#include<stdio.h>

#include<stdlib.h>

**struct** node

1. {
2. **int** key;
3. **struct** node \*left;
4. **struct** node \*right;
5. };
7. //return a new node with the given value
8. **struct** node \*getNode(**int** val)
9. {
10. **struct** node \*newNode;
12. newNode = malloc(**sizeof**(**struct** node));
14. newNode->key   = val;
15. newNode->left  = NULL;
16. newNode->right = NULL;
18. **return** newNode;
19. }
21. //inserts nodes in the binary search tree
22. **struct** node \*insertNode(**struct** node \*root, **int** val)
23. {
24. **if**(root == NULL)
25. **return** getNode(val);
27. **if**(root->key < val)
28. root->right = insertNode(root->right,val);
30. **if**(root->key > val)
31. root->left = insertNode(root->left,val);
33. **return** root;
34. }
36. //inorder traversal of the binary search tree
37. **void** inorder(**struct** node \*root)
38. {
39. **if**(root == NULL)
40. **return**;
42. //traverse the left subtree
43. inorder(root->left);
45. //visit the root
46. printf("%d ",root->key);
48. //traverse the right subtree
49. inorder(root->right);
50. }
52. **int** main()
53. {
54. **struct** node \*root = NULL;

57. **int** data;
58. **char** ch;
59. /\*  Do while loop to display various options to select from to decide the input  \*/
60. **do**
61. {
62. printf("\nSelect one of the operations::");
63. printf("\n1. To insert a new node in the Binary Tree");
64. printf("\n2. To display the nodes of the Binary Tree(via Inorder Traversal).\n");
66. **int** choice;
67. scanf("%d",&choice);
68. **switch** (choice)
69. {
70. **case** 1 :
71. printf("\nEnter the value to be inserted\n");
72. scanf("%d",&data);
73. root = insertNode(root,data);
74. **break**;
75. **case** 2 :
76. printf("\nInorder Traversal of the Binary Tree::\n");
77. inorder(root);
78. **break**;
79. **default** :
80. printf("Wrong Entry\n");
81. **break**;
82. }
84. printf("\nDo you want to continue (Type y or n)\n");
85. scanf(" %c",&ch);
86. } **while** (ch == 'Y'|| ch == 'y');
88. **return** 0;
89. }

