| # Lab Program - 10 #include attain h> #include attain hode "(eft #include and "(eft #include and "(eft #include and "(eft #include and "(eft #include attain #include attain | | |
|---|---------|--|
| #include https://www.node wit data; Atout node * right; Atout node * right; Atout node * tump; buint (Futu Root dement; 'n'); temp = (Atout node) malloc (rize of (Atout node)); temp = (Atout node) malloc (rize of (Atout node)); temp > left = temp > right = Nove; subturn temp; void ment (Street node * note > data) if (temp > data < note > data) if (temp > data > note > data) if (temp > data > note > data) if (temp > data > note > data) | | Lab PROGRAM - 10 |
| Findede | | LIVE TO STANK TO STANK THE |
| Findede | | Hinclude cottées.h |
| wit data; Atout node * left; Atout node * reate() Atout node * temp; Limit node * temp > data; Limit node * temp > light = NULL; Limit node * temp; Liftenp > data < not > data; Liftenp > data < not > data; Liftenp > data > noot -> data; | | Findude estalib.h> |
| Atout node * right; struct node * weate () Arout node * warp; paint (Futu Root element; nr); temp = (Arout node) malloc (vice) (Atrut node); temp > data (, temp > data); temp > telt = temp > hight = Nuce; which temp > data < noot > data) if (temp > data < noot > left != nuce) liment (noot > left != temp; else noot > left = temp; if (temp > data > noot -> data) | | mut node |
| Atout node * right; struct node * weate () Arout node * warp; paint (Futu Root element; nr); temp = (Arout node) malloc (vice) (Atrut node); temp > data (, temp > data); temp > telt = temp > hight = Nuce; which temp > data < noot > data) if (temp > data < noot > left != nuce) liment (noot > left != temp; else noot > left = temp; if (temp > data > noot -> data) | | t in the state of |
| Arrut node * right; Arrut node * weath () Arrut node * tump; paint ("futu Root dement; n"); temp = [strut node) malloc (size of (strut node)); tamp > left = tump > right = N ble; Autum temp; Li (temp > data < not => data) il (temp > data < not => data) il (noot > left != NVLL) linearly node > left ! tump; linearly left > left ! tump; linearly left > left ! tump; y if (temp > data > noot -> data) | | the data |
| struct node * weate () Atruct node * tump; bainty ("futu Root element; n"); temp = (struct node) malloc (size of (struct node)) trump > left = tump > sight = NULL; subtrum temp; word much (struct node * noot, struct node * tump) if (temp > data < noot => data) if (noot > left! = NULL)) insunct (noot > left! = NULL) le noot > left! = tump; y Leturp > data > noot -> data) | | shut made the |
| Atreat node * weate () Atreat node * temp; brinty ("futu Root dement: n"); temp = [Atreat*node) mode (size of (Atreat node)); team ("% of", ftemp > defect; temp > left = temp > right = NULL; steam temp; wid ment (Atreat node * noot, struct node * temp) if (temp > data < noot => data) if (noot > left! = NULL) if (noot > left! = temp; class of temp > data > noot -> data) if (temp > data > noot -> data) | | 4 rooti; |
| About hode * timp; brint ("Futu Root Ulment; n"); temp = (About node) mallor (Lize of (About node)); trans to lod", I temp > data); tump > telt = temp > night = Nove; sutum temp; void inent (About node * noot = data) if (temp > data < noot = data) if (noot > left! = Nove) humut (noot > left; tump); else y noot > left = temp; y (femp > data > noot -> data) | | |
| About hode * timp; brint ("Futu Root Ulment; n"); temp = (About node) mallor (rize of (About node)); trans (10/0 d ', ftemp > data); trus > telt = temp > night = Nove; suturn temp; void inent (About node * noot > data) if (temp > data < noot > left! = Nove); liment (noot > left! = temp; y noot > left = temp; y if (temp > data > noot -> data) | | struct node* weat () |
| temp = [Arut Root dement: nr); temp = [Arut Root] malloc (rize of (Atrut Rode)); tam (| . (| . Care 2: injustic black . |
| temp = [struct*node) malloc (size of (struct node)] leant (10/0 d), fromp -> data); temp > left = temp > sight = Nove; subtrum temp; void must (struct node * moot, struct node * temp) if (temp -> data < moot -> data) if (noot -> left! = Nove) in temp -> data > noot -> data) if (temp -> data > noot -> data) | | mut hade * timp; |
| woid ment (street node * woot, street node * temp) if (temp > data < noot > data) if (noot > left! = NULL) immet (noot > left, temp); else noot > left = temp; y if (temp > data > noot -> data) | | punt (futu koot ulment: 'n"); |
| woid ment (street node * woot, street node * temp) if (temp > data < noot > data) if (noot > left! = NULL) immet (noot > left, temp); else noot > left = temp; y if (temp > data > noot -> data) | | draw from hole malloc (fixed (struct nod)) |
| word went (street node * woot, street node * temp) if (temp -> data < wot => data) if (woot -> left!= were) proset -> left , temp); else voot -> left = temp; y if (temp -> data > voot -> data) | bille | tumb > left = temp > water) |
| void much (street node * mot, street node * temp) if (temp > data < noot > data) if (noot > left!= noce) in much (noot > left, temp); else noot > left = temp; y if (temp > data > noot -> data) | fruita" | Litum temp: |
| if (temp -> data < noot -> data) if (noot -> left != nous) in (noot -> left != nous); else noot -> left = temp; y (temp -> data > noot -> data) | | y v m |
| if (temp -> data < noot -> data) if (noot -> left != nous) innert(noot -> left, temp); else noot -> left = temp; y (temp -> data > noot -> data) | | in it is a first with they of |
| if (not -> left != NULL)) insert noot -> left , temp); clae noot -> left != temp; y if (temp -> data > noot -> data) | | s west street node * woot, street node * temp) |
| if (not -> left!=NULL)) insert (noot -> left, temp); else noot -> left != temp; y if (temp -> data > noot -> data) | | il (forms > data (in Av. 11) |
| lle noot -> left, temp); lle noot -> left, temp); y (temp -> data > root -> data) | | form = and = mol-s data) |
| lle noot -> left, temp); lle noot -> left, temp); y (temp -> data > root -> data) | | if (not -> left!=NUW) |
| if (temp -> data > root -> data) | | must root -> left, temp); |
| if (temp -> data > root -> data) | | Ulse |
| | | root → left = temp; |
| | | il Ctour Coto Supet stall |
| Illanot -> esalt 1 min 1) | | |
| I ALIVOU / MUML != MUVU | | U[Noot → right!= NULL) |
| injut (400t -> right; tems); | | inky (Hoot -> right; tems); |
| Un Un | | eln |
| Scanned with CamScanner | | |

| 07 |
|--|
| noot -> night = temp; |
| 7 |
| V STEW WINE GOLDSWA |
| The state of the s |
| void printPortorder (struct rode * node) |
| MOLA DIVINCTANOVIEW CNIVERS |
| if (node = 2 NULL) I divers |
| The water of the state of the s |
| fræstfæst brintfostorder (node -> left); brintfostorder (node -> bright); brintf ("%d\t", node -> dala); |
| huntputorder (node -) night); |
| by in H ("% 1) + " node -> data); |
| $ \mathbf{V} = (1 - 7) \cdot (1 $ |
| void printénoiser (street no de * node) |
| The short of endoy what I through a mo |
| moderation with the service of the s |
| return; |
| print Inorder (node -slift); |
| buint (3% dt minode data); |
| print morder (node - right) |
| g (1 toas) reprositiving of mas |
| = Lox No |
| wid bunt morder (strut node * node) |
| (au S: builterfor la (nort) bull |
| I (node vi= ZINIVCIC) Listed of mos |
| Dulin; died |
| print (" (od t" node - s data); |
| mint morder (node -> left); |
| sunt Puorder (node - right) jelihar |
| 13 Camusas |
| · · · · · · · · · · · · · · · · · · · |
| int main () |
| \mathcal{E} |
| int droite; |
| frut nodi * temp; |
| do |
| |

| c=== | |
|------|--|
| | buints (" \n MENO "); |
| | blints (" 1. CREate (n"); |
| | bright (" 2. Insect \n") |
| | printe (" tras. Preorde Traversal nn); |
| | bungling. Thorder Traversal son; |
| | bruhl ["5. Postordu Traverial mi); |
| | build In Friting; |
| | brink " Enter your choice correcter |
| | 100 , 2000(C) |
| | muital (choice) to travel |
| | (idib & ban 15/68) I daily |
| | (an 1: Nort = clasic reale(); |
| | when show bheatigh reproved my bing |
| | can 2: print ("Enter Value to Print in): |
| | Temp = 1 Mut node t) mallo c (min of |
| | MITH WITH WALL |
| | leany ("6d", & temp - data); |
| | ment (Moot), temp) |
| | (desir break je on) representationed |
| | can 3: punthuorder (root) |
| | an 4: print morder (root); break: |
| | |
| | can 5: printporter (roots); break; |
| | buak: (Auda) |
| | about : punt ["Incorrect etroice! in); |
| | View 1 1 701 ~ View 1 1 1 1 November 1 |
| | I while (choice != b);) whroughted |
| | returno; |
| | y . |
| | Int main () |
| | <u> </u> |
| | Total Chi |
| | junet * Dan turk |
| | ab |
| | <i>3</i> |