DSA521S Project

Group 13

|  |  |  |
| --- | --- | --- |
| Stu.Number | Stu.Name | Created module for |
| 224031066 | Nangukuii Kangootui | Delete contact |
| 223113298 | Kayando Kandjungu | Search contact |
| 224035622 | Anopaishe Mukwendi | Display contact |
| 224013092 | Gabriel Kasoma | Sorting contacts |
| 224010395 | Rauha Kadhingula | Update contact |
| 224016997 | Emily Mudjanima | Insert contact |

**Table of Content**

Description…………………………………………………………………………………………………………………… [2]

Data structure used …………………………………………………………………………………………………….. [2]

Pseudocode representation ………………………………………………………………………………………… [2]

Flowchart representation

**Short Description**

A Namibian telecommunications company is looking for an efficient phonebook algorithm to perform typical phonebook operations such as, insert, search, delete, and update and sort contacts. Linked lists is one of the simple algorithms that can deliver these operations efficiently.

**Data structure used**  
Linked lists

**Pseudocode representation**

Start

Print “welcome to the phonebook application”

Print “1. Display Contacts”

Print “2. Search Contact”

Print “3. Insert Contact”

Print “4. Delete Contact”

Print “5. Update Contact”

Print “Please enter the number for the function you want to carry out:”

Get number

If ( number==1) then

displayContact()

else if ( number==2) then

searchContact()

else if (number==3) then

Prompt user for name, phoneNumber to insert

Get name, phoneNumber

insertContact(name,phoneNumber)

else if( number==4) then

Prompt user for name to delete

Get name

deleteContact(name)

else if (number==5) then

Prompt user for contact name and new contact number

get contact name and new contact number

updateContact(name,newNumber)

else

display “INVALID OPTION”

endif

endif

endif

endif

endif

displayContact() {

IF (head == NULL) THEN

PRINT “Phonebook is empty.”

RETURN

ENDIF

temp = head

WHILE (temp != NULL) {

PRINT “ Name: “ + temp.name + “ , Phone “ + temp.phone

temp = temp.next

}

END WHILE

}

searchContact() {

Print “ Enter a name to search “

Get searchItem

temp = head

found = 0

While (temp ==! null )

If (temp.name == searchItem ) THEN

DISPLAY “Contact found: “ + current.name + “, Phone number: “+ temp.phoneNumber

found = 1

EndIf

temp = temp.next

EndWhile

If (found == 0) THEN

DISPLAY “Contact not found”;

EndIf

}

newNode(){

String name

Int phoneNumber

node next

}

insertContact(name, phoneNumber){

newNode()

newNode.name = name

newNode.phoneNumber=phoneNumber

newNode.next = null

If (head == null) Then

head = newNode

Else

Temp=head

While (temp.next ! = null )

temp = temp.next

EndWhile

temp.next = newNode

endif

sortPhoneBook(newNode)

}

sortPhoneBook(newNode){

If (head== null) Then

Display “Phonebook is sorted”;

End if

Temp=head

While (temp.next ! = null )

IF (temp.name -> temp.next.name) THEN

temp = temp.name

temp.name = temp.next.name

endif

temp.next.name = temp

EndWhile

PRINT “Phonebook sorted”;

}

delete(name){

if( head == null) Then

Display “PhoneBook is empty”

RETURN

endif

if(head.name = name) then

temp=head

head= head -> next

temp=free

DISPLAY “Contact deleted” + name

endif

temp=head

while (temp.next!= null)

if(temp.next.data.name = = name)

nodeToDelete = temp.next

temp.next = temp.next.next

free = nodeToDelete

temp=temp.next

DISPLAY “Contact deleted: “ + name

Else

Display “Contact not found”

endif

endwhile

}

updateContact(name,newNumber) {

temp = head;

while (temp != null)

if (temp.data.name=name)

temp.number = newNumber;

Print “Contact updated successfully.”

return

endif

temp = temp.next;

PRINT “Contact not found”

    }