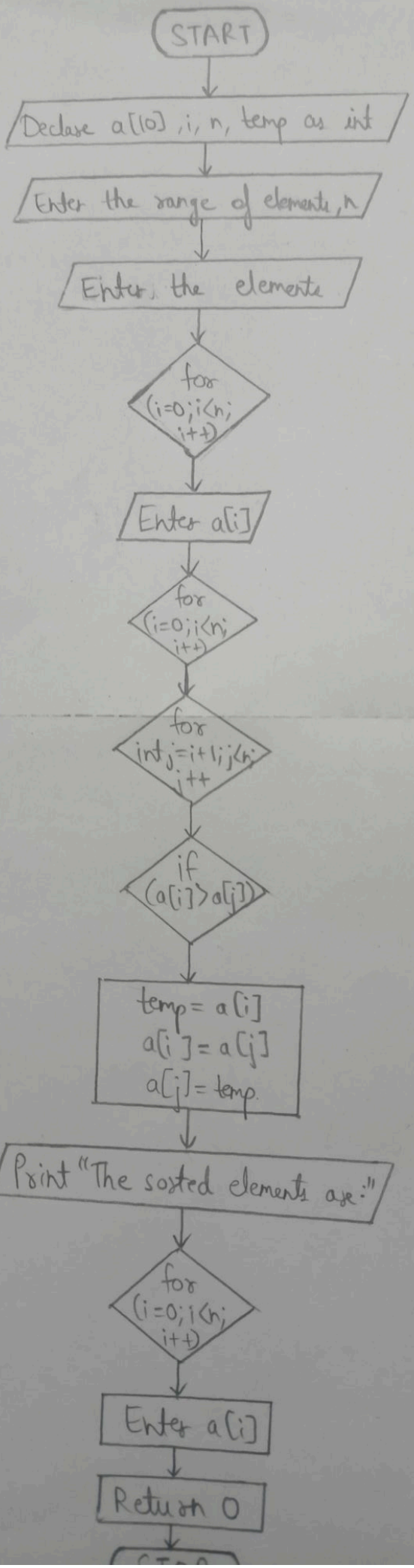


10

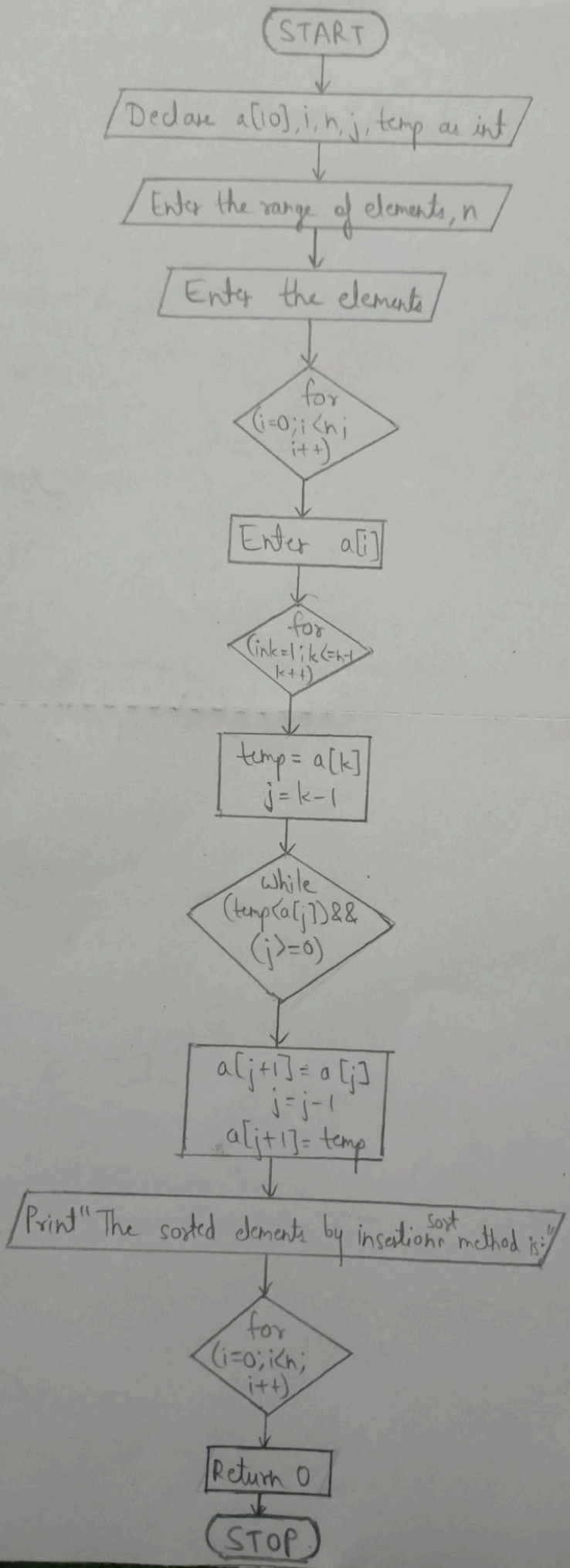
int fibo (int n)



Flowchart

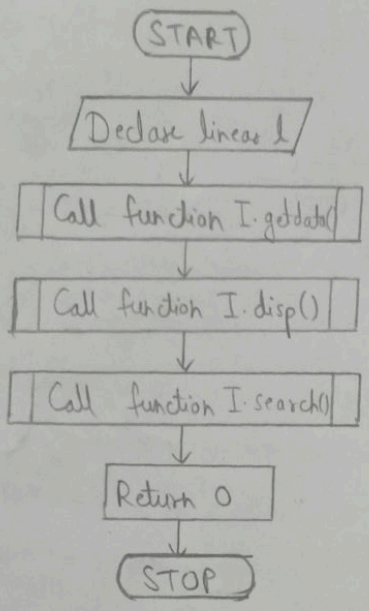


11

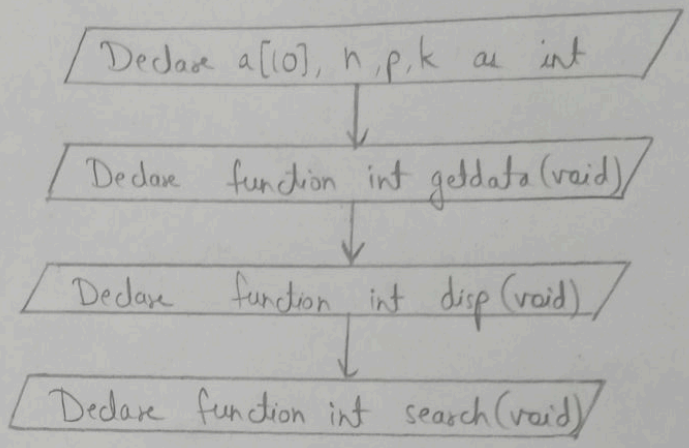

START
Declare a[10], i, n, j, temp as int
Enter the range of elements, n
Enter the elements
for (i=0; i<n; i++)
Enter a[i]
for (int k=1; k<=n-1; k++)
temp = a[k]
j = k-1
while (temp<a[j] && (j>=0)
a[j+1] = a[j]
j = j-1
a[j+1] = temp
Print "The sorted elements by insertion sort method is:"
for (i=0; i<n; i++)
Return 0
STOP

12

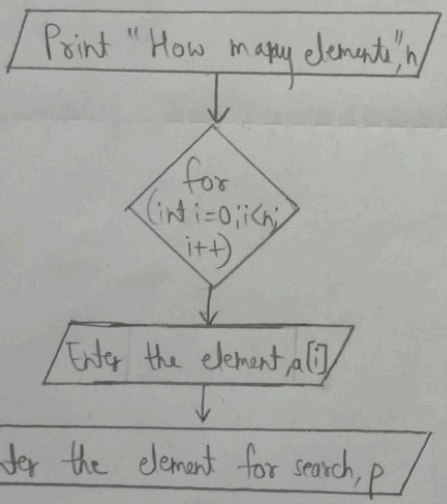
Flowchart



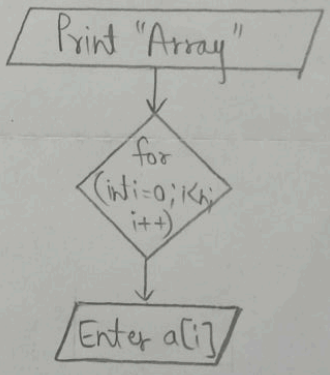
Class linear



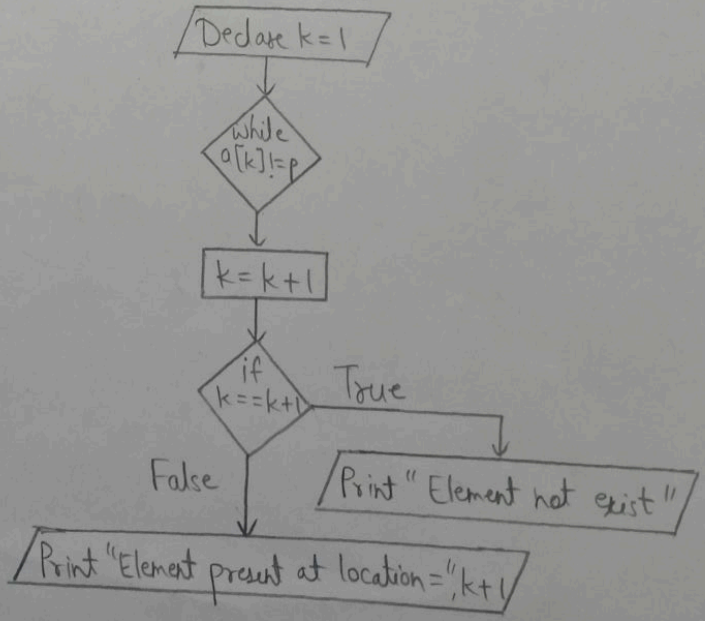
int linear :: getdata(void)



int linear :: disp(void)

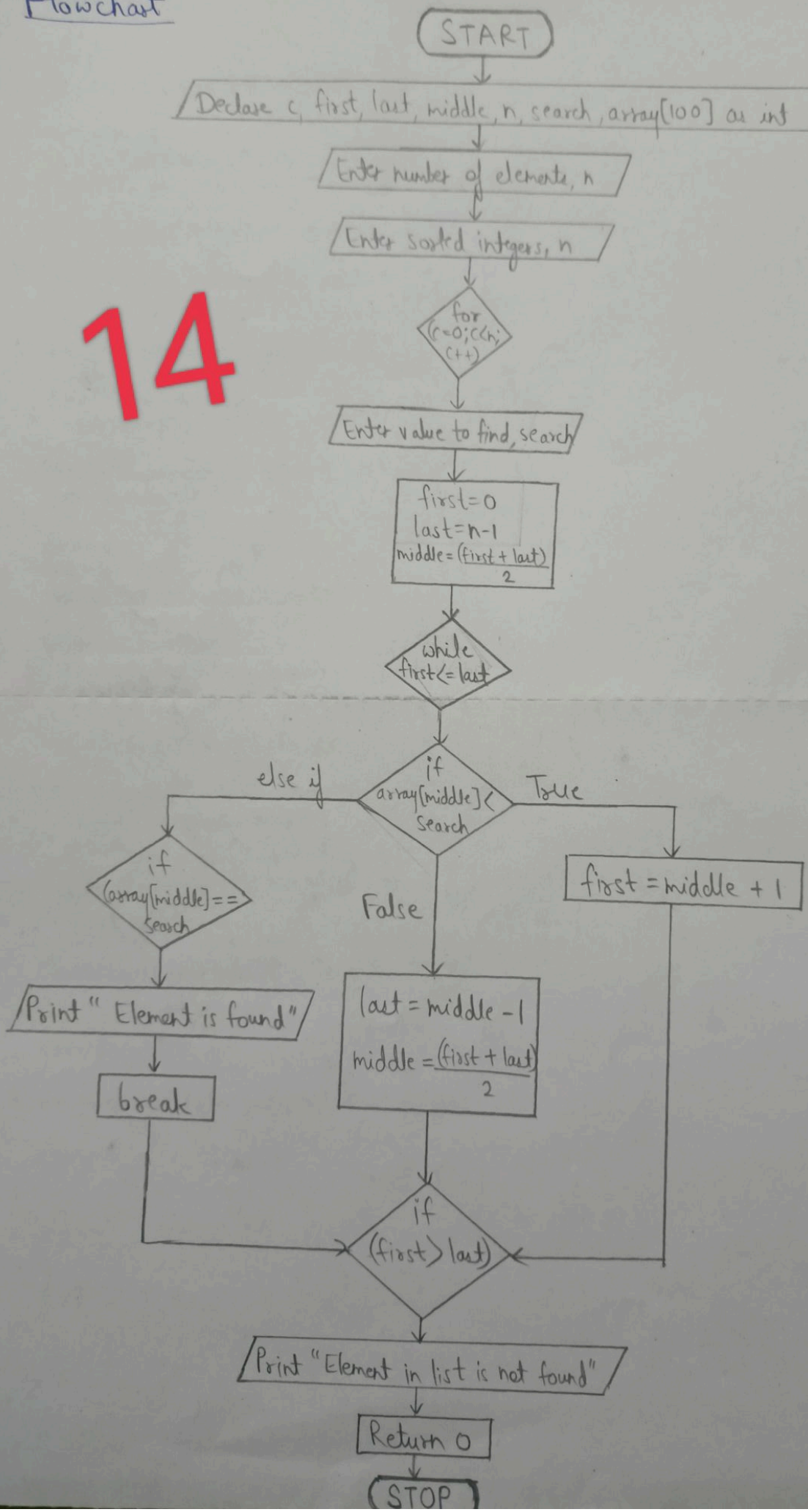


int linear :: search (void)



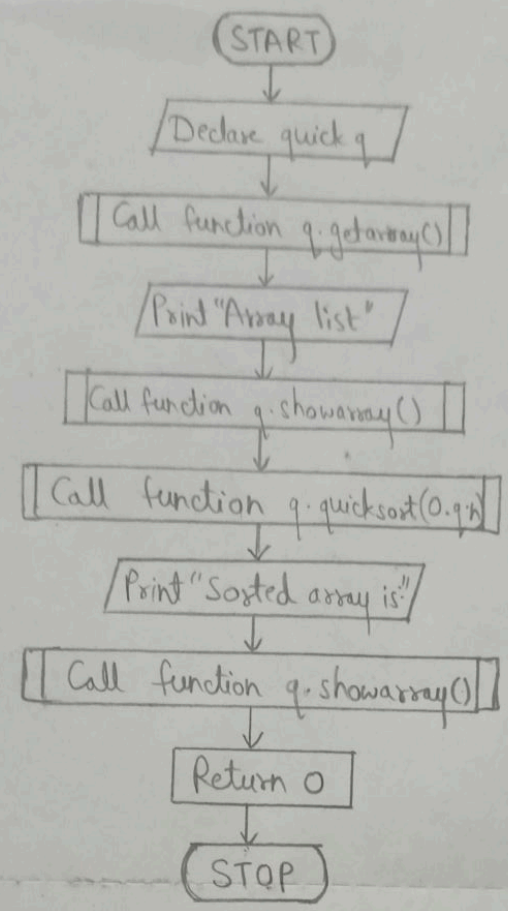
Flowchart

14

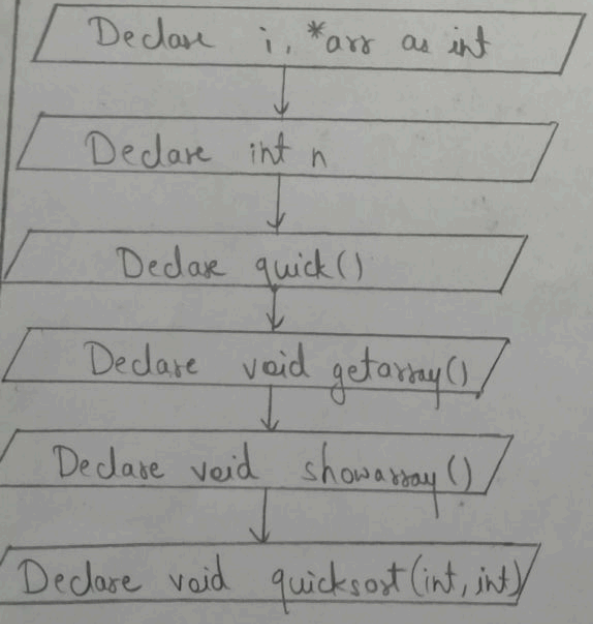


15

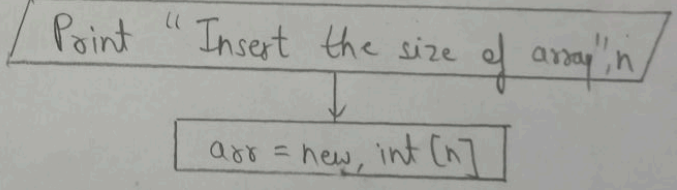
Flowchart



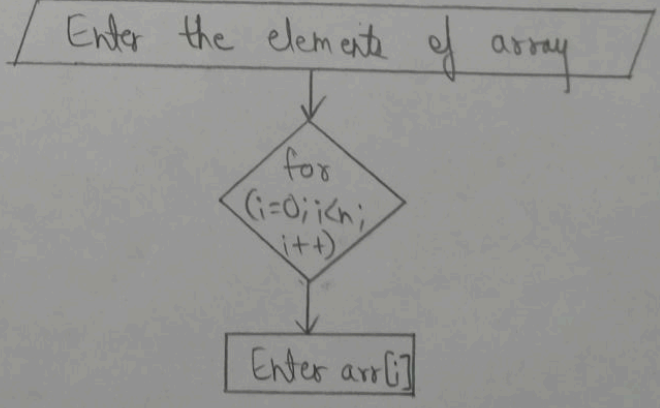
class quick



quick :: quick()



void quick :: getarray()



void quick :: showarray()

void quick :: quicksort (int beg, int end)