Single Linked Lists

Insert_At_Tail(NewNode)

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                    SET HEAD = NewNode
                    SET TAIL = NewNode
          ELSE
                    SET TAIL->NEXT = NewNode
                    SET TAIl = NewNode
          [ END OF IF ]
Step 2 : EXIT
```

Insert_At_Head(NewNode)

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                        SET HEAD = NewNode
                        SET TAIL = NewNode
          ELSE
                        SET NewNode->NEXT = HEAD
                        SET HEAD = NewNode
          [ END OF IF ]
Step 2 : EXIT
```

Insert_At_Position(POS,NewNode)

```
Step 1 :  SET TEMP = HEAD, P = 1
Step 2 : REPEAT WHILE P < POS -1
                    IF TEMP = NULL
                              PRINT “Insertion is not possible”
                              GOTO Step 5
                  [ END OF IF ]
                  SET TEMP = TEMP->NEXT
                  SET P = P + 1
          [ END OF LOOP]
Step 3 : SET NewNode->NEXT = TEMP->NEXT
Step 4 : SET TEMP->NEXT = NewNode
Step 5 : EXIT
```

Delete_At_Tail()

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                     SET RES = NULL
          ELSE IF HEAD = TAIL
                     SET RES = HEAD
                     SET TAIl = NULL
                     SET HEAD = NULL
          ELSE
                     SET TEMP = HEAD
                                  REPEAT WHILE TEMP->NEXT->NEXT
                                         SET TEMP = TEMP->NEXT
                                  [END OF LOOP]
                     SET RES = TAIL
                     SET TEMP->NEXT = NULL
                     SET TAIL = TEMP
          [ END OF IF ]
Step 2 : RETURN RES
```

Delete_At_Head()

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                    SET RES = NULL
          ELSE IF HEAD = TAIL
                    SET RES = HEAD
                    SET TAIl = NULL
                    SET HEAD = NULL
          ELSE
                    SET RES = HEAD
                    SET HEAD = HEAD->NEXT
                    SET RES->NEXT = NULL
          [ END OF IF ]
Step 2 : RETURN RES
```

Delete_At_Position(POS)

```
Step 1 :  SET TEMP = HEAD, P = 1
Step 2 : REPEAT WHILE P < POS -1
                    IF TEMP = NULL
                              SET RES = NULL
                              PRINT “Deletion is not possible”
                              GOTO Step 6
                    [ END OF IF ]
                     SET TEMP = TEMP->NEXT
                     SET P = P + 1
          [ END OF LOOP]
Step 3 : SET RES = TEMP->NEXT
Step 4 : SET TEMP->NEXT = RES->NEXT
Step 5 : SET RES->NEXT = NULL
Step 6 : RETURN RES
```

Circular Linked Lists

Insert_At_Tail(NewNode)

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                      SET HEAD = NewNode
                      SET TAIL = NewNode
                      SET HEAD->NEXT = HEAD
          ELSE
                      SET TAIL->NEXT = NewNode
                      SET TAIl = NewNode
                      SET TAIL->NEXT = HEAD
          [ END OF IF ]
Step 2 : EXIT
```

Insert_At_Head(NewNode)

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                  SET HEAD = NewNode
                  SET TAIL = NewNode
                  SET HEAD->NEXT = HEAD
          ELSE
                  SET NewNode->NEXT = HEAD
                  SET HEAD = NewNode
                  SET TAIL->NEXT = HEAD
          [ END OF IF ]
Step 2 : EXIT
```

Insert_At_Position(POS,NewNode)

```
Step 1 :  SET TEMP = HEAD, P = 1
Step 2 : REPEAT WHILE P < POS -1
                   IF TEMP->NEXT = HEAD
                             PRINT “Insertion is not possible”
                             GOTO Step 5
                 [ END OF IF ]
                 SET TEMP = TEMP->NEXT
                 SET P = P + 1
         [ END OF LOOP]
Step 3 : SET NewNode->NEXT = TEMP->NEXT
Step 4 : SET TEMP->NEXT = NewNode
Step 5 : EXIT
```

Delete_At_Tail()

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                    SET RES = NULL
          ELSE IF HEAD = TAIL
                    SET RES = HEAD
                    SET TAIl = NULL
                    SET HEAD = NULL
          ELSE
                    SET TEMP = HEAD
                            REPEAT WHILE TEMP->NEXT->NEXT != HEAD
                                      SET TEMP = TEMP->NEXT
                              [END OF LOOP]
                    SET RES = TAIL
                    SET TEMP->NEXT = HEAD
                    SET TAIL = TEMP
                    SET RES->NEXT = NULL
          [ END OF IF ]
Step 2 : RETURN RES
```

Delete_At_Head()

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                    SET RES = NULL
          ELSE IF HEAD = TAIL
                    SET RES = HEAD
                    SET TAIl = NULL
                    SET HEAD = NULL
          ELSE
                    SET RES = HEAD
                    SET HEAD = HEAD->NEXT
                    SET TAIL->NEXT = HEAD
                    SET RES->NEXT = NULL
          [ END OF IF ]
Step 2 : RETURN RES
```

Delete_At_Position(POS)

```
Step 1 :  SET TEMP = HEAD, P = 1
Step 2 : REPEAT WHILE P < POS -1
                    IF TEMP->NEXT = HEAD
                              PRINT “Deletion is not possible”
                              SET RES = NULL
                              GOTO Step 6
                     [ END OF IF ]
                    SET TEMP = TEMP->NEXT
                    SET P = P + 1
          [ END OF LOOP]
Step 3 : SET RES = TEMP->NEXT
Step 4 : SET TEMP->NEXT = RES->NEXT
Step 5 : SET RES->NEXT = NULL
Step 6 : RETURN RES
```

Double Linked Lists

Insert_At_Tail(NewNode)

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                    SET HEAD = NewNode
                    SET TAIL = NewNode
          ELSE
                    SET TAIL->NEXT = NewNode
                    SET NewNode->PREV = TAIl
                    SET TAIl = NewNode
          [ END OF IF ]
Step 2 : EXIT
```

Insert_At_Head(NewNode)

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                    SET HEAD = NewNode
                    SET TAIL = NewNode
          ELSE
                    SET NewNode->NEXT = HEAD
                    SET HEAD->PREV = NewNode
                    SET HEAD = NewNode
          [ END OF IF ]
Step 2 : EXIT
```

Insert_At_Position(POS,NewNode)

```
Step 1 :  SET TEMP = HEAD, P = 1
Step 2 : REPEAT WHILE P < POS -1
                    IF TEMP = NULL
                              PRINT “Insertion is not possible”
                              GOTO Step 7
                  [ END OF IF ]
                  SET TEMP = TEMP->NEXT
                  SET P = P + 1
          [ END OF LOOP]
Step 3 : SET NewNode->NEXT = TEMP->NEXT
Step 4 : SET NewNode->PREV = TEMP
Step 5 : SET TEMP->NEXT = NewNode
Step 6 : SET NewNode->NEXT->PREV = NewNode
Step 7 : EXIT
```

Delete_At_Tail()

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                    SET RES = NULL
          ELSE IF HEAD = TAIL
                    SET RES = HEAD
                    SET TAIl = NULL
                    SET HEAD = NULL
          ELSE
                    SET RES = TAIL
                    SET TAIL = TAIL->PREV
                    SET TAIL->NEXT = NULL
                    SET RES->PREV = NULL
          [ END OF IF ]
Step 2 : RETURN RES
```

Delete_At_Head()

```
Step 1 :  IF HEAD = NULL and TAIL = NULL
                    SET RES = NULL
          ELSE IF HEAD = TAIL
                    SET RES = HEAD
                    SET TAIl = NULL
                    SET HEAD = NULL
          ELSE
                    SET RES = HEAD
                    SET HEAD = HEAD->NEXT
                    SET HEAD->PREV = NULL
                    SET RES->NEXT = NULL
          [ END OF IF ]
Step 2 : RETURN RES
```

Delete_At_Position(POS)

```
Step 1 :  SET TEMP = HEAD, P = 1
Step 2 : REPEAT WHILE P < POS -1
                    IF TEMP = NULL
                              SET RES = NULL
                              PRINT “Deletion is not Possible”
                              GOTO Step 8
                    [ END OF IF ]
                    SET TEMP = TEMP->NEXT
                    SET P = P + 1
          [ END OF LOOP]
Step 3 : SET RES = TEMP->NEXT
Step 4 : SET TEMP->NEXT = RES->NEXT
Step 5 : SET TEMP->PREV = RES->PREV
Step 6 : SET RES->PREV = NULL
Step 7 : SET RES->NEXT = NULL
Step 8 : RETURN RES
```