Source Code

//---------------------------- MAIN.CPP ----------------------------//

```
/*****************************************************************************/
/************************* Name    :-Divesh Uttamchandani  ************************/
/************************* Class   :-XII  A                ************************/
/************************* Session :-2014-2015             ************************/
/*****************************************************************************/
/************************          Computer Science Project       ************************/
/************************        Advance English Dictionary       ************************/
/*****************************************************************************/


/***************************** Give Path of the Files Here ***************************/
#define bgi_loc             "g:/bgi"
#define my_graphics         "g:/project/XII/include/graph.h"
#define fonts_styles        "g:/project/XII/include/styles.h"
#define classes             "g:/project/XII/include/classes.cpp"
#define functions           "g:/project/XII/include/function.h"

#define INDEX               "g:/project/XII/files/m_index.dic"
#define DEFINATION          "g:/project/XII/files/m_def.dic"
#define ABBREVIATIONS       "g:/project/XII/files/abbr.txt"
#define HOWTOUSE            "g:/project/XII/files/use.txt"
/*****************************************************************************/


////////////////////////////////////Variable To Avoid Blinking For The First Time/////////////////////////////////////////
int no_of_times=0;
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


////////////////////////////////////////////////////// Including Header Files ////////////////////////////////////////////
#include <fstream.h>
#include <ctype.h>
#include <conio.h>
#include <graphics.h>
#include <process.h>
#include <dos.h>
#include <string.h>
#include <stdio.h>
#include classes
#include fonts_styles
#include my_graphics
#include functions
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//*********************************Void Main()*********************************//
void main()
{
 intro();
 erase_name();

 do
 {
  int choice=0;
  main_menu(choice);//gets the choice from the user
                    //1)Search A Word
                    //2)Add A Word
                    //3)Abbreviations
                    //4)How To Use
                    //5)Exit
  FULLSCREEN;
  txtbkcolor;
  msgcolor;
  clrscr();

  switch(choice)
  {
   case 1: search();
           break;

   case 2: add();
           break;

   case 3: abbreviations();
           break;

   case 4: how_to_use();
           break;

   case 5: break;
   }

  if(choice==5)
  break;

  FULLSCREEN;    //clear all the screens
  clrscr();

  cout<<"\n\n\n\n\n\n\n\n\n\n\n";
  cout<<"\t\t\t     Press Y to Continue "<<char(174)<<"-";
  cout<<"\n\t\t\t Press Any Other Key To Exit -"<<char(175);
  no_of_times++;
  }while(toupper(getche())=='Y');

end();
}
//*********************************End Of Main*********************************//
```

```
//-------------------------CLASSESS.CPP----------------------------//

#define WORDMAX       80
#define DEFMAX        6000


////////////////////////.///////////Declaration and Definition of class index_word ////////////////////////////////////
class index_word
{
  char       w[WORDMAX];
  char linked_w[WORDMAX];
  unsigned int wordno;                                          //wno of the linked word

public:

  index_word()
  {
     strcpy(w,"NULL");
     strcpy(linked_w,"NULL");
     wordno=0;
  }

  ~index_word()
  {
     ;
  }
  void add(char w1[], char linked_w1[],unsigned int code)
  {
  strcpy(w,w1);
  strcpy(linked_w,linked_w1);
  wordno=code;
  }

  void display()
  {
  cout<<"\n\nWORD\t"<<w;
  cout<<"\nLinked word\t"<<linked_w;
  cout<<"\nCode\t"<<wordno;
  }

  char *getword()
  {
  return w;
  }

  unsigned int retno()
  {
  return wordno;
  }
};
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//////////////////////////////////////Declaration and Definition of class definition//////////////////////////////////////
class defination
{
  unsigned int wordno;
  char        w[WORDMAX];
  char         d[DEFMAX];
  public:
  defination()
  {
     strcpy(w,"NULL");
     strcpy(d,"NULL");
     wordno=0;
  }

  ~defination()
  {
     ;
  }

  void add(unsigned int wno,char *w1,char *d1)
  {
  strcpy(w,w1);
  strcpy(d,d1);
  wordno=wno;
  }

 void display()
  {
  int i=0;
  cout<<"\n";//<<"WORD\t";
  puts(w);
  cout<<"\n";//Defination\n";
  while(d[i]!='\0')
  {
    if(i%45==0 && i!=0)
    {
    if(d[i-1]!=' ' && d[i]!=' ')
    cout<<'-';
    cout<<"\n";
    }

    if(i%(45*15)==0 && i!=0)
    {
       cout<<"\nPress Any Key To Scroll";
       getch();
       clrscr();
    }
    cout<<d[i];
    i++;
  }
  }
};
```

```
//------------------------------STYLES.H------------------------------//

#define INPUTHEADER             window(1,22,48,22)
#define INPUTWINDOW             window(1,23,48,25)

#define WORDLISTWINDOW          window(50,2,80,21)
#define WORDHEADER              window(50,1,80,1)

#define OUTPUTHEADER            window(1,1,48,1)
#define OUTPUTWINDOW            window(1,2,48,21)

#define USERHEADER              window(50,22,80,22)
#define USERNOTE                window(50,23,80,25)

#define FULLSCREEN              window(1,1,80,25);

#define PARTITION               window(49,1,49,25);

//fonts and colors

/****************************** Background Color ******************************/
#define bkcolor                 setbkcolor(15)
/******************************************************************************/


/************************ colors and fonts for main menu ************************/
#define erasercolor             setcolor(15)    //same as bkcolor

#define topiccolor              setcolor(4)
#define topicstyle              settextstyle(6,0,6)

#define namecolor_intro         setcolor(2)
#define namecolor_end           setcolor(3)
#define namestyle               settextstyle(6,0,4)
#define th_style                settextstyle(6,0,1)

#define mainmenucolor           setcolor(7)
#define mainmenustyle           settextstyle(6,0,4)
#define menumsgcolor            setcolor(1)
#define menumsgstyle            settextstyle(6,0,3)

#define optionscolor            setcolor(13)
#define optionsstyle            settextstyle(6,0,3)

#define choiceboxcolor          setcolor(4)

#define abtstyle_topic          settextstyle(6,0,6)
#define abtcolor_topic          setcolor(4)
#define abtstyle_body           settextstyle(6,0,4)
#define abtcolor_body           setcolor(3)
#define abtstyle_msg            settextstyle(6,0,2)
#define abtcolor_msg            setcolor(4)
```

```
#define thankscolor                 setcolor(5)
#define thanksstyle                 settextstyle(6,0,6)
/*************************************************************************/

/*************************Textbackground etc. for main *********************/
#define txtbkcolor                  textbackground(2)
#define msgcolor                    textcolor(1)
#define iocolor                     textcolor(4)
#define errcolor                    textcolor(4)
/*************************************************************************/
```

Please Turn Over.....

```
//---------------------------- GRAPH.H ----------------------------//
//Defining Co-ordinates Of mid points
#define xmax getmaxx()
#define ymax getmaxy()
#define midx xmax/2
#define midy ymax/2

/************************** Declaration of Functions ****************************/
//////////////////////////////////////////////////funtion to initialize graphics/////////////////////////////////////////////////////////
void graphics_initialize();
/////////////////////////////////////////////Functions For Introduction Animation/////////////////////////////////////////////
void intro();
   void erase_name();
/////////////////////////////////// Functions For Printing Menu And Getting Choice ///////////////////////////////////
void main_menu(int &choice);
   void print_menu();
   int get_choice();
/////////////////////////////////////////// Function For End Thanks Animation ////////////////////////////////////////////
void end();
/////////////////////////////////// Function Which Helps In Key Controlled Delay ///////////////////////////////////
void mydelay(int time, int &flag);
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/*********************************************************************************************/

/**************************** Defination of Functions ****************************/
void graphics_initialize()
{
  /* request auto detection */
  int gdriver = DETECT, gmode, errorcode;

  /* initialize graphics and local variables */
  initgraph(&gdriver, &gmode, bgi_loc);

  /* read result of initialization */
  errorcode = graphresult();
  /* an error occurred */
  if (errorcode != grOk)
  {
    cout<<"Graphics error: %s\n"<<grapherrormsg(errorcode);
    cout<<"Press any key to halt:";
    getch();
    exit(1);
  }

  bkcolor;
}

//the function prints topic and my name.
//it uses kbhit for pausing and skipping the animation
//if user presses any key (exept enter) it gets into the statement if(kbhit)
//and than getch(). to mimic pause play
//it breaks out of the animation loop if user presses enter
```

```
void intro()
{
  graphics_initialize();
  int x = xmax;
  int y = ymax;
  int i,j;

  for(i=x, j=0; i>40; i--,j++)
  {
   delay(20);

   erasercolor;
   bar(x,y,0,0);

   topicstyle;
   topiccolor;
   outtextxy(i,10,"Advance English Dictionary");

   if(j<=200)
    {
     namestyle;
     namecolor_intro;
     outtextxy(j,350,"Made By :-Divesh Uttamchandani");
     outtextxy(j,400,"Class     :-XII -A");

     th_style;
     outtextxy(j+179,397,"th");
    }

   if(j>=200)
   {
   y=80;
   }


  if(kbhit()!=0)
   {
        if(getch()==13)
                break;
        else
                getch();
   }
}
  cleardevice();
  topicstyle;
  topiccolor;
  outtextxy(40,10,"Advance English Dictionary");

  namestyle;
  namecolor_intro;
  outtextxy(200,350,"Made By :-Divesh Uttamchandani");
  outtextxy(200,400,"Class     :-XII -A");
```

```
   th_style;
   outtextxy(200+179,397,"th");
   delay(500);
}

//it erases my name from the screen
//skips animation if any key is pressed using my delay
void erase_name()
{
   //graphics initialized in intro

   int flag=0;                                                                    //for mydelay
   mydelay(500,flag);

   erasercolor;
   for(int i=195;i<=xmax;i++)
   {
   bar(0,300,i,ymax);
   mydelay(5,flag);
   }
   cleardevice();
  // closegraph();                                                          //transfered to main menu
}

//it prints the main menu. and gets choice from the user using
//functions print_menu() and get_choice()
//NOTE:-graphics are initialized in this function only if the user is
//coming back to this menu 2nd and later times
//else the graphics are brought forward from intro.
//(As i Don't Want The Screen To Blink due to closegraph)
//for the above purpose it uses global variable no_of_times
void main_menu(int &choice)
{
   if(no_of_times>=1)                    //As i Don't Want The Screen To Blink due to
   graphics_initialize();                //closegraph


   topicstyle;
   topiccolor;
   outtextxy(40,10,"Advance English Dictionary");

   print_menu();

   choice=get_choice();

   cleardevice();
   closegraph();
}
```

```
//the function prints the menu
void print_menu()
{
int flag=0;                                                                    //for mydelay

mydelay(500,flag);

mainmenustyle;
mainmenucolor;
outtextxy(250,120,"Main Menu");

mydelay(700,flag);

optionsstyle;
optionscolor;

outtextxy(230,180,"1) Search A Word");

mydelay(700,flag);
outtextxy(230,220,"2) Add A Word");

mydelay(700,flag);
outtextxy(230,260,"3) Abbreviations");

mydelay(700,flag);
outtextxy(230,300,"4) How To Use");

mydelay(700,flag);
outtextxy(230,340,"5) Exit");


menumsgcolor;
menumsgstyle;
delay(20);
outtextxy(140,410,"Use Arrow Keys To Move The Selection");

delay(20);
outtextxy(220,440,"Press \'Enter\' To Enter");
}


//function helps in getting user's choice using arrow keys
//it draws and changes the selection box(rectangle) at required position.
//it returns choice whose value is from 1 to 5
//1) Search A Word
//2) Add A Word
//3) Abbreviations
//4) How to Use
//5) Exit
int get_choice()
{
//grapics inherited from main_menu()
```

```
delay(100);
int choice=1;
int y1=180;
int y2=217;
int ch=0;

do
{
choiceboxcolor;
rectangle(200,y1,440,y2);                                                    //print the box

ch=getch();

switch(ch)
        {
        case 72:    //down arrow key
           erasercolor;
                rectangle(200,y1,440,y2);                                    //erase the previous rectangle
                y1-=40;
                y2-=40;
                choice--;
                break;

        case 80:   //up arrow key
           erasercolor;
                rectangle(200,y1,440,y2);                                    //erase the previous rectangle
                y1+=40;
                y2+=40;
                choice++;
                break;
        }
 if(choice<1)                                                                //if less than search a word
 {
 y1+=40,y2+=40;                                  //since y1 & y2 also get decrement in switch by 40
 choice++;
 }

 if(choice>5)                                                                //if greater than exit
 {
 y1-=40,y2-=40;                                   //since y1 & y2 also get increment in switch by 40
 choice--;
 }
}while(ch!=13);

return choice;
}

//thankyou note similar to intro
void end()
{
  graphics_initialize();
```

```
  int x = xmax;
  int y = ymax;

  int i,j;

  for(i=x, j=0; i>210; i--,j++)
  {
   delay(20);

   erasercolor;
   bar(x,y,0,0);

   thanksstyle;
   thankscolor;
   outtextxy(i,100,"Thank You");

   if(j<=100)
    {
     namestyle;
     namecolor_end;
     outtextxy(j,350,"Made By :-Divesh Uttamchandani");
     outtextxy(j,400,"Class     :-XII -A");

     th_style;
     outtextxy(j+179,397,"th");
    }

   if(j>=100)
   {
   y=250;
   }

   if(kbhit()!=0)
   {
         if(getch()==13)
               break;
         else
               getch();
   }
}
  cleardevice();
  thanksstyle;
  thankscolor;
  outtextxy(210,100,"Thank You");

  namestyle;
  namecolor_end;
  outtextxy(100,350,"Made By :-Divesh Uttamchandani");
  outtextxy(100,400,"Class     :-XII -A");

  th_style;
  outtextxy(100+179,397,"th");
```

```
  /* clean up */
  for(i=0;i<3000;i++)
  {
  if( !(kbhit()) )
        delay(1);
  else
        break;
  }

  cleardevice();
  closegraph();
  exit(0);
}

//mydelay function holds responsible for key based delay
//it only delays if parameter flag is 0 i.e no key is pressed untill now
//if key is pressed in b/w it changes the value of flag to 1 (flag is called
//by ref. here)and breaks of the loop
//next time when it is invoked as it changed the value of flag to 1 it doesn't
//enters the loop untill flag given is again turned to 0
void mydelay(int time,int &flag)
{
if(flag==0)
   {
        for(int i=0;i<=time;i++)
        {
           if(kbhit()!=0)
           {
                getch();
                flag=1;
                break;
           }
           else
                delay(1);
        }
   }
}
/*************************************************************************/
```

```
//---------------------------- FUNCTION.H ----------------------------//

#define ESC 27          //define escape key


/*************************** Declaration of Functions ******************************/
int my_strcmpi(char *userword,char *indexword);
void search();                              //gets word from user and serches for it and calls disp def
unsigned int search(char *&A);          //search funtion overloaded this searches for A and returns its-
                                                                                    //-wordno
void disp_def(unsigned int wordno);                                         //displays defination

void add();                                                                 //add a word to the dictionary

void abbreviations();                                                       //display list of abbreviations

void how_to_use();                                                      //displays how to use the dictionary
/***********************************************************************************/


/***************************  Definition of Functions  *****************************/
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//function that matches userword and indexword
int my_strcmpi(char *userword,char *indexword)
{
 /////////correct A/////////
 if(indexword[0]=='-'||indexword[0]=='\'')
 indexword++;
 if(indexword[0]=='-'||indexword[0]=='\'')
 indexword++;
 ////////////////////////////
 int s=strlen(userword);
 return strncmpi(userword,indexword,s);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//function to display defination
void disp_def(unsigned int wordno)
{
 OUTPUTWINDOW;
 clrscr();
 long long unsigned int t;
 defination d;

 ifstream fin(DEFINATION,ios::binary);

 if(!fin)
        cout<<"error";
 else
 {
  fin.seekg(0);
```

```
 t=(wordno-1)*(long long unsigned)(sizeof(d));
 fin.seekg(t,ios::beg);
 fin.read((char*)&d,sizeof(d));
 d.display();
 }
 getch();
 fin.close();
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//function to search a word
void search()
{
 char *A,*B;
 unsigned int wlist[20];
 int i=1,n,temp;
 clrscr();

 PARTITION;
 clrscr();
 for(temp=1;temp<=25;temp++)
 {
  putch('|');
 }
 //Input Word From User
 INPUTHEADER;
 puts("------------------Input Window------------------");

 OUTPUTHEADER;
 clrscr();
 puts("------------------Defination------------------");

 //Search Results
 WORDHEADER;
 puts("---------Search Results--------");

 USERHEADER;
 puts("----------Note To User----------");

 USERNOTE;
 clrscr();
 puts("Press Any Key To Scroll");
 putch(13);
 puts("Press (Enter) to select list");

 INPUTWINDOW;
 textattr(1+128);
 txtbkcolor;
 clrscr();
 puts("Enter Word to be Searched for:");
 gets(A);
```

```
msgcolor;
clrscr();
puts("Enter Word to be Searched for:");
putch(13);
puts(A);

ifstream fin(INDEX,ios::binary);

if(!fin)
cout<<"error";

else
{
fin.seekg(0);
index_word I;
WORDLISTWINDOW;
 clrscr();

 puts("    ");
 putch(13);
 cout<<"Searching...";

 while(fin.read((char*)&I,sizeof(I)))
  {
  B=I.getword();

  if(my_strcmpi(A,B)==0)
    {
        //20 words at a time
        if(i==1)
        clrscr();
          if(i>20)
          {
          char ch;
          ch=getch();

           if(ch==13)
            break;
           else
           {
           clrscr();
           i=1;
           }
    }

  wlist[i-1]=I.retno();
                                          //Truncate long words
        if(strlen(B)>23)
        {
        B[20]=B[21]=B[22]='.';
        B[23]='\0';
        }
```

```
        printf(" %d)\t",i);
        puts(B);
        if(i<20)
        putch(13);
          i++;
   }
  }
if(i>1)
{
 USERNOTE;
 clrscr();

 INPUTWINDOW;
 do
 {
        textattr(1+128);
        txtbkcolor;
        clrscr();
        puts("Enter Choice(Word No.)");
        putch(13);
        cin>>n;

        msgcolor;
        clrscr();
        puts("Enter Choice(Word No.)");
        putch(13);
        printf("%d",n);

        if(!(n>=1 && n<=i-1))
        {
        textattr(1+128);
        txtbkcolor;
        clrscr();
        cout<<"Invalid! Enter Again!";
        getch();
        }
 }while(!(n>=1 && n<=i-1));
fin.close();

OUTPUTWINDOW;
disp_def(wlist[n-1]);
}
else
 {
 WORDLISTWINDOW;
 clrscr();
 cout<<"No Match Found";
 getch();
 }
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//search function overloaded returns the word code of  A
unsigned int search(char *&A)
{
 char *B;
 unsigned int wlist[20];
 char *lwlist[20];
 int i=1,n,temp;

 clrscr();

PARTITION;
 clrscr();
 for(temp=1;temp<=25;temp++)
 {
  putch('|');
 }
 //Input Word From User
 INPUTHEADER;
 puts("------------------Input Window------------------");

 OUTPUTHEADER;
 clrscr();
 puts("------------------Defination------------------");

 //Search Results
 WORDHEADER;
 puts("---------Search Results--------");

 USERHEADER;
 puts("----------Note To User----------");

 USERNOTE;
 clrscr();
 puts("Press Any Key To Scroll");
 putch(13);
 puts("Press (Enter) to select list");

 INPUTWINDOW;
 puts("Enter Word to be Searched for:");
 putch(13);
 puts(A);

 ifstream fin(INDEX,ios::binary);

 if(!fin)
 cout<<"error";
```

```
else
{
fin.seekg(0);
index_word I;
WORDLISTWINDOW;
 clrscr();

 puts("    ");
 putch(13);
 cout<<"Searching...";

 while(fin.read((char*)&I,sizeof(I)))
  {
  B=I.getword();

  if(my_strcmpi(A,B)==0)
    {
        //20 words at a time
        if(i==1)
        clrscr();
          if(i>20)
          {
          char ch;
          ch=getch();

           if(ch==13)
            break;
           else
           {
           clrscr();
           i=1;
           }
    }

  wlist[i-1]=I.retno();
  strcpy(lwlist[i-1],I.getword());

        //Truncate long words
        if(strlen(B)>23)
        {
        B[20]=B[21]=B[22]='.';
        B[23]='\0';
        }

        printf(" %d)\t",i);
        puts(B);
        if(i<20)
        putch(13);

          i++;
    }
  }
```

```
if(i>1)
{
 USERNOTE;
 clrscr();

 INPUTWINDOW;
 do
 {
        textattr(1+128);
        txtbkcolor;
        clrscr();
        puts("Enter Choice(Word No.)");
        putch(13);
        cin>>n;

        msgcolor;
        clrscr();
        puts("Enter Choice(Word No.)");
        putch(13);
        printf("%d",n);

        if(!(n>=1 && n<=i-1))
        {
        textattr(1+128);
        txtbkcolor;
        clrscr();
        cout<<"Invalid! Press Any Key To Enter Again!";
        cout<<"\nPress Esc to Exit";
        char ch=getch();
        if(ch==ESC)
         {
         txtbkcolor;
         msgcolor;
         clrscr();
         return 0;
         }
        }
 }while(!(n>=1 && n<=i-1));
fin.close();

OUTPUTWINDOW;

disp_def(wlist[n-1]);
strcpy(A,lwlist[n-1]);
return wlist[n-1];
}
```

```
else
   {
   WORDLISTWINDOW;
   clrscr();
   cout<<"No Match Found";
   getch();
   return 0;
   }
  }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//function to add a word
void add()
{
 char *w,*linked_w,*def;
 unsigned int code=0;
 int flag=0;

 w=new char[WORDMAX];
 linked_w=new char[WORDMAX];
 def=new char[DEFMAX];

 index_word I;
 defination D;

 cout<<"Add A Word To The Dictionary\n";
 cout<<"\nEnter Word\t";       gets(w);

 cout<<"\nDo You Want To Link It To An Already Existing Word?\t(Y/N)\t";

 if(toupper(getche())=='Y')
 {
 cout<<"\n\nEnter Linked Word\t";
 gets(linked_w);
 cout<<"\nNow You Are Being Redirected To The Search Window...";
 cout<<"\nPress Any Key";
 getch();
 code=search(linked_w);

 FULLSCREEN;
 clrscr();
 }

 if(code==0)
 {
 clrscr();
 cout<<"\nYou didn't Link It To A Word";
 cout<<"\n\nEnter Defination Of The Word\n";
 cout<<"\nPress Any Key To Continue\nPress Esc To Exit\t";
 char ch=getch();
```

```
if(ch==ESC)
{
clrscr();
cout<<"Word Not Added!!!!!!!\nPress Any Key\t";
getch();
return;
}

else
{
cout<<"\nEnter Defination\n";
gets(def);
flag=1;

ifstream fin(INDEX,ios::binary);
if(!fin)
cout<<"Error";
else
 {
 fin.seekg(-(int)(sizeof(I)),ios::end);
 fin.read((char*)&I,sizeof(I));
 code=I.retno();
 code++;
 strcpy(linked_w,w);
 fin.close();
 }
}
}

FULLSCREEN;
clrscr();

//copy the word to index
I.add(w,linked_w,code);

ofstream ind(INDEX,ios::binary|ios::app);
if(ind)
 {
  ind.seekp(ios::end);
 ind.write((char*)&I,sizeof(I));
 ind.close();
 }
else
 cout<<"Error";

if(flag==1)    //then add to defination also
{
D.add(code,w,def);
ofstream def(DEFINATION,ios::binary|ios::app);
def.seekp(ios::end);
```

```
 if(def)
   {
   def.write((char*)&D,sizeof(D));
   def.close();
   }
  else
  cout<<"Error";
  }

  cout<<"\nWord Added";

  I.display();
  getch();

  delete w;
  delete linked_w;
  delete def;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Function to print abbreviations
void abbreviations()
{
ifstream fin(ABBREVIATIONS);
if(!fin)
cout<<"Error";

else
 {
 cout<<"Abbreviations Used In Dictionary\n\n";
 fin.seekg(0);
 char ch;
  while(fin.get(ch))
   {
   cout<<ch;
    if(wherey()==24)
     {
     cout<<"\nPress Any Key To Scroll";
     getch();
     clrscr();
     cout<<"Abbreviations Used In Dictionary\n\n";
     }
   }
 cout<<"\n\nPress Any Key";
 getch();
 fin.close();
 }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//print how to use the dictionary
void how_to_use()
{
ifstream fin(HOWTOUSE);

if(!fin)
cout<<"Error";

else
 {
 cout<<"Organization of The Dictionary\n\n";
 fin.seekg(0);
 char ch;
  while(fin.get(ch))
   {
   cout<<ch;
    if(ch=='\n')
     {
     fin.get(ch);
     if(ch=='\n')
      {
      cout<<"\n\nPress Any Key To Scroll";
      getch();
      clrscr();
      cout<<"Organization of The Dictionary\n\n";
      }
     else
     cout<<ch;
     }
   }
 cout<<"\n\nPress Any Key";
 getch();
 fin.close();
 }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/*************************************************************************************/
```