1. **Ex1**

**a.**

# input: A – an array of n numbers

for i 0 to n do

for j i + 1 to n – 1 do

if A[i] == A[j] return false

return true

**b.**

def f(A)

# input A is an array of n number

n = len(A)

for i in range(0, n-1):

max = A[i]

imax = i

for j in range(i+1, n):

if A[j] > max:

max = A[j]

imax = j

A[i], A[imax] = A[imax], A[i]

**c.**

Function F(n):

#Input: n = , k integer, positive

#Output: count

count ← 0; i = n

while (i ≥ 1)

for j ← 1 to n do

count ← count + 1

print(j)

end for

i ← i/3

end while

**d.**

ALGORITHMS F(A0..n-1)

for i←0 to n-2 of the

for j←i+1 to n-1 of the

if A[i]==A[j]\*A[j] return false

return true

**e.**

Function f(k):

int i, even;

i := 1;

even := 0;

while (i < k){

even := even + 2

i := i + 1

}

Return even.

f.

Algorithm: f(natural number k)

int i, power

i := 0

power := 1

while (i < k){

power := power \* 2

i := i + 1

}

Return power

**g.** Algorithm bubbleSort(A : list of sortable items)

n := length(A)

repeat

swapped := false

for i := 1 to n-1 inclusive do

/\* if this pair is out of order \*/

if A[i-1] > A[i] then

swap(A[i-1], A[i])

swapped := true

end if

end for

until not swapped

end procedure

**h.**  ALGORITHMS func(A[0..n-1])

for to n – 2 do

for j i + 1 to n – 1 do

if A[i] = A[j] return false

return true

1. Function insertionSort(array A)

01. i ← 1

02. while i < length(A)

03. x ← A[i]

04. j ← i - 1

05. while j >= 0 and A[j] > x

06. A[j+1] ← A[j]

07. j ← j - 1

08. end while

09. A[j+1] ← x

10. i ← i + 1

11.end while