### Shak alking Linked list

Step 1: 8-1 art

Step 2 : Declare the mode and the prequired

Step 3: Declare the bunctions for push
Pop display and reach.

Step 4: Read the choice from the user to push, pop display or scatch an clement.

Step 5: If the user choose to push on element then read the element to be pushed and call the functions to push the clement by passing the value to the functions.

Step 5:1: Deelnee the new mode and allocate memory for the new node.

. But newhode to date = walne . Step 5 : 2 : Charle if top = = mull. Stop 5 . a then set howwards a next = null : Else set newnode > next = top. step 5 . 4 : Set top = neumode and then Step 5.5 point innection is successful step 6 I if the user choosed to pup on element from the stock then call the function to pop the alement. : which if top = = mull them Step 6.1 point stack is empty. " Else obsolve a pointer variable temp and initialize it to top. stop 6.2 Step 6.3 : Point the element that is being deleted. step 8 4 : set temp = temp = next. Step 6.5 : free the temp. : If the week chouse to display step + the element in the stock then call the function to display the element is the Stack,

```
sheet if top = = null the print
Step 7.1:
            Stock is anopty
            Else declare a printer variable
step 7. 2:
             -tamp and initialize its top
            : Repeate stop 7.4 and 7.5
step
     7. 3
             While temp so next 1 = next.
            : must temp - date
     7. 4
Step
             : set temp = temp = next
Hep
     705
             . If the user choose to seatch
     4
step
             on element from the stack
             then call the bunction to
              Scaech an element.
            : Decline a pointer variable ptr
Stop
     8 . 1
              and other necessary racialle
            : Initialize pto = top,
     8.2
Stop
            : Check if ptr = null . then
     8.3
step
             print stack is empty.
            : Else send the clament to
     8-4
step
              be searched brooms user
             : Repeate the stop 8.6 and to
Step
     8-5
              x, x while ptr 1 = null.
```

Step 8:6: Wheele if ptn => data == item

then point alement Ground

and to location and set

flag = 1

Step 8:7: Also set blag = 0.

Step 8:8: Incomment i by 1 and set

pts = pts -> nist.

Step 8:9: Wheele if flag == 0. Here

print clament not bound.

Step 9: Stop.

# (Enquene, Dequeue) Montas Queue

Step 1 : Start.

step 2: Declare the cause and required

step 3: Delease the bunction bur orquere, dequere, display and search.

Step 4 : Road the chouse booms the user to enquere, dequene display and seach.

Step 5: If the user choose the options enquene then read the element to be inserted from the user, the Call the function enquene and pair the value to the bunction.

Stop 5.1: Whack if bount = =-1 and rear = -- 1 then get fromto 0, ocar = u and set quene (rear) - rlument. else if reart 1 mod max==for Step 5. 7 or front = = rear + 1 then Pint cenene is orceflow. : Elu set rear = reart med Step 5.3 max and set quene ["rear"]= element. : If the urser chause the option Step & dequene then call the function dequene : where if bont = = -1 and step bil "scar - = - 1 then point oxusure is und caplons. : Else sheds if boot == reer Stop 5. 2 then point the clament is to be deleted. Then set front = -1 and 8004 = -1. : Else point the element to step 6.3 be degrened get browst = 6 mitte mod max.

```
stop 7: If the user chause the uption
        to display the queue the call
        the bunction display
Step 7. 9: wheek if bront = -- 1 and
             beer = = - 1 than print
              ceuse is empty.
stop 4. 2
            : Else ropeute the stop 7-3
             While ic = rear.
Step 7. 3
            ! point queue (i) and set
             i = i+1 mod max.
Stop 8.
           : If the user choose to search
             an element in the quent
             then call the junction to
             search on Element in
             quene.
8.1
           i Read the element to be searched
             in the quene.
           : esheek if item = = quene(1)
 Stop
              then point item focused
              and its position and increment
            : cheek if c = = 0. then print
Step 6-3
              item not bucushed.
step 9
            · Encl.
```

#### Merging

Step 1 1 start . Delare the variables. Step 2 : Read the size of the bisst Step 3 anny. : Read the element of first Step 4 among in sosted order . Read the size of the second Step 5 array. : Road the claments of second Step 6 army in sorted order. . Repeat step 8 and 9 while Step 7 1 4 m 4 17 M : cheek if alij = b[j] then Step 8 C[k++] = b[3++]

Step 7: Else C[k++] = a[i++]

Step 10: Repeat Step 11 while 12 m.

Step 11: C[k++] = a[i++].

Step 12: Repeat Step 13 while Jen.

Step 13: C(k++) = b[i++].

Step 13: C(k++) = b[i++].

Step 14: Print the Birst newm.

Step 14: Print the Birst newm.

Step 15: Print the Nieged newm.

#### Doubly linked list Operation

Step 1: Start

Step 7: Declare a stometime and related variables.

Stop 3. Declare from etions to except.

Node, insert a mode in the head and headining, at the end and given position, display the list and search as element in the

Step 4: Define bunction to create a mode, declare the required arrivables.

Step 4.1: Set memory allocated to
the mode = temp.

then set temp > prev= null
and lemp > next = null.

- step 4.7: Rend the value to be insuted to the mode.
  - step 4.8: Set temp is n = date and decrement count by 1.
  - estep : Read the choice booms the want to perform abberrant operations on the list.
  - step is the the user chouse to perform the investions expection at the leginning them call the function to perform the investion.
  - Step 6.1: wheele of head = = null.

    then call the foundion

    to execute a mode, perform

    step 4 and 4.3.
  - Step 5. 2 3et head = terrip and terrip = head.
  - step 5. 3 : Else call the function

    to create a mode:

    perform step 4 tar 4.3

    then set temp -> next-sheed,

set head a prome temp and head a temp. of the users chaire is to neithern 51 ap 7 innertion at the end of the list, then is! the function to perform the inscrtim at the End. : Check if head = = null Stop Yel then Call the function to Oreate a newnode then eset temp a hoad and then set head = templ. : Else call the burneture to Stop 7 2 creat a new node then ext temp 1 -> next = temp, temp -> pren = templ and temp! = temp. : If the user chouse to Step 4 pargum insertion in the test At any position than call the bunction to perform the inscrition operation.

set head a prome temp and head a temp. of the users chaire is to neithern 51 ap 7 innertion at the end of the list, then is! the function to perform the inscrtim at the End. : Check if head = = null Stop Yel then Call the function to Oreate a newnode then eset temp a hoad and then set head = templ. : Else call the burneture to Stop 7 2 creat a new node then ext temp 1 -> next = temp, temp -> pren = templ and temp! = temp. : If the user chouse to Step 4 pargum insertion in the test At any position than call the bunction to perform the inscrition operation.

set head a prome temp and head a temp. of the users chaire is to neithern Stop 7 incretion at the end of the list, then call the function to perform the inscetion at the End. : Check if head = = nell Step Y.1 then Call the function to Oscate a newmode then eset temp a hoad and then set head = templ. : Else call the burnetum to Stop 7 2 create a new node them ext temp 1 > next = temp, temp -> pren = templ and temp! = temp. ; If the user chouse to Step 4 parform insertim in the test all my position then call the bunction to perform the inscrtico operation.

". Declare the mecentary ratiable : Read the partition where the 5+ sp 8. 2 node and need to the inserted, act temp or should. : cheek it pos el un post= Stop 8. 3 court +1 . Her mont the position is out of range. : check if hoad = = null and Step 8.4 pres - 1. there print "Enorthy list Connot inscrit and other other than 1st position. ; check if head = = vivil. 3 top x-5 and pos = 1. then call the brunetim te create neumade, then set themp = head and head = -temp! : While i'm pos then set Step 8. 6 tomp & = tempt of hext then inercoment i by 1. : Call the buneton to create stop 8. 7 a new mode and then set temp > pier = temp?

temp of next = temp of of next -> prim = temp. temp 2 -> next = temp. Step 9: If the uses chause to perprens deletion operation is the list then all the Gundlins to perform the deletion operation. Hep 9.1; Detlace the necessary voliable step 97; Read the position where node need to be delotted set temp & = head. Step 93: wheek if pos c 1 or pos >= count + 1. then point position out of singe. : wheek if head = = null Stop 7. 4 then point the list is empty , : while i'c pos then temp? Step 7-5 = -long & -> next and

increment i by 1.

temp => next = temp => next -> prim = temp. temp => next = temp.

stop 9: If the uses choose to present ion is prepared deletion operation is the list then all the function to perform the perform the deletion operation.

Stop 9.1: Declare the necessary variable

step 9.7: Read the position where node need to be delotted set temp & = head.

Step 93: wheek if pes all as pes? =

count + 1. then point position

out of senge.

step 7. 4: wheek if head = = null then point the list is empty.

Step 7.5: While i'c pos then temp?

= -temp = > next and
increment i by 1.

Stop 9.6: check if i==1 then check

16 temp 2 -> next == hull

then point node deleted to be

(temp 2) set temp 3 = head=

rull.

stop 9.7: which if temp & s mexit

= = rull then temp = = =

print > next = null.

then, gree (temps) then

Point node deleted.

Step 9.8: temp 2 -> next -> prev = temp?

-> prev. them check

if 1! = 1 then temp a

-> prev > next = temp? -> next.

step 9.9 : wheels if i = -1 than thead

= temp a > heat. then point

node deleted then free

temp a and decrement

count by 1.

Step to: If the user chause to newpen the display operation the display operation to display the function

```
set temp a = n.
Step 10.1 :
step 10. 2: Chesk to + emp 2 = rull
              then point list is employ
               While temp & = next = null
Step 10.3 :
               then print temp a >n
               then temp = temp2-next.
            . If the user charge to
Step 11
              purposen the search operation
              them call the function
              to perform reach operation
             : Deplace the necessary validate.
Step 11 - 1
            : Set temp 2 = hord.
5tep 11, 2
             : wheek if temp 2 = = nuit
est-ep 11. 3
              then point the dist is
               empty.
             : Read the value to be
strp 11. 4
              Gearched.
St ep 11.5
             : While temp al = null.
                the the ek of temporan
                = = date them point
               element found at position
                count +1.
```

Step 11. 6: Else 4et temp a = temp 2.3

next and increment countly 1.

Step 11. 7: Print relement not form?

in the list

Step 19: End.

#### Set Guerations

Step 1 : Slast.

step ? : Delace the necessary resideles.

step 3 : Read the choice bronn the

user to purposer. eset operation

stop 4; if the orser chause to perform

union:

step \$ 1 : He were Read the oridinality

Step 4.7: Wheate is m, - n then print cannot pregum union.

step 4.3: Else road the elements in both the sets:

step 4.4; Repeat the step 4.5 to 4.7 until fcm.

stop 4.5: C[i] = 4[i] (B[i]

Step 4.6; Print (C)

```
Step 4.7: Increment i dig 1.
Atop 9: Read the charge from the errer
           to perform investion.
step 5.1: Read the cardinality of sects
Step 3. 7: chieft if not = n there print
         Connect perpense intexsection
Step 3.3: Else read the Elements is both
         the sets
step 5.4: Repeat the step 5.5 to 5.7
            until izm.
             ([i] = n (i) 4 B [i]
5tep 5.5
          : baint c (1).
step 5.6
           : Into ownerst i toy 1.
Step 5.7
           : If the user chouse to perform
Step &
            set difference operation.
            : Read. the condinality of
step 6.1
            sets.
           : where if mil = in their
step 6-2
             print can not perform
             set difference operation.
```

Step 6.3: Else orad the element in herth sets Step 6.4: Repart the step 6.5 +06-8 Until ich. Step bot : which if A [1] = = 0. then ([7] = 12. Step 6.6 : Else 4 13 a) = -1 then (1)=0. Step 6-7 : Else [1] = 1 Step 6.8 : Incoment i day 1. : Repeat the step 7.1 and 7.2 51 up 7. Until 1 4 mm. : print ( (1). Step 7.1 Step 7.2: Incomment i day 1.

## Binary Brazeh Tree

Step 1 : Start.

Step 3: Declare a structure and
Structure printers for insertions
deletion and seach operations
and also declare a function
for inorder traversal.

step 3: Declare a pointer as sout and also the required reciable.

step 4: Read the chaire homes the user to perform investion, deletion, reaching and shooter toxicisely

step 5 : If the over chouse to pulsions insertion operation then read the value which is to be incerted to the tree booms

Step 5.1: pacs the value to the insect pointer

Step 5. 7: cheek if I suot then allecte memory for the sout

Step 5. 3: Set the value to the info that of the root and then set left and eight part of the noot to null and rather

Step 5.4; which is not is into 2x thin call the insect pointer to insect to left of the out

Step 5.5 : wheele it suot -> info or.

Then call the invert pointer

to the sight of

the snot.

Stop 5.6; Retiren the svot.

Hep 6: If the creek charge to prepare deletion prelation then read the element to be deleted from the

tree pass the root pointer and the item to the delete pointer

Step 6.1: check it not play them point node node bound.

step 6. 2: Else if ptr > into ex the call dolete pointer by pairing the Ett pointer and the item.

Step 6. 3: she by pursing the right

6. 4: wheele is ptr s indo == items.

Their check is ptr => lest

= = ptr -> sight them

bree ptr and s starm well.

step 6.5 : Else is ptr -> lebt = = rull.

set p1. ptr -> sight and
bree ptr, rolinen p1.

step 6.6: 4 lse 16 ptr sight == rull

set P1 = ptr > telt

and gove ptr, return P1.

step 6.7; Else set PI - ptr -> sight end pa= ptv -> sight. stop 0.8: While PI + left not Equal to rull, set. Pl 3 lebt. Mr -> left and free Mr. return Dr. Step 6-9 : Retnem ptr. Step 7 : If the inex choirse to preparent search opination the call the painter to prestring : Declare the necessary Step 7 . 1 pointers and variables. ; Read the Element to step 7 . 7 be surched. step 7. 3: white ptr sheet 15 itum > ptr > into. then ptr = ptr > sight. Step 7: 4; Else if item < ptr >info

then ptr = ptr -> lebt.

step 7 5: Else break.

that the rement is bound.

step 7.7 : Else point Element nat found in tree and

return ourt.

Step \$ ... If the wer charge to perform toinversal then call the tearrical bundies

Step 8.1 : If not not Equals to

hull receivesively call the bundlen try parring

svot -> lept

Step 8. 7: Print suot -> luto.

step &. 3 : Call the transversal functions
conversally by pairing
suot > right.