Date

Expt. No.   Page No.

BINARY SEARCH TREE

```
# include<stdio.h> .
# include <stdlib.h> .

struct  node

{ int  info;
   struct node *rlink;
   struct  node *llink;
   };

typedef  struct node  *NODE;

NODE  getnode()

{ NODE  x;
x = (NODE) malloc (sizeof (struct  node));

if  x(== NULL)

{ printf (" memmemory full \n");

   exit (0); }

  return  x ; }
```

Teacher's Signature :

```
void freenode (NODE x)
{ free(x) ; }

NODE insert ( NODE root, int item)
{ NODE temp , curr , prev ;
  temp = getnode() ;
  temp -> rlink = NULL ;
  temp -> llink = NULL ;
  temp -> info = item ;
  if (root = = NULL)
    return temp ;
  prev = NULL ;
   curr = root ;
 while (curr != NULL)
 { prev = curr ;
   curr = ( item < curr -> info) ? curr -> llink : curr -> rlink ;
  }
```

Date ....

Expt. No. .... Page No. ....

```
if (item < prev->info)
    prev->llink = temp;
else
prev->rlink = temp ;
 return root ; }
void display (NODE root , int i)
{ int j;
if (root != NULL)          if(root != NULL)
{ display (root->rlink , i+1);
  for (j = 0; j<i ; j++)
      printf (" ");
  printf ("%d", root->info ;
  display (root>llink, i+1);
  }  }
```

Teacher's Signature : ____________

Date ....

Expt. No. ....   Page No. ....

```
NODE delete (NODE root, int item)
{ NODE cur, parent, q, suc;
if (root == NULL)
  { printf ("empty\n");
    return root; }
parent = NULL;
cur = root;
while (cur != NULL && item != cur -> info)
{ parent = cur;
cur = (item < cur -> info) ? cur -> llink : cur -> rlink;
}
if (cur == NULL)
{ printf ("not found\n");
  return root; }
if (cur -> llink == NULL)
    q = cur -> rlink;
```

Teacher's Signature : ____________

Date

Expt. No.   Page No.

```
void preorder (NODE root)
{ if(root != NULL)
  { printf ("%d \n", root -> info );
     preoorder (root -> llink );
     preorder (root -> rlink
                                 )
  }
 void  postorder (NODE root )
{ if (root != NULL)
 { postorder(root ->llink);
   postorder (root ->rlink);
   printf  ( "%d \n", root ->info);
  }}
void inorder (NODE root)
{
    if (root != NULL)
{  inorder (root -> llink);
     printf ("%d \n, root ->info);
      inorder(root -> rlink); }
                               }
```

Teacher's Signature :

Date ....

Expt. No. .... Page No. ....

```
void main ()
{ int item, choice;
  NODE root = NULL;
  clrscr();

for ( ;;)
printf ("\n 1. insert \n 2. display \n 3. pre \n 4. post
        \n 5. in \n6. delete \n7. exit \n");

printf ("Enter a choice \n");

scanf ("%d", &choice);

switch (choice)

{ case 1 : printf ("enter the item \n");
    scanf ("%d", &item

  root = insert (root, item);
    break;

case 2 : display (root, 0);
           break;

case 3 : ~~pos~~ preorder (root);
           break;
```

Teacher's Signature :

Date

Expt. No. Page No.

```
case 4 : postorder(root);
         break;

case 5 : inorder (root);
         break;
case 6 : printf ("enter item \n");
         scanf (" %d , &item );
         root = delete ( root , item);
         break;
   default : exit (0);
         break; } }

  }
```

Teacher's Signature :