

PANASA TEJA

DBMS LAB EX: 7

```
user@jack:~/Documents/Sem 5/DBMS_LAB/assg 6$ gcc b-tree.c -o b-tree&&./b-tree
1. Insertion 2. Deletion Enter your choice:1
3. Searching 4. Traversal", &ch);
5. Exit switch (ch) {
Enter your choice:1 case 1:
Enter your input:1 printf("Enter your input:");
scanf("%d", &val);
insertion(val);
break;
case 2:
Enter your choice:1 printf("Enter the element to delete:");
Enter your input:3 scanf("%d", &val);
deletion(val, root);
break;
case 3:
Enter your choice:1 printf("Enter the element to search:");
Enter your input:5 scanf("%d", &val);
searching(val, &ch, root);
break;
case 4:
Enter your choice:1 traversal(root);
Enter your input:6 break;
case 5:
Enter your choice:1 exit(0);
Enter your input:7
default:
Enter your choice:1 printf("U have entered wrong option!!\n");
Enter your input:8 break;
Enter your choice:1 }
Enter your input:9 printf("\n");
Enter your choice:1
```

```
Enter your choice:1 / \
Enter your input:6 printf("Enter your choice:");
scanf("%d", &ch);
Enter your choice:1 switch (ch) {
Enter your input:7 case 1:
printf("Enter your input:");
scanf("%d", &val);
insertion(val);
break;
case 2:
Enter your choice:1 printf("Enter the element to delete:");
Enter your input:9 scanf("%d", &val);
deletion(val, root);
break;
case 3:
Enter your choice:3 printf("Enter the element to search:");
Enter the element to search:4 scanf("%d", &val);
Given data 4 is present in B-Tree searching(val, &ch, root);
Enter your choice:3 break;
Enter the element to search:11 case 4:
traversal(root);
break;
case 5:
Enter your choice:2 exit(0);
Enter the element to delete:4
default:
Enter your choice:2 printf("U have entered wrong option!!\n");
Enter the element to delete:5 break;
Enter your choice:4 }
1 2 3 6 7 8 9 10 printf("\n");
Enter your choice:2
Enter the element to delete:6
```

```
Activities Applications Terminal Tue Oct 12 1:58:33 PM
user@jack:~/Documents/Sem 5/DBMS_LAB/assg 6$ gcc b-tree.c -o b-tree&&./b-tree
1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:25
1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:26
1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:3
Enter the element to search:24
1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:2
Enter the element to delete:6
Given value is not present in B-Tree
1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:4
25 26
1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
```

```
Activities Applications Terminal Tue Oct 12 2:01:00 PM
user@jack:~/Documents/Sem 5/DBMS_LAB/assg 6

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:11

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:22

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:33

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:44

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:55

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
```

Consider a database for storing the details of students currently pursuing courses in National Institute of Technology Calicut. The database maintains the lists of students opting each elective course. The details of students taking an elective are recorded in a B-Tree with student roll number as the index. The B-Tree has the following functionalities:

1. INSERT – A new record (i.e., roll number) should be inserted
2. DELETE – The specified record should be deleted
3. SEARCH – If the record is present in the tree, return TRUE. Else, return FALSE
4. PRINT – All the records currently present in the tree should be displayed

Considering the order of the tree to be 4, implement the B-Tree with above functionalities. Note that you may use C/C++ for implementing the B-Tree. The following should be included in a single zip file for submission:

1. Source code for implementing the B-Tree (.c/.cpp file)
2. A document (.pdf) consisting of:
 - a. Screenshots of the output terminal obtained on running the program. Note that three runs of the implementation should be performed, with the first run having insertion of 10 records, the second run having insertion of 15 records, and the third run having insertion of 20 records. After all the required insertions have been performed, display the current records present in the resultant B-Tree. Each run should involve two deletions. After each deletion, the records currently present in the B-Tree should be displayed. Also, two search operations should be there in each run, with one for searching an item already present in the B-Tree, and the other for searching an item that is not present in the B-Tree.
 - b. Pictorial representations of the B-Trees obtained after the required insertions (i.e., after 10 insertions in the first run, 15 insertions in the second run, and 20 insertions in the third run) and each deletion in all the three runs.

If any doubts on the exercises, please mail us
(prabum@nitc.ac.in).

```
Activities Applications Terminal
user@jack: ~/Documents/Sem 5/DBMS_LAB/assg 6
Tue Oct 12 2:01:07 PM

3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:444

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:555

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:3
Enter the element to search:555
Given data 555 is present in B-Tree
1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:2
Enter the element to delete:555

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:4
11 22 33 44 55 66 77 88 99 110 111 222 333 444

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:|
```

Consider a database for storing the details of students currently pursuing courses in National Institute of Technology Calicut. The database maintains the lists of students opting each elective course. The details of students taking an elective are recorded in a B-Tree with student roll number as the index. The B-Tree has the following functionalities:

1. INSERT – A new record (i.e., roll number) should be inserted
2. DELETE – The specified record should be deleted
3. SEARCH – If the record is present in the tree, return TRUE. Else, return FALSE
4. PRINT – All the records currently present in the tree should be displayed

Considering the order of the tree to be 4, implement the B-Tree with above functionalities. Note that you may use C/C++ for implementing the B-Tree. The following should be included in a single zip file for submission:

1. Source code for implementing the B-Tree (.c/.cpp file)
2. A document (.pdf) consisting of:
 - a. Screenshots of the output terminal obtained on running the program. Note that three runs of the implementation should be performed, with the first run having insertion of 10 records, the second run having insertion of 15 records, and the third run having insertion of 20 records. After all the required insertions have been performed, display the current records present in the resultant B-Tree. Each run should involve two deletions. After each deletion, the records currently present in the B-Tree should be displayed. Also, two search operations should be there in each run, with one for searching an item already present in the B-Tree, and the other for searching an item that is not present in the B-Tree.
 - b. Pictorial representations of the B-Trees obtained after the required insertions (i.e., after 10 insertions in the first run, 15 insertions in the second run, and 20 insertions in the third run) and each deletion in all the three runs.

If any doubts on the exercises, please mail us
(prabum@nitc.ac.in).

```
Activities Applications Terminal
user@jack: ~/Documents/Sem 5/DBMS_LAB/assg 6
Tue Oct 12 2:01:04 PM

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:55

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:66

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:77

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:88

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
Enter your choice:1
Enter your input:99

1. Insertion 2. Deletion
3. Searching 4. Traversal
5. Exit
```

Consider a database for storing the details of students currently pursuing courses in National Institute of Technology Calicut. The database maintains the lists of students opting each elective course. The details of students taking an elective are recorded in a B-Tree with student roll number as the index. The B-Tree has the following functionalities:

1. INSERT – A new record (i.e., roll number) should be inserted
2. DELETE – The specified record should be deleted
3. SEARCH – If the record is present in the tree, return TRUE. Else, return FALSE
4. PRINT – All the records currently present in the tree should be displayed

Considering the order of the tree to be 4, implement the B-Tree with above functionalities. Note that you may use C/C++ for implementing the B-Tree. The following should be included in a single zip file for submission:

1. Source code for implementing the B-Tree (.c/.cpp file)
2. A document (.pdf) consisting of:
 - a. Screenshots of the output terminal obtained on running the program. Note that three runs of the implementation should be performed, with the first run having insertion of 10 records, the second run having insertion of 15 records, and the third run having insertion of 20 records. After all the required insertions have been performed, display the current records present in the resultant B-Tree. Each run should involve two deletions. After each deletion, the records currently present in the B-Tree should be displayed. Also, two search operations should be there in each run, with one for searching an item already present in the B-Tree, and the other for searching an item that is not present in the B-Tree.
 - b. Pictorial representations of the B-Trees obtained after the required insertions (i.e., after 10 insertions in the first run, 15 insertions in the second run, and 20 insertions in the third run) and each deletion in all the three runs.

If any doubts on the exercises, please mail us
(prabum@nitc.ac.in).