

Write Algorithm for add element at end

Procedure ADD-END (A, n, s, ele)

Description :- This procedure adds element at the end of array. A[i] 'n' is the number of elements present in the array and 's' is the size of an array and 'ele' is the element which to be add at end.

Declaration :-

Global integer A (1:s), n, s, ele
Local integer i

if Is-Full (s = n)

return (true)

else

return (false)

endif

else if Is-EMPTY (n = 0)

A(n) = ele

else

n = n + 1

A(n) = ele

endif

end ADD-END

Write algorithm for add element at the beginning of an array.

Procedure ADD-BEGIN (A, n, ele, s)

Description :- This Procedure adds an element at the beginning of an array A[], 'n' is the number of elements present in an array. 's' is the size of an array and 'ele' is the element which is to be added.

Declaration :-

Global integer A (1:s), n, ele, s

if (n = s), then

print ("Array is Full")

end if n = 0, then

A(n) = ele

else

for i = n to 1 by -1 do

A(i+1) = A(i)

repeat

A(1) = ele

end ADD-BEGIN

write algorithm for adding element at given position in an array.

Procedure ADD_POS (A, S, n, ele, Pos)

Description :- This Procedure adds elements for given particular position of an array. A[] is the number of element present in array. 'Pos' is position of the element which is element to be add. 'ele' is element which is to be add. 'S' size of an array.

Declaration :-

Local integer i;

if $n = S$, then

print ("Array is full")

return (1)

end if

if $Pos \geq 1$ & $Pos \leq n+1$, then

for $i = n$ to pos by -1 do

$A[i+1] = A[i]$

repeat

$A[Pos] = ele$

$n = n + 1$

end if

print ("position is invalid")

end ADD_POS

Write algorithm for element from an array of Particular Position

Procedure DEL_POS (A, n, Pos, ele, s)

Description :- This procedure deletes element from entered position of an array $A[]$, 'n' is to number of element present in the array. 'Pos' is to position of element which is to be deleted, 'ele' is to deleted element & 's' is to array size.

Declaration :-

Global integer :- $A(1:s), n, pos, ele, s$
Local integer i

if $(n=0)$, then

print ("Array is empty")
return (NULL)

$ele = A(Pos)$

For $i = Pos + 1$ to n by $+1$ do

$A(i-1) = A(i)$

repeat

Write Algorithm for Delete from beginning of an array list.

Procedure DEL-BEG (A, n, ele)

Description:- This procedure deletes elements from beginning of an array A []. 'n' is the number of elements present in the array and 'ele' is the element which is deleted from an array list.

Declaration:-

Global integer A (1:n), n, ele

Local integer i

if (n = 0), then

print ("Array is empty")

return (NULL)

endif

ele = A(1)

for i = 2 to n by 1 do

~~A(i-1) = A(i)~~

repeat

~~n = n - 1~~

return (ele)

end DEL-BEG.

Write Algorithm for Delete from end of an array.

Procedure DEL-END (A, n, ele, s)

Description :- This Procedure delete elements at the end of array.

A[], 'n' is the number of elements currently present in an array and 'ele' is the element which is to be delete from array list and 's' is size

Declaration :-

Global integer A (1:s), n, ele

if (n=0), then

print ("Array is Empty")

endif

ele = A(n)

n = n - 1

return (ele)

end DEL-END