PSUEDOCODE

```
PROGRAM Phonebook

// Global variables
names = empty list
numbers = empty list


MODULE main()
    exit = false
    WHILE NOT exit
        displayMenu()
        decision = getUserChoice()

        SWITCH decision
            CASE 1: insertContact()
            CASE 2: searchContact()
            CASE 3: displayAllContacts()
            CASE 4: deleteContact()
            CASE 5: updateContact()
            CASE 6: sortContacts()
            CASE 7: exit = true
            DEFAULT: PRINT "Invalid choice, try again"
        END SWITCH
    END WHILE

    PRINT "THANK YOU FOR USING THE PHONEBOOK APPLICATION"
END MODULE



MODULE sortContacts()
    PRINT "1: Sort contacts in alphabetical order."
    PRINT "2: Sort contacts in ascending order of numbers."
    PRINT "Enter your choice: "
    choice = getUserChoice()

    IF choice == 1 THEN
        sortByName()
    ELSE IF choice == 2 THEN
        sortByNumber()
    ELSE
        PRINT "Invalid choice"
        RETURN
    END IF

    PRINT "Contacts sorted successfully."
    displayAllContacts()
END MODULE




MODULE searchContact()
    PRINT "Enter number you are searching for: "
    searchNo = getUserChoice()

    found = false
    FOR i = 0 TO numbers.size() - 1
        IF numbers[i] == searchNo THEN
            PRINT "Contact found at position " + (i + 1)
            PRINT names[i] + " " + numbers[i]
            found = true
            BREAK
        END IF
    END FOR

    IF NOT found THEN
        PRINT "Contact not found"
    END IF
END MODULE



FUNCTION displayMenu()
    PRINT menu options
    PRINT "Enter the number of the option you want to select: "
END FUNCTION

FUNCTION getUserChoice()
    WHILE input is not an integer
        PRINT "Invalid input. Please enter a number."
        CLEAR input
    END WHILE
    RETURN integer input
END FUNCTION
```

```
FUNCTION insertContact()
    PRINT "Enter the number you want to insert: "
    number = getUserChoice()
    numbers.add(number)

    PRINT "Enter contact name: "
    name = get user input
    names.add(name)

    PRINT "Contact added successfully."
END FUNCTION


FUNCTION displayAllContacts()
    IF names is empty THEN
        PRINT "No contacts to display."
        RETURN
    END IF

    FOR i = 0 TO names.size() - 1
        PRINT (i + 1) + ". " + names[i] + " " + numbers[i]
    END FOR
END FUNCTION

FUNCTION deleteContact()
    PRINT "Enter name or number of contact you want to delete: "
    input = get user input

    deleted = false
    FOR i = 0 TO names.size() - 1
        IF names[i] equals input (case insensitive) OR numbers[i] as string equals input THEN
            remove names[i]
            remove numbers[i]
            PRINT "Contact deleted successfully."
            deleted = true
            BREAK
        END IF
    END FOR

    IF NOT deleted THEN
        PRINT "Contact not found"
    END IF
END FUNCTION
```

```
FUNCTION updateContact()
   PRINT "Enter name or number of contact you want to update: "
   input = get user input

   FOR i = 0 TO names.size() - 1
      IF names[i] equals input (case insensitive) OR numbers[i] as string equals input THEN
         PRINT "Enter new name: "
         newName = get user input
         PRINT "Enter new number: "
         newNumber = getUserChoice()

         names[i] = newName
         numbers[i] = newNumber
         PRINT "Contact updated successfully."
         RETURN
      END IF
   END FOR

   PRINT "Contact not found"
END FUNCTION


FUNCTION sortByName()
   entries = empty list
   FOR i = 0 TO names.size() - 1
      entries.add(new Entry(names[i], numbers[i]))
   END FOR

   sort entries by key (name)

   clear names and numbers
   FOR EACH entry IN entries
      names.add(entry.key)
      numbers.add(entry.value)
   END FOR
END FUNCTION



FUNCTION sortByNumber()
   entries = empty list
   FOR i = 0 TO numbers.size() - 1
      entries.add(new Entry(numbers[i], names[i]))
   END FOR
```

```
    sort entries by key (number)

    clear names and numbers
    FOR EACH entry IN entries
        numbers.add(entry.key)
        names.add(entry.value)
    END FOR
END FUNCTION

END PROGRAM
```