(lab-10) Binary Search tree # include c stdio. h # includec process.h struct node \* rlink; Struct node " llink; (708) mid typeday struct node \* NODE; soon NOD TRO AL NOOF gernede () & NODEN; HIPTON ATT NO SOUNDER TO PROPERTY X = (NODE) malloc (size of (struct node)); if (N==NULL) 2 prints ("mem fullm"); (x1+(0); return 11;

```
NODE insert (NODE root, int item)
d NODE temp, prev, cur;
 temp= getnode();
 temp ->rlink == NU'LL; / / wedon "mble" / fill
  temp -> Ilink = NULL; (daile too)
  temps info = item; (daily adoct) is in
 if (root = = MULL)
  return temp;
 prev = NULL.
 cur = root;
while (cu: = * NUL)
d prev = cur;
 cor= (item c cor sinfo)? cor slink: corstink;
if (item < prev > info) poor 20000) and retain book
  prev > llink = temp;
  else
   pred-relink = temp;
  return good;
 Void display (NODE root, inti)
d int is
 if (root!= NULL)
 d display (root-raink, i+1);
   for (j=0; j 21, j++) $
     Brints (" ");
    printy (" 1.d lu", rost > info;
   a display (not -> 11ink, i+1);
```

```
void preorder (NODE root)
 if (100+1= xULL)
 1 prints (".1-9 in", not > sufo);
  preorder (root-sllink);
  preorder (root) rlink),
 Void inorder (NODE
 1 if (root!= NULL)
d inorder (root silink);
    print (".tdin", roots info);
    printy (root > rhink); 100 ? (price 100 > moti) - 10
  void postorder (NODE root) (africano)
of if (woot) > NAM)
                             ignot sight - on
 of printflata
   postorder (root > llink);
   postorder (noot=) rlink);
   printy ("laln", roots info); or accord polyxil
 void main()
I int item, choice;
 NODE robt = NULL!
  for (ii)
d prints(" |n 1: Insert In 2: Display In 3: pre-orde
          4: Inorder | n 5: 1 Postors Scanned with CamScanner
```

```
Scarf (".1-d/n', & choice);
switch (choice)
I case 1: prints ("Enter the item (m");
     Scanf ("Id", & item);
   root = insert (root, item);
   break',
case 2 & display (root, o);
     break ,
 Case 3: preorder (root);
      break',
  case y: Inorder (root);
    break',
  Case 5: postorder (root);
       break',
   Ott Ex
    default : exit (0);
         break ,
```