

# اصول و مبانی برنامه نویسی




مجید شبیری

کارشناسی ارشد IT، گرایش شبکه  
از دانشگاه صنعتی امیرکبیر



پاسخ تمرینات : تفکر الگوریتمی و تکنیک های حل مسئله

۱. الگوریتم پیدا کردن max سه عدد  $a, b, c$  (با استفاده از متغیر و بدون استفاده از آرایه)
۲. الگوریتم جابجایی (swap) مقدار ذخیره شده در دو متغیر  $a$  و  $b$  (با استفاده از متغیر کمکی)
۳. الگوریتم جابجایی (swap) مقدار ذخیره شده در دو متغیر  $a$  و  $b$  (بدون استفاده از متغیر کمکی)
۴. الگوریتم چاپ مثلث متساوی الاضلاع با کاراکتر '\*' (ارتفاع مثلث از ورودی دریافت می شود) 
۵. الگوریتم یافتن عنصر ماکزیمم آرایه (آرایه ای بصورت  $A = \{L1, L2, L3, \dots, LN\}$  از ورودی دریافت می شود)
۶. الگوریتم مرتب سازی اعداد داخل آرایه (آرایه ای بصورت  $A = \{L1, L2, L3, \dots, LN\}$  از ورودی دریافت می شود)
۷. الگوریتم تشخیص عدد اول (عدد اول عددی است که به جز ۱ و خودش به عدد دیگری بخش پذیر نباشد)

START

Step 1 → Take two integer variables, say A, B & C

Step 2 → Assign values to variables

Step 3 → If A is greater than B & C, Display A is largest value

Step 4 → If B is greater than A & C, Display B is largest value

Step 5 → If C is greater than A & B, Display A is largest value

Step 6 → Otherwise, Display A, B & C are not unique values

STOP

procedure compare(A, B, C)

IF A is greater than B AND A is greater than C  
DISPLAY "A is the largest."

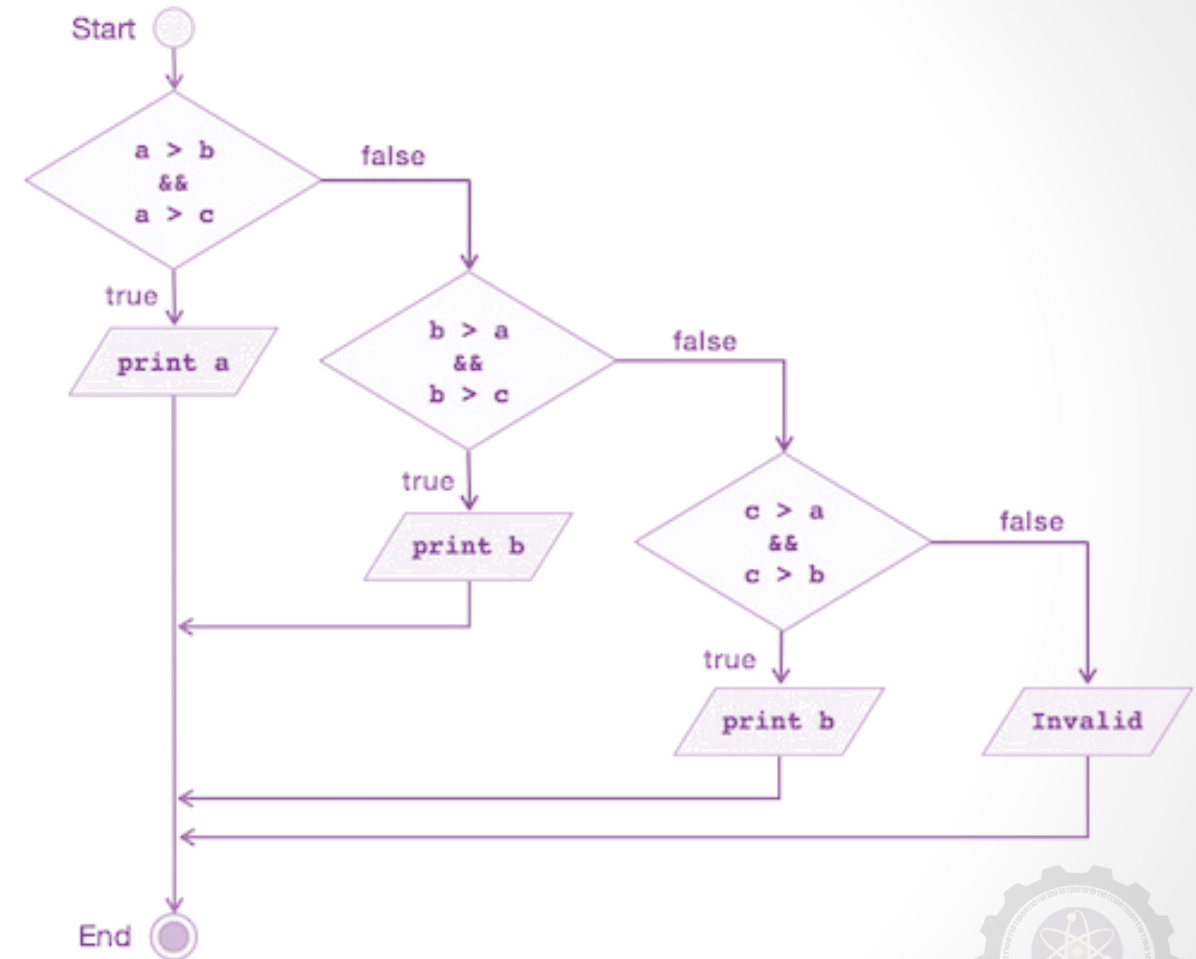
ELSE IF B is greater than A AND A is greater than C  
DISPLAY "B is the largest."

ELSE IF C is greater than A AND A is greater than B  
DISPLAY "C is the largest."

ELSE  
DISPLAY "Values not unique."

END IF

end procedure



START

Var1, Var2, Temp

Step 1 → Copy value of Var1 to Temp

Step 2 → Copy value of Var2 to Var1

Step 3 → Copy value of Temp to Var2

STOP

```
procedure swap(a, b)
```

```
    set temp to 0
```

```
    temp ← a
```

```
    a ← b      // a holds value of b
```

```
    b ← temp   // b holds value of a stored in temp
```

```
end procedure
```

START

Var1, Var2

Step 1 → Add Var1 and Var2 and store to Var1

Step 2 → Subtract Var2 from Var1 and store to Var2

Step 3 → Subtract Var2 from Var1 and store to Var1

STOP

procedure swap(a, b)

```
a ← a + b    // a holds the sum of both
b ← a - b    // b now holds the value of a
a ← a - b    // a now holds value of b
```

end procedure

```

      *
     * *
    * * *
   * * * *
  * * * * *

```

- Step 1 - Take number of rows to be printed, n.
- Step 2 - Make an iteration for n times
- Step 3 - Print " " (space) for in decreasing order from 1 to n-1
- Step 4 - Print "\*" (start, space) in increasing order
- Step 5 - Return

```
procedure equi_triangle
```

```

  FOR I = 1 to N DO
    FOR J = 1 to N DO
      PRINT " "
    END FOR

    FOR J = 1 to I DO
      PRINT "*"
    END FOR
  END FOR

```

```
end procedure
```

$$A = \{ L1, L2, L3, \dots, LN \}$$

START

Step 1 → Take an array A and define its values

Step 2 → Declare largest as integer

Step 3 → Set 'largest' to 0

Step 4 → Loop for each value of A

Step 5 → If  $A[n] > \text{largest}$ , Assign  $A[n]$  to largest

Step 6 → After loop finishes, Display largest as largest element of array

STOP

procedure largest\_array(A)

Declare largest as integer

Set largest to 0

FOR EACH value in A DO

IF  $A[n]$  is greater than largest THEN  
largest  $\leftarrow A[n]$

ENDIF

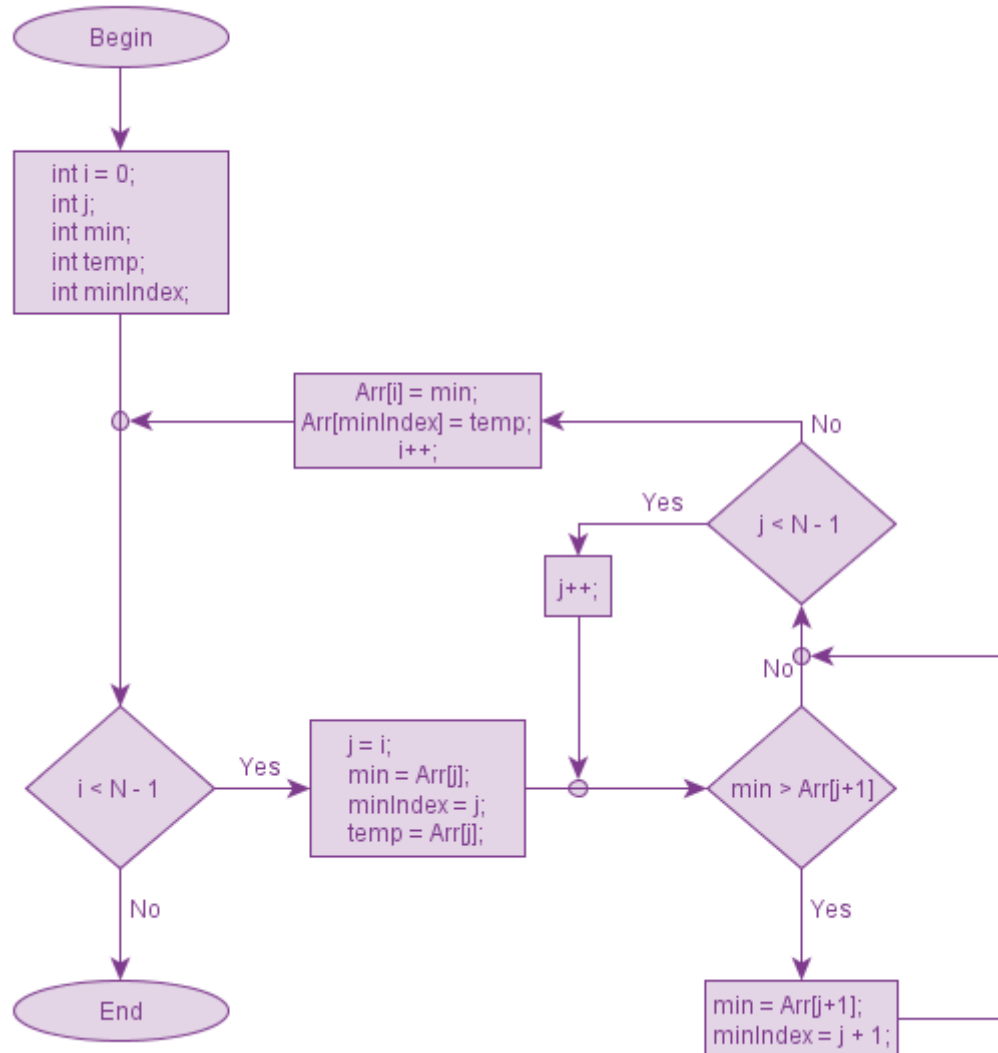
END FOR

Display largest

end procedure

- 1- Initialize minimum value(**min\_idx**) to location 0.
- 2- Traverse the array to find the minimum element in the array.
- 3- **While** traversing
  - if any element smaller than **min\_idx** is found then
  - swap** both the values.
- 4- Then, increment **min\_idx** to point to the **next element**.
- 5- **Repeat** until the array is sorted.





42	16	84	12	77	26	53
----	----	----	----	----	----	----

The array, before the selection sort operation begins.

12	16	64	42	77	26	53
----	----	----	----	----	----	----

The smallest number (12) is swapped into the first element in the structure.

12	16	84	42	77	26	53
----	----	----	----	----	----	----

In the data that remains, 16 is the smallest; and it does not need to be moved.

12	16	26	42	77	84	53
----	----	----	----	----	----	----

26 is the next smallest number, and it is swapped into the third position.

12	16	26	42	77	84	53
----	----	----	----	----	----	----

42 is the next smallest number; it is already in the correct position.

12	16	26	42	53	84	77
----	----	----	----	----	----	----

53 is the smallest number in the data that remains; and it is swapped to the appropriate position.

12	16	26	42	53	77	84
----	----	----	----	----	----	----

Of the two remaining data items, 77 is the smaller; the items are swapped. The selection sort is now complete.

START

Step 1 → Take integer variable A

Step 2 → Divide the variable A with (A-1 to 2)

Step 3 → If A is divisible by any value (A-1 to 2) it is not prime

Step 4 → Else it is prime

STOP

```
procedure prime_number : number
```

```
  FOR loop = 2 to number - 1
```

```
    check if number is divisible by loop
```

```
      IF divisible
```

```
        RETURN "NOT PRIME"
```

```
      END IF
```

```
  END FOR
```

```
  RETURN "PRIME"
```

```
end procedure
```

# اصول و مبانی برنامه نویسی



**مجید شبیری**

کارشناسی ارشد IT، گرایش شبکه

از دانشگاه صنعتی امیرکبیر

