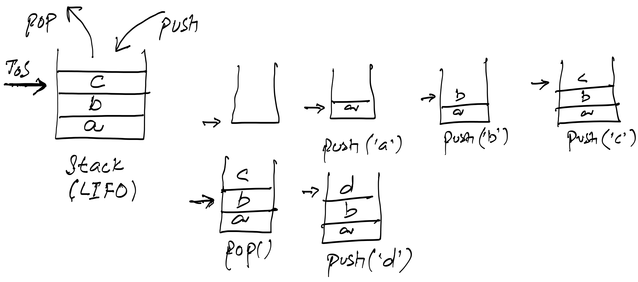
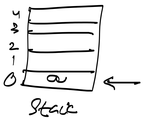
Stack Queues

Saturday, 19 February 2022 4:46 PM



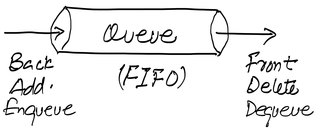
```
// DECLARATION.
DECLARE Stack : ARRAY[0:4] OF CHAR.
DECLARE ToS : INTEGER.
ToS ← -1    // -1 = Null.
```

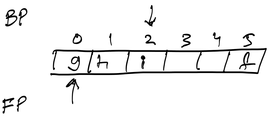



```
// PROCEDURE Push ( X : CHAR)
  IF ToS = 4
    THEN
      OUTPUT "Overflow".
    ELSE
      ToS ← ToS+1
      Stack[ToS] ← X
  ENDIF
END PROCEDURE.
```

```
// Function Pop() RETURNS CHAR
  DECLARE Val : CHAR
  IF ToS = -1 THEN
    RETURN ''
  ELSE
    Val ← Stack[ToS]
    ToS ← ToS-1
    RETURN Val
  ENDIF
END Function.
```

QUEUE

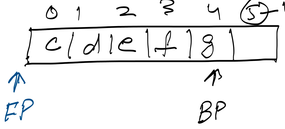





```
Function Dequeue() RETURNS CHAR
  DECLARE X : CHAR.
  IF QSize = 0 THEN
    X ← ''
  ELSE
    IF FP = 5 THEN
      FP = 0
    ELSE
      FP ← FP + 1
    ENDIF
    X = Q[FP]
    QSize ← QSize - 1
  ENDIF
  RETURN X
End Function.
```

```
// QUEUE ( Circular Q) :
  DECLARE Q : ARRAY[0:5] OF CHAR
  DECLARE FP, BP, QSize : INT.
  FP ← -1
  BP ← -1
  QSize ← 0
PROCEDURE Enqueue ( X : Char)
  IF QSize = 6 THEN
    OUTPUT "Overflow!!"
  ELSE
    IF BP = 5 THEN
      BP = 0
    ELSE BP ← BP + 1
    ENDIF
    Q[BP] ← X
    QSize ← QSize + 1
  ENDIF
END PROCEDURE
```

```
// Adjusting Queues.
```




```
// Create Adjusting Q.
Procedure CreateAQ()
    DECLARE i : INT
    For i ← 0 TO 5
        Queue[i] ← ''
    Next.
    FP ← -1
    BP ← -1
End Procedure.
```

```
// Enqueue.
Procedure Enqueue (X : CHAR)
    IF BP = UB Then
        OUTPUT "Overflow!!!"
    ELSE
        BP = BP+1
        Queue[BP] = X
    End if
End Procedure.
```

```
// Dequeue
Function Dequeue() RETURN CHAR
    Declare item : CHAR
    Declare i : INT
    IF BP = -1 Then
        item = ''
    ELSE
        FP = FP+1
        item = Queue[FP]
        For i ← LB TO (UB-1)
            Queue[i] = Queue[i+1]
        Next
        FP = FP-1
        BP = BP-1
    ENDIF
    RETURN item
End Function.
```