Dictionary with Binary Search

GROUP MEMBER NAMES AND SIDs		
SR#	***NAMES***	***SIDs***
1	***Muhammad Kashif***	***11762***
2		
3		

Description

English Dictionary: contains more than 1000 words and Meaning, User Search a Meaning by entering a word, the Program Uses Binary Search to fine the word

Other Feature: Program can remove the entered word using clear function and also close the program using close function

Program (Project) is made in Python Language. Uses Libraries Numpy as Backend and Tkinter as frontend

Algorithm

1. Start
2. Array = {Contain more than 1000 words}
3. arrayMeaning = {Contain more than 1000 Meaning of words}
4. Word = search input
5. Left = 0
6. Right = length of array – 1
7. While loop till left is equal to right
    a. Mid = (left – right)/2
    b. If array[mid] == word
        i. Print(Meaning [mid])
    c. Else if array[mid] > word
        i. Left = mid
    d. Else if array[mid] < word
        i. Right = mid

Code

```
from tkinter import *
import json
from numpy import array
from difflib import get_close_matches
from tkinter import messagebox
#get_close_matches(word,possibilities,n,cuttoff)
#close_match =
get_close_matches('appel',['ape','apple','peach','puppy'],n=3,cuttoff=0.6)

#Back End

def search():
    data=json.load(open('data.json'))
    word=enterwordEntry.get()
    word = word.lower()
    k = open("keys.txt","r")
    key = array([0 for i in range(2610)], dtype = 'object')
    for i in range(len(key)):
        key[i] = k.readline()
    value = array([0 for i in range(2610)], dtype = 'object')
    v = open("values.txt","r")
    for i in range(len(value)):
        value[i] = v.readline()
    """
    for i in data.keys():
        key[count] = i
        count+=1
    count = 0
    for i in data.values():
        value[count] = i
        count+=1"""
```

```
    if word in data:
        """
        meaning=data[word]
        textarea.delete(1.0,END)
        for item in meaning:
            addtextarea.insert(END,u'\u2822'+item+'\n\n')
            textarea.insert(END,u'\u2822'+item+'\n\n')"""
        right = len(key)
        midpoint = len(key)//2
        left = 0
        while(True):
            if(key[midpoint] == word):
                meaning=value[midpoint]
                textarea.delete(1.0,END)
                for item in meaning:
                    addtextarea.insert(END,u'\u2822'+item+'\n\n')
                    textarea.insert(END,u'\u2822'+item+'\n\n')
                break
            elif(word > key[midpoint]):
                left = midpoint
                midpoint = (right - left)//2
            elif(word < key[midpoint]):
                right = midpoint
                midpoint = (right - left)//2
    elif len(get_close_matches(word,data.keys()))>0:
        cloased_match = get_close_matches(word,data.keys())[0]
        res = messagebox.askyesno('Confirm','Did you mean '+cloased_match+' instead?')
        if res == True:
            enterwordEntry.delete(0,END)
            enterwordEntry.insert(END,cloased_match)
            meaning = data[cloased_match]
            textarea.delete(1.0, END)
            for item in meaning:
                addtextarea.insert(END,u'\u2822'+item+'\n\n')
```

```
                textarea.insert(END,u'\u2822'+item+'\n\n')


        else:
            textarea.delete(1.0, END)
            messagebox.showinfo('Information', 'Please type a correct word')
            enterwordEntry.delete(0, END)
    else:
        messagebox.showerror('Error', 'The word doesnt exist.Please double check it.')
        enterwordEntry.delete(0,END)
        textarea.delete(1.0,END)
"""
def binary_search():
    data=json.load(open('data.json'))
    word=enterwordEntry.get()
    word = word.lower()
    #data.sort()
    midpoint = len(data)//2
    while(True):
        if(data[midpoint] == word):
            meaning=data[midpoint]
            textarea.delete(1.0,END)
            for item in meaning:
                addtextarea.insert(END,u'\u2822'+item+'\n\n')
                textarea.insert(END,u'\u2822'+item+'\n\n')
            break
        if(word > data[midpoint]):
            data = data[midpoint]
        if(word < data[midpoint]):
            data = data[0: midpoint]
        midpoint = (len(data)//2)
"""

def clear():
    enterwordEntry.delete(0,END)
```

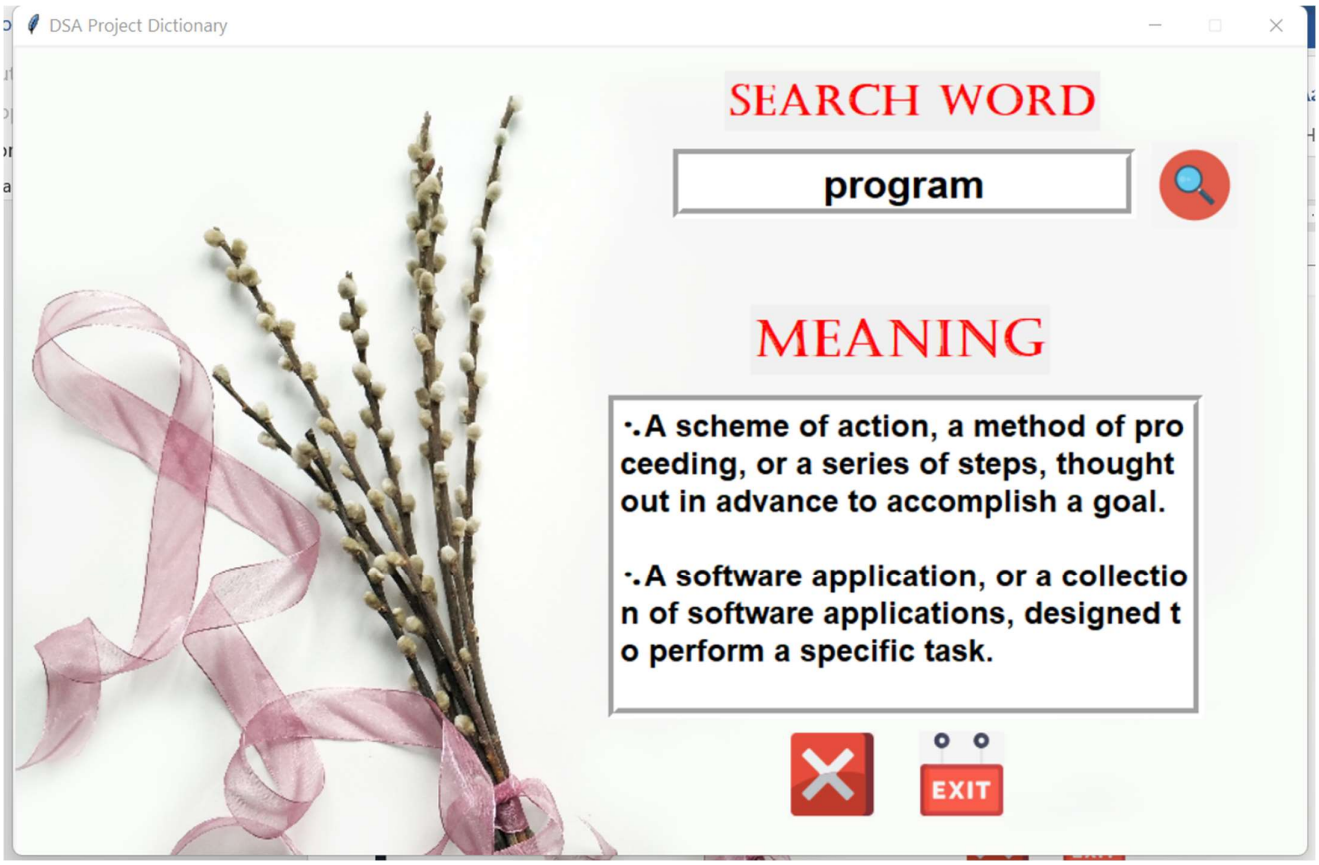
```
    textarea.delete(1.0,END)

def addclear():
    addenterwordEntry.delete(0,END)
    addtextarea.delete(1.0,END)

def iexit():
    res = messagebox.askyesno('Confirm','Do you want to exit? ')
    if res == True:
        root.destroy()
    else:
        pass




#Front End
root = Tk()
root.geometry('1000x626+100+50')
root.title('DSA Project Dictionary')
root.resizable(False,False)
bgimage = PhotoImage(file='bg.png')
bgLabel = Label(root,image=bgimage)
bgLabel.place(x=0,y=0)

enterwordlabel = Label(root,text='Search
Word',font=('castellar',25,'bold',),foreground='red')
enterwordlabel.place(x=550,y=20)

enterwordEntry =
Entry(root,font=('arial',23,'bold'),justify=CENTER,bd=8,relief=GROOVE)
enterwordEntry.place(x=510,y=80)
```

```
addenterwordlabel = Label(root,text='Add
Word',font=('castellar',25,'bold',),foreground='red')
addenterwordlabel.place(x=80,y=20)

addenterwordEntry =
Entry(root,font=('arial',23,'bold'),justify=CENTER,bd=8,relief=GROOVE)
addenterwordEntry.place(x=30,y=80)

searchimage = PhotoImage(file='search.png')
searchButton = Button(root,image = searchimage, bd =
0,bg='whitesmoke',cursor='hand2',activebackground='whitesmoke',command=search)
searchButton.place(x=880,y=75)

Meaninglabel =
Label(root,text='Meaning',font=('castellar',30,'bold',),foreground='red')
Meaninglabel.place(x=570,y=200)

textarea = Text(root,width=34,height=8,font=('arial',18,'bold'),bd=8,relief=GROOVE)
textarea.place(x=460,y=270)

addMeaninglabel = Label(root,text='Add
Meaning',font=('castellar',30,'bold',),foreground='red')
addMeaninglabel.place(x=60,y=200)

addtextarea = Text(root,width=32,height=8,font=('arial',18,'bold'),bd=8,relief=GROOVE)
addtextarea.place(x=10,y=270)

clearimage = PhotoImage(file='clear.png')
clearButton = Button(root,image = clearimage, bd =
0,bg='whitesmoke',cursor='hand2',activebackground='whitesmoke', command=clear)
clearButton.place(x=600,y=530)

addclearimage = PhotoImage(file='clear.png')
```

```
addclearButton = Button(root,image = clearimage, bd =
0,bg='whitesmoke',cursor='hand2',activebackground='whitesmoke', command=addclear)
addclearButton.place(x=100,y=530)

addButtonimage = PhotoImage(file='addition.png')
addButton = Button(root,image = addButtonimage, bd =
0,cursor='hand2',activebackground='whitesmoke', command=addclear)
addButton.place(x=200,y=530)

exitimage = PhotoImage(file='exit.png')
exitButton = Button(root,image = exitimage, bd =
0,bg='whitesmoke',cursor='hand2',activebackground='whitesmoke', command=iexit)
exitButton.place(x=700,y=530)

def enter_function(event):
    searchButton.invoke()

root.bind('<Return>',enter_function)

root.mainloop()
```

Output


DSA Project Dictionary
SEARCH WORD
MEANING
EXIT


DSA Project Dictionary
SEARCH WORD
program
MEANING
·.A scheme of action, a method of pro ceeding, or a series of steps, thought out in advance to accomplish a goal.
·.A software application, or a collectio n of software applications, designed t o perform a specific task.
EXIT