Phonebook Application Pseudocode

```
MODULE Phonebook

  DEFINE MAX_CONTACTS = 100

  DEFINE contactList AS ARRAY[0...MAX_CONTACTS]

  DEFINE currentSize AS INTEGER = 0


  FUNCTION InsertContact(name, phoneNumber)

    IF currentSize < MAX_CONTACTS THEN

      contactList[currentSize] = new Contact(name, phoneNumber)

      currentSize = currentSize + 1

      PRINT "Contact added: " + name

    ELSE

      PRINT "Phonebook is full!"

    ENDIF

  ENDFUNCTION


  FUNCTION SearchContact(name)

    FOR i FROM 0 TO currentSize - 1 DO

      IF contactList[i].name EQUALS name THEN

        RETURN contactList[i]

      ENDIF

    ENDFOR

    RETURN NULL

  ENDFUNCTION


  FUNCTION DisplayAllContacts()
```

```
    IF currentSize = 0 THEN

        PRINT "No contacts available."

    ELSE

        FOR i FROM 0 TO currentSize - 1 DO

            PRINT contactList[i].name + ": " + contactList[i].phoneNumber

        ENDFOR

    ENDIF

ENDFUNCTION


FUNCTION DeleteContact(name)

    FOR i FROM 0 TO currentSize - 1 DO

        IF contactList[i].name EQUALS name THEN

            contactList[i] = contactList[currentSize - 1]  // Replace with last contact

            currentSize = currentSize - 1

            PRINT "Contact deleted: " + name

            RETURN TRUE

        ENDIF

    ENDFOR

    PRINT "Contact not found: " + name

    RETURN FALSE

ENDFUNCTION


FUNCTION UpdateContact(name, newPhoneNumber)

    FOR i FROM 0 TO currentSize - 1 DO

        IF contactList[i].name EQUALS name THEN

            contactList[i].phoneNumber = newPhoneNumber

            PRINT "Contact updated: " + name

            RETURN TRUE
```

```
        ENDIF
      ENDFOR
      PRINT "Contact not found: " + name
      RETURN FALSE
    ENDFUNCTION


    FUNCTION SortContacts()
      FOR i FROM 0 TO currentSize - 1 DO
        FOR j FROM 0 TO currentSize - i - 1 DO
          IF contactList[j].name > contactList[j + 1].name THEN
            SWAP contactList[j], contactList[j + 1]
          ENDIF
        ENDFOR
      ENDFOR
      PRINT "Contacts sorted."
    ENDFUNCTION


    FUNCTION AnalyzeSearchEfficiency(targetName)
      START TIMER
      SearchContact(targetName)
      STOP TIMER
      PRINT "Search time: " + ELAPSED TIME
    ENDFUNCTION
  ENDMODULE
```