Eugenean 10 let wichede 2 etdis. h & michaele ( stalib h) typedy thuck Node ! street Node + left; cuit dater; struct Node \* myld; 3 + rude; usde geturde ( int item) & ande toup " ( ender cuallor ( eject ( dunch unde )); laup -> left = NULL; Comp -> data = ilem; Gemp -> sught = NVLL) nelvan temp; unche mideral ( unde crest, crist ele/ & if I west > > NOLL/ return get esde (de); else if ( ele & that -) dara/

resport -> regert = resport -> left, ele);

else if (ele > resport -> elacal

resport -> regert = rejert (resot -> rejert, del);

relian resot;

usid hierder (mod mot) 
if ( reset = = NULL!
heturn;
imporder (mod -> let

puint (" " od", mot -> datal;

Com 1 J curredur (rest -> augus); 3 U 0 lesid presider ( usele most / h V of ( root = = NOZZ) W) relin ; formily (" 0/od" rust 3 dolin); Johnson ( visot -> left ); presider ( rost -> rejla ); 000 Tossid postprder ( ensde visst / h if ( Just == NULL) return; food order ( rost -> loft /; post order ( rost -> right /; printy (" "/~ of", rost > data); wit main 1/ L custe vort > pull; uit e, ch :1; well ( Th! = 5/ & purity ("n/n/. Just 12. Preseder In 3. Enorder 29. Apstorder \ m"/; print ("5. Evit 1 my) frant ( " % d", ( ch); print ("/m"/> purlich (ch) ? caul: plinif (" Element:"1; sceny (" % d" & c/)
rost = light ( rost, c/) breess;

1

Case 2; poweder (root);

bureak;

(ase 3: worder (root);

preak;

cose 4: post order (root);

preak;

cose 5: point ("Europe");

cose 5: point ("Europe");

cose 5: point ("Europe");

cose 6: point ("Europe");

cose 6: point ("Europe");

cose 7: point ("Europe");

cose 6: point ("Europe");

cose 7: point ("Europe");

cose 6: point ("Europe");

cose 7: point ("Europe");

cose 7: point ("Europe");

cose 8: point ("Europe");

cose 8: point ("Europe");

cose 8: point ("Europe");

cose 9: point ("Europe");