DOCUMENTATION OF FUNCTIONS USED IN THE PROGRAM

FUNCTION NAME	DESCRIPTION	RETURN VALUE
Find(links, url)	To find the link in the list of links.	False or True
spider_links(url , links)	Find all the links on the url page and add it to links.	links
number_links(url, links)	Find number of links on the url that exist in the list links	number of such links
printMatrix(matrix)	Prints the pagerank matrix	None
Swap(matrix, i, j)	Swaps ith and jth value of matrix	None
transpose(matrix)	Finds the transpose of the matrix	Transpose of matrix
MultiplyByScalar(matrix, scalar)	Multiplies matrix by scalar	Final Matrix
magnitude(matrix)	Finds magnitude of column vector 'matrix'	Final matrix
MultiplyByMatrix(matrix, rank)	Multiply 'matrix' (square matrix) with 'rank' (column matrix)	Final Matrix
addMatrix(matrix_1, matrix_2)	Adds two square matrices	Final Matrix
ConvergeCheck(matrix_1, matrix_2)	Check if matrix_1 has converged to matrix_2	True or False
transMatrix(file_name)	Creates transition matrix with the data stored in file 'file_name'	Transition matrix
markovMat(file_name)	Creates markov matrix from transition matrix	Markov matrix
googleMat(p, file_name)	Creates google matrix from markov matrix with p as the damping factor	Google matrix
page_ Rank(file_ name)	Calculates page ranks using data from filre 'file_name'	Ranks
helpinsert(string tmp, int index, node** hash_ t)	Breaks up the string to be inserted into words and then sends it to the insert	void
insert(string tmp, int index, node** hash_ t)	Inserts the string along with its corresponding website index into the hash table	void
finding(string search, node** hash _t)	Performs a typical hash search	Node*

	on the string passed int the passed hash table	
ret_ascii(string tmp)	Computes the sum % 100 of the ASCII values of the characters of the string passed to it	int
openpage(string website ,bool offload)	Pushes website onto stack only if not present and at the end sorts them calling QuickSort	void
openwebpage(int a)	Opens the webpage corresponding to the clicked button	void
helpfind(string search, bool offload1)	Performs preliminary checks on a string before it is searched for	bool
quicksort(vector <int> &v1, int left, int right)	To sort large number of page ranks	void
CreateWindowEx(DWORD dwExStyle, “STATIC”, LPCTSTR lpWindowName, DWORD dwStyle, int x , int y , int nWidth , int nHeight, HWND hWndParent, HMENU hMenu, HINSTANCE hInstance, LPVOID lpParam);	It creates a static text which displays the string given by the programmer in place of LPCTSTR lpWindowName. X , y are position of left corner of the text display.	Handle to a window HWND
CreateWindowEx(DWORD dwExStyle, “BUTTON”, LPCTSTR lpWindowName, DWORD dwStyle, int x , int y , int nWidth , int nHeight, HWND hWndParent, HMENU hMenu, HINSTANCE hInstance, LPVOID lpParam);	It creates a button displaying the text in the lpWindowName string. When the button is clicked send a message with wPARAM as hMenu .	Handle to window HWND
WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)	This is the main method of graphics WIN32 GUI similar to int main().	Int
WNDCLASSEX wc	Creates a window class when given sufficient parameters.	
CreateWindowEx(DWORD dwExStyle, LPCTSTR lpWindowClass, LPCTSTR lpWindowName,	lpWindowClass is replaced by the name of the class created by user.	Handle to window HWND

DWORD dwStyle, int x , int y , int nWidth , int nHeight, HWND hWndParent, HMENU hMenu, HINSTANCE hInstance, LPVOID lpParam);		
LRESULT CALLBACK WndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)	It is a windows procedure function which processes all the messages .	Int
LoadMyImage(void)	Loads the background image	Void
SendMessage(hsti, STM_SETIMAGE,(WPARAM) IMAGE_BITMAP, (LPARAM) hBitmap);	Displays the background image	Message
cleaning(hwnd,non_ex)	Destroys all the useless buttons texts and cleans the stacked variables when non_ex is zero.	Void
GetWindowText(TextInput,&Searchinput[0],90) ;	Gets the text in the text box of which the handle is given in place of TextInput into an array of characters SearchInput.	Int(!=0 if successful)
ShowWindow(hwnd, nCmdShow);	Displays the main window	bool
UpdateWindow(hwnd)	Updates the window	bool
TranslateMessage(&Msg);	translates virtual-key messages into character messages.	bool
DispatchMessage(&Msg)	It is typically used to dispatch a message retrieved by the Getmessage function.	bool
check(const char* word)	Returns true if word is in dictionary else false.	bool
load(const char* dictionary)	Loads dictionary into memory. Returns true if successful else false.	bool
size(void)	Returns number of words in dictionary if loaded else 0 if not yet loaded.	int
unload(void)	Unloads dictionary from memory. Returns true if successful else false.	bool