

PROJECT REPORT

NAME: SOUMYA JHA

REGISTRATION NUMBER: 20BCE0885

1. What is your role?

I made the entire project on my own. I have made a dictionary using binary tree.

2. What you learned while building the project?

I learnt how to make binary trees using C language and also learnt how they can be used to make a dictionary.

3. Did you take any courses like coursera, udemy...etc, while working for your project?

I referred to youtube videos to learn making dictionary using binary trees. I also have Data structures and algorithms in this semester. So that also helped me in making in this project.

4. Mention the libraries that you have installed for the project.

I just used Codeblocks to do all the coding in C.

5. Also explain the work flow of the project.

I just made different functions to carry out the four options that this dictionary does i.e search, insert, view and exit. It provides the user to choose their desired option which can easily be done using switch case. We have to insert words with their meanings. Initially the dictionary is just an empty dictionary.

GITHUB REPO LINK:

https://github.com/soumya0204/dictionary_using_binary_tree

6. Screen shots

```
Start here x dictionary.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct dict {
5      struct dict *left,*right;
6      char word[10],meaning[20];
7  }*Root=NULL;
8
9  typedef struct dict dictionary;
10 int check(char[],char[]);
11 void insert(dictionary *);
12 void Search();
13 void view(dictionary *);
14
15 int check(char a[],char b[]){
16     int i,j,c;
17     for(i=0,j=0 ; a[i]!='\0'&&b[j]!='\0' ; i++,j++){
18         if(a[i]>b[j]){
19             c=1;
20             break;
21         }
22         else if(b[j]>a[i]){
23             c=-1;
24             break;
25         }
26         else
27             c=0;
28     }
29     if(c==1)
30         return 1;
31     else if(c==-1)
32         return -1;
33     else
34         return 0;
35 }
36
37 void Search(){
38     int flag=0;
39     dictionary *ptr;
40     ptr=Root;
41     char w[10];
42     printf("\nEnter word");
43     scanf("%s",w);
44     while(ptr!=NULL && flag==0){
45         if(check(w,ptr->word)>0)
46             ptr=ptr->right;
47         else if(check(w,ptr->word)<0)
```

```
Start here x dictionary.c x
48         ptr=ptr->left;
49     else if (check(w, ptr->word)==0) {
50         flag=1;
51         printf("\n%s", ptr->meaning);
52     }
53
54     }
55     if(flag==0)
56         printf("\nWord not found");
57 }
58
59 void insert(dictionary *temp) {
60     int flag=0;
61     dictionary *ptr, *par;
62     ptr=Root;
63
64     if(Root==NULL)
65         Root=temp;
66     else{
67         while(ptr!=NULL ) {
68             if(check(temp->word, ptr->word)>0) {
69                 par=ptr;
70                 ptr=ptr->right;
71             }
72
73             else if(check(temp->word, ptr->word)<0)
74             {
75                 par=ptr;
76                 ptr=ptr->left;
77             }
78             else if(check(temp->word, ptr->word)==0) {
79                 flag=1;
80                 printf("\nWord exists!!");
81                 break;
82             }
83
84         }
85         if(flag==0 && ptr==NULL) {
86
87             if(check(par->word, temp->word)==1)
88                 par->left=temp;
89             else if(check(par->word, temp->word)==-1)
90                 par->right=temp;
91         }
92     }
93 }
94
```

```
Start here x dictionary.c x
93     }
94
95 }
96
97 void view(dictionary *ptr) {
98     if(Root==NULL)
99         printf("\nEmpty dictionary\n");
100
101     else if(ptr !=NULL) {
102         view(ptr->left);
103
104         printf("\nWord:%s\n",ptr->word);
105         printf("\nMeaning:%s\n",ptr->meaning);
106
107         view(ptr->right);
108     }
109
110 }
111
112 int main(int argc, char const *argv[]) {
113     int ch;
114     dictionary *temp;
115     while(ch!=4) {
116         printf("\n1.Search\n2.Insert\n3.View\n4.Exit\nYour choice please..");
117         scanf("%d",&ch);
118         switch (ch) {
119             case 1: Search();break;
120             case 2:
121                 temp=(dictionary*)malloc(sizeof(dictionary));
122                 temp->left=NULL;
123                 temp->right=NULL;
124                 printf("\nInsert word:\n");
125                 scanf("%s",temp->word);
126                 printf("\nInsert meaning:\n");
127                 scanf("%s",temp->meaning);
128                 insert(temp);
129                 break;
130             case 3:
131                 view(Root);
132                 break;
133             case 4:exit(0);
134         }
135     }
136     return 0;
137 }
138
```

```
1.Search
2.Insert
3.View
4.Exit
Your choice please..2
```

```
Insert word:
apple
```

```
Insert meaning:
fruit
```

```
1.Search
2.Insert
3.View
4.Exit
Your choice please..2
```

```
Insert word:
cat
```

```
Insert meaning:
animal
```

```
1.Search
2.Insert
3.View
4.Exit
Your choice please..2
```

```
Insert word:
council
```

```
Insert meaning:
board
```

```
1.Search
2.Insert
3.View
4.Exit
Your choice please..1
```

```
Enter wordcouncil
```

```
board
```

```
1.Search
2.Insert
3.View
4.Exit
Your choice please..3
```

```
Word:apple
```

```
Meaning:fruit
```

```
Word:cat
```

```
Meaning:animal
```

```
Word:council
```

```
Meaning:board
```

```
1.Search
2.Insert
3.View
4.Exit
Your choice please..4
```

```
Process returned 0 (0x0)    execution time : 33.897 s
```