DICTIONARY USING BST

Data Structure:

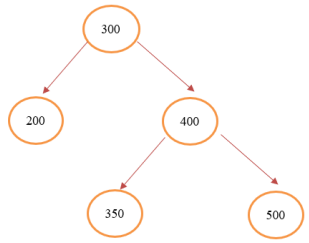
Binary Search Tree

It is a node-based binary tree data structure which has the following properties:

i. The left subtree of a node contains only nodes with keys lesser than the node's key.

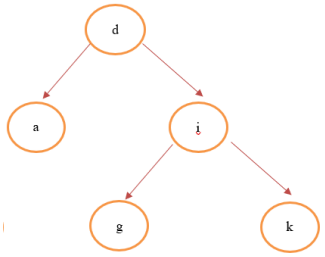
ii. The right subtree of a node contains only nodes with keys greater than the node's key.

Numbers

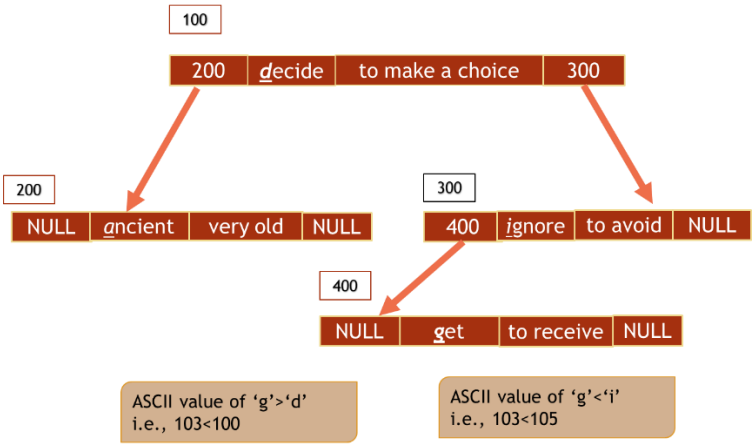


Root	300
Left node	200 (200 < 300)
Right node	400 (400 > 200)

Characters



Root	d
Left node	a ASCII(a < d)
Right node	i ASCII (i > d)



Modules:

- Insert - comparing the word to be inserted with the BST and insert
- Delete- search for the given word in BST and delete
- Search- traverse the BST, find the word and display the meaning
- Edit meaning- traverse the BST, find the word and edit the meaning
- View Dictionary-to display the BST using Inorder traversal

Functions:

strcasecmp()

It compares the two strings s1 and s2, ignoring the case of the characters. It returns an integer less than, equal to, or greater than zero if s1 is found, respectively, to be less than, to match, or be greater than s2.

WALKTHROUGH:

1) Structure Declaration:

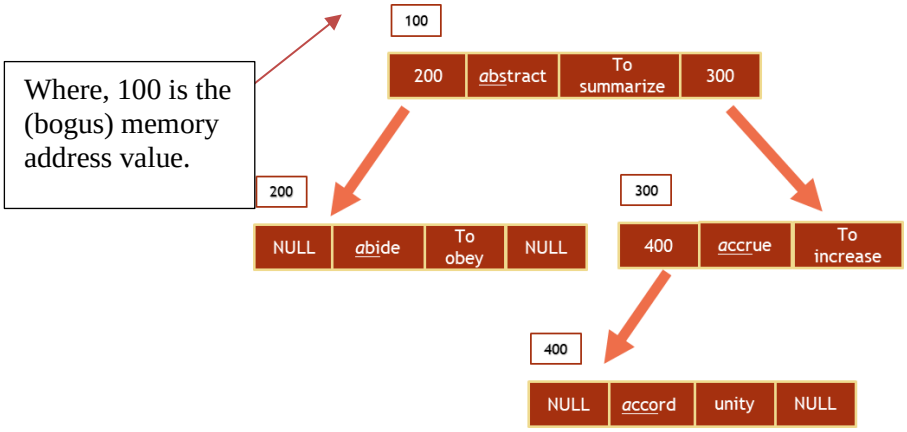
Below is the structure to implement BST.

```
struct dictionary
{
    char word[150], meaning[300];
    struct dictionary *left, *right;
}*root = NULL;
typedef struct dictionary dict;
```

This creates a structure like,

2) Insert Module:

1) If there is no root node, create and set this new node as root.
2) Traverse the tree, comparing the ascii values,
    i) If the ASCII value doesn't match with any of the nodes, print "Word exists".
    ii) If it's ASCII value is lesser than the current node, insert to its left.
    iii) If it's ASCII value is greater than the current node, insert to its right.



```c
dict * createNode(char *word, char *meaning)
{
        dict *newnode;
        newnode=(dict *)malloc(sizeof(dict));
        strcpy(newnode->word, word);
        strcpy(newnode->meaning, meaning);
        newnode->left = newnode->right = NULL;
        return newnode;
}

int insert(char *word, char *meaning)
{
    dict *par=NULL,*newnode=NULL;
    int res = 0;
    if (!root)
    {
        root=createNode(word, meaning);
    }
    else
    {
        dict *cur=root;
        while(cur!=NULL)
        {
            res=strcasecmp(word,cur->word);
            if(res==0)
             {   printf("Word already exists!!!\n");return 0;break;
             }
            par=cur;
            if(res>0)
                cur=cur->right;
            else
                cur=cur->left;
        }
        newnode=createNode(word, meaning);
        if(res>0)
            par->right=newnode;
        else
            par->left=newnode;
    }
    return 1;
}
```

Insert module Output:

INSERTING A WORD

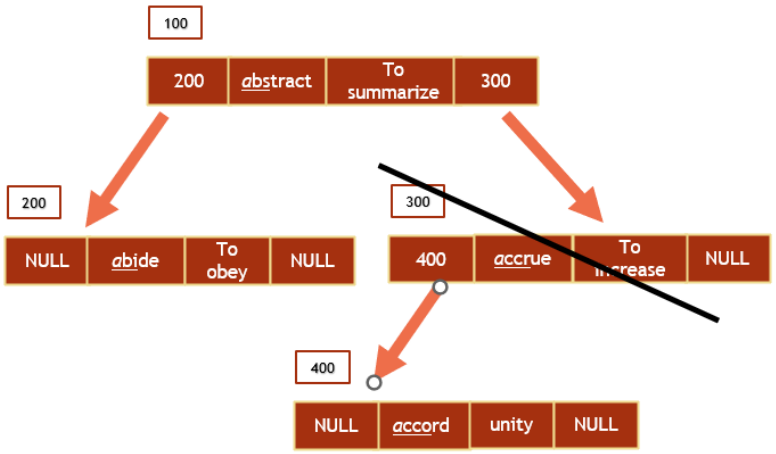
```
DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. Traversal
5. Edit         6. Exit
Enter your choice: 1
Enter the Word to insert with meaning:
baffle
to confuse
You've inserted the word "baffle" into the Dictionary

DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. Traversal
5. Edit         6. Exit
Enter your choice: 4
        Word                    :       Meaning
1       abide                   :       to bear patienly
2       abstract                :       to summarize; to epitomize
3       access                  :       gain, obtain
4       accord                  :       to be in agreement, harmony or unity
5       accrue                  :       to increase; to accumulate
6       aced                    :       to get the better of
7       adorn                   :       to make sth more beautiful; to decorate
8       affiliate               :       to trace origin to; to accept as a member
9       allay                   :       to calm, alleviate or relieve
10      augment                 :       to grow in size; intensify
11      backlog                 :       a build up of work, an accumulation or reserve
12      baffle                  :       to confuse
13      banter                  :       converse in playful or teasing way
14      benign                  :       kindly; good natured
15      bias                    :       influence in an unfair way
```

```
DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. Traversal
5. Edit         6. Exit
Enter your choice: 1
Enter the Word to insert with meaning:
accord
unity; harmony
Word already exists!!!
```

3) Delete Module:

1) If ASCII value of the search node matches with any of the node,
   check if it has a subtree/ node.
   i) If no, simply delete it.
   ii) If yes, and
      i) If the left node is null, make the pointer point to its right.
      ii) If the right node is null, make the pointer point to its left.
      iii) If both the nodes exist (not null), find the minimum between them and make the pointer point to it.
2) If it doesn't match with any node, the print "Word doesn't exist".




```
dict* find_minimum(dict *root)
{
    if(root == NULL)
        return NULL;
    else if(root->left != NULL)
        return find_minimum(root->left);
    return root;
}
dict* delete(dict *root, char* word)
{
    if(root==NULL)
    {    printf("The word \"%s\" is not found in the Dictionary\n",word);return NULL; }
    if (strcasecmp(word,root->word)>0)
        root->right = delete(root->right, word);
    else if(strcasecmp(word,root->word)<0)
        root->left= delete(root->left, word);
    else
    {
        if(root->left==NULL && root->right==NULL)
        {
            printf("You've deleted the word \"%s\" from the Dictionary\n",root->word);
            free(root);
            return NULL;
        }
        else if(root->left==NULL || root->right==NULL)
        {
            dict *temp;
            if(root->left==NULL)
            {
                temp = root->right;
                printf("You've deleted the word \"%s\" from the Dictionary\n",root->word);
            }
            else
            {
                temp = root->left;
                printf("You've deleted the word \"%s\" from the Dictionary\n",root->word);
            }
            free(root);
            return temp;
        }

        else
        {
            dict *temp = find_minimum(root->right);
            root = temp;
            root->right = delete(root->right, temp->word);
        }
    }
    return root;
}
```

Delete module Output:

```
DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. View Dictionary
5. Edit meaning 6. Exit
Enter your choice: 2
Enter the word to delete:
abide
You've deleted the word "abide" from the Dictionary

DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. View Dictionary
5. Edit meaning 6. Exit
Enter your choice: 4
        Word                    :       Meaning
1       abstract                :       to summarize; to epitomize
2       access                  :       gain, obtain
3       accord                  :       to be in agreement, harmony or unity
4       accrue                  :       to increase; to accumulate
5       aced                    :       to get the better of
6       adorn                   :       to make sth more beautiful; to decorate
7       affiliate               :       to trace origin to; to accept as a member
8       allay                   :       to calm, alleviate or relieve
9       augment                 :       to grow in size; intensify
10      backlog                 :       a build up of work, an accumulation or reserve
11      banter                  :       converse in playful or teasing way
12      benign                  :       kindly; good natured
13      bias                    :       influence in an unfair way
14      bizzare                 :       very strange; unusual; odd
15      bolster                 :       support; strengthen
16      brackish                :       salty
17      breach                  :       break a rule or agreement
18      buoyant                 :       able to float cheerful
19      bypass                  :       to provide an alternative way around
20      cagey                   :       very clever or careful
```

```
DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. Traversal
5. Edit         6. Exit
Enter your choice: 2
Enter the word to delete:
abide
The word "abide" is not found in the Dictionary

DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. Traversal
5. Edit         6. Exit
Enter your choice: 2
Enter the word to delete:
stem
The word "stem" is not found in the Dictionary
```

3)Search Module:

1) Using strcasecmp(), compare the ASCII values of the words,
   i) If it is less than the current node, search in the left subtree, else in the right subtree.
2) If it doesn't match with any if the words, print "Word not found"

```c
void search(char *word)
{
    dict *temp=NULL;
    int flag=0;
    temp=root;
    while(temp)
    {
        if (strcasecmp(word,temp->word)==0)
        {
            printf("Word   : %s\n", word);
            printf("Meaning: %s\n", temp->meaning);
            flag = 1;
            break;
        }
        if(strcasecmp(word,temp->word)<0)
            temp=temp->left;
        else
            temp=temp->right;
        }
        if(flag==0)
            printf("The word \"%s\" is not found in the Dictionary\n",word);
}
```

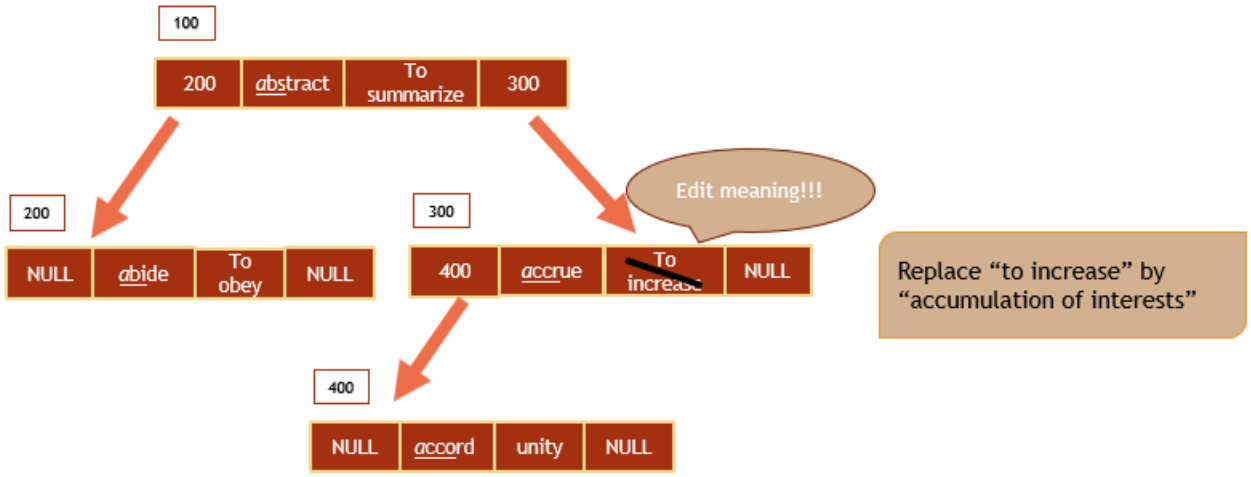
.

```
DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. Traversal
5. Edit         6. Exit
Enter your choice: 3
Enter the search word:
xper
Word   : xper
Meaning: programmer; coder

DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. Traversal
5. Edit         6. Exit
Enter your choice: 3
Enter the search word:
penguin
The word "penguin" is not found in the Dictionary
```

4) Edit Module:

1) Using strcasecmp(), compare the ASCII values, after finding the word,
   i) Change the meaning node.
2) If it doesn't match with any if the words, print "Word not found".




```
void editmeaning(char *word, char *meaning)
{
    dict *temp=NULL;
    int flag=0;
    temp=root;
    while(temp)
    {
        if (strcasecmp(word,temp->word)==0)
        {
            printf("Word    : %s\n", word);
            strcpy(temp->meaning, meaning);
            printf("Meaning: %s\n", temp->meaning);
            printf("You've edited the meaning of the word \"%s\"\n",temp->word);
            flag = 1;
            break;
        }
        if(strcasecmp(word,temp->word)<0)
            temp=temp->left;
        else
            temp=temp->right;
        }
        if(flag==0)
                printf("The word \"%s\" is not found in the Dictionary\n",word);
}
```

Edit module Output:

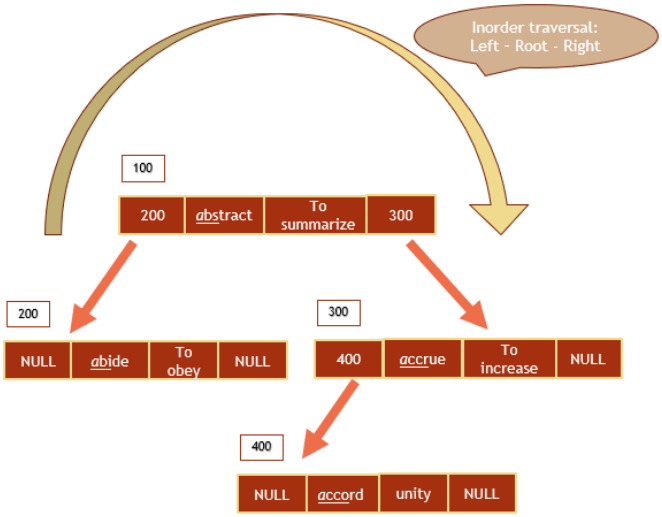
```
DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. View Dictionary
5. Edit meaning 6. Exit
Enter your choice: 5
Enter word and it's new meaning:
accrue
accumulation of intersts
You've edited the meaning of the word "accrue"
The new meaning is
Word    : accrue
Meaning: accumulation of intersts

DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. View Dictionary
5. Edit meaning 6. Exit
Enter your choice: 4
        Word                    :       Meaning
1       abide                   :       to bear patienly
2       abstract                :       to summarize; to epitomize
3       access                  :       gain, obtain
4       accord                  :       to be in agreement, harmony or unity
5       accrue                  :       accumulation of intersts
6       aced                    :       to get the better of
7       adorn                   :       to make sth more beautiful; to decorate
8       affiliate               :       to trace origin to; to accept as a member
9       allay                   :       to calm, alleviate or relieve
10      augment                 :       to grow in size; intensify
```

```
DICTIONARY
1. Insertion    2. Deletion
3. Searching    4. View Dictionary
5. Edit meaning 6. Exit
Enter your choice: 5
Enter word and it's new meaning:
below
towards a lower level
The word "below" is not found in the Dictionary
```

5) View Module:

Using Inorder traversal, traverse the whole tree, which gives us a dictionary. (LEFT → ROOT → RIGHT)




```c
void inorder(dict *myNode)
{
    int i=1;
    dict *temp=myNode;
    void view(dict *temp)
    {
        if (temp)
        {
            view(temp->left);
            printf("%-5d   %-20s      :     %-30s\n",i, temp->word, temp->meaning);i++;
            view(temp->right);
        }
    }
    view(temp);
}
```

```
        Word                    :     Meaning
1       abide                   :     to bear patiently
2       abstract                :     to summarize; to epitomize
3       access                  :     gain, obtain
4       accord                  :     to be in agreement, harmony or unity
5       accrue                  :     to increase; to accumulate
6       aced                    :     to get the better of
7       adorn                   :     to make sth more beautiful; to decorate
8       affiliate               :     to trace origin to; to accept as a member
9       allay                   :     to calm, alleviate or relieve
10      augment                 :     to grow in size; intensify
11      backlog                 :     a build up of work, an accumulation or reserve
12      banter                  :     converse in playful or teasing way
13      benign                  :     kindly; good natured
14      bias                    :     influence in an unfair way
15      bizzare                 :     very strange; unusual; odd
16      bolster                 :     support; strengthen
17      brackish                :     salty
18      breach                  :     break a rule or agreement
19      buoyant                 :     able to float cheerful
20      bypass                  :     to provide an alternative way around
21      cagey                   :     very clever or careful
22      charismatic             :     possessing an extraordinary ability to attract; magnet
23      cheerful                :     joyful, happy
24      chiseled                :     having a distinct and clean outline
25      civil                   :     adequate in courtesy and politeness
26      colossal                :     huge; of enormous size
27      concise                 :     brief and to the point
28      copious                 :     plentiful; abundant; fruitful
29      crucial                 :     extremely important or significant
30      curious                 :     eager to learn or know
31      daffodil                :     a tall yellow flower that grows in spring
32      debtor                  :     the person who owes money
33      decay                   :     the process of being slowly destroyed
34      digit                   :     any of the numbers from 0-9
35      disparage               :     to talk about sb/sth in a critical way
36      dispirited              :     having lost confidence
37      divine                  :     connected with god
38      drab                    :     not interesting or attractive
39      droop                   :     bend or hang downwards
40      dutiful                 :     happy to respect and obey sb
41      earnest                 :     very serious, sincere intent
42      emanate                 :     issue from a source, to come forth
43      epitome                 :     a perfect example
44      equivocal               :     ambiguous, uncertain, doubtful
45      erratic                 :     uneven, irregular
46      erudite                 :     very learned, scholarly
47      eschew                  :     deliberately avoid doing
48      evince                  :     to show or indicate
49      exasperate              :     greatly irritate
50      extrinsic               :     coming from outside
```