ASSIGNMENT NO:3 **DATE: / /2015**

PROGRAM TITLE:Make a Binary Search Tree with the words of a file. Store the frequency count of each of the words in the tree nodes.
. Make the tree case insensitive.
. Convert all letters to lowercase.
. Remove all special characters and use them as word delimiters.

PROGRAM ALGORITHM:

```
create()
{
      convert the file contents into lowercase and remove all the special
characters;
      extract each word from the file and call insert(&root,word);
      print the tree in alphabetical order;
}
insert(root,word)
{
      if(data(root) equal to NULL)
            create new node with word as data(root);
      else
      {
            if(data(root)>word)
                  call  insert(left(root),word);
            else if(data(root)<word)
                  call  insert(right(root),word);
            else
                  increase freq(root) by 1;
      }
}
```

PROGRAM CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct node
{
      char data[100];
      int freq;
      struct node *left;
      struct node *right;
};

void modfile(FILE *);
void insert(struct node **root,char[]);
int display(struct node **root);

int main()
{
    char name[20];
    printf("\n\tEnter the name(with extension) of the file you want to make a
tree of\n\t(NOTE:The file will be modified completedly, so we suggest you keep
a backup)::");
    scanf("%s",name);
    FILE *fp;
    if((fp=fopen(name, "r+")))
    {
        printf("\n\tFile has been opened sucessfully.\n");
        modfile(fp);
        fclose(fp);
    }
    else
    {
        printf("\n\tNo such file exists.Program Terminating.\n");
        return 1;
    }

    //The Tree Part begins
    struct node* root=NULL;
    fp=fopen(name,"r");
    while(fscanf(fp,"%s",name)==1)
    {
           insert(&root,name);
    }
    printf("\n\tThe Tree is::\n\tWORD|FREQUENCY\n\t---------------------");
    display(&root);
    printf("\n");
    fclose(fp);
    return 0;
}

void modfile(FILE *f)
{
    int c;
    char ch;
    ch=getc(f);
    while (ch != EOF)
    {
        c=(int)ch;
        if((c>=65)&&(c<=90))//Changing the uppercase into lowecase
        {
            ch = ch + 32;
            fseek(f, -1L, 1);
            fputc(ch, f);
        }
        else if(((c>=47)&&(c<=57))||((c>=97)&&(c<=122)))//Not harming the
numbers or lowercase
        {}
        else
        {
           ch = 32;
            fseek(f, -1L, 1);
            fputc(ch, f);
        }
        ch=getc(f);
    }
}
void insert(struct node **root,char* item)
```

```
{
    if(*root==NULL)
    {
        struct node* temp=(struct node *)malloc(sizeof(struct node));
        memset(temp->data, '\0', sizeof(temp->data));//this function is used
to initialise all with null character so that the string knows where to end
        strcpy(temp->data,item);
        temp->freq=1;
        temp->left=NULL;
        temp->right=NULL;
        *root=temp;
    }
    else
    {
        int x=strcmp((*root)->data,item);//library function used for
comparing
        if(x>0)
        {
            insert(&(*root)->left,item);
        }
        else if(x<0)
        {
            insert(&(*root)->right,item);
        }
        else
        {
            (*root)->freq=(*root)->freq+1;
        }
    }
}
int display(struct node **root)
{
    if(*root==NULL)
        return 0;
    display(&(*root)->left);
    printf("\n\t%s|%d",(*root)->data,(*root)->freq);
    display(&(*root)->right);
    return 1;
}
```

OUTPUT:

```
    Enter the name(with extension) of the file you want to make a tree of
    (NOTE:The file will be modified completedly, so we suggest you keep a
backup)::abc.txt

    File has been opened sucessfully.

    The Tree is::
    WORD|FREQUENCY
    ---------------------
    11|1
    1920|1
    2012|1
    92|1
    a|1
```

```
after|1
age|1
all|1
american|1
an|2
and|7
as|2
at|1
awards|1
began|1
best|2
born|1
brother|1
california|1
classical|1
collaborating|1
composer|2
culture|1
dance|1
december|1
died|1
director|1
for|2
george|1
glass|1
grammy|1
grew|1
harrison|1
he|1
his|2
in|5
including|1
india|3
indian|3
is|1
known|2
many|1
member|1
music|2
musician|2
musicians|1
notable|1
of|2
on|1
philip|1
popularizing|2
radio|1
ravi|2
s|1
serving|1
shankar|4
sitar|2
states|1
studying|1
success|1
the|3
three|1
```

```
to|1
tour|1
toured|1
troupe|1
united|1
up|1
was|1
western|1
winning|1
with|1
```

DISCUSSION:

1. The original contents of “abc.txt” are

“Ravi Shankar was an Indian musician and composer best known for popularizing the sitar and Indian classical music in Western culture.

Born in India in 1920, Ravi Shankar is an Indian musician and composer best known for his success in popularizing the sitar. Shankar grew up studying music and toured as a member of his brother's dance troupe. After serving as director of All-India Radio, he began to tour India and the United States, winning three Grammy Awards and collaborating with many notable American musicians, including George Harrison and Philip Glass. Shankar died in California on December 11, 2012, at age 92.”

2. The above program modifies the contents of the file to create the tree.
3. The program is a simple application of the B.S.T combined with managing a file.