Name: Mean Diamand
Neptun Code: KSG25Z

Birthday and Anniversary

Documentation

My current program is operating on the window command prompt. When the users run and compile the program, the command prompt will pop up on the screen showing the introduction of the program which will display the picture, the title of the program, and the main menu. Initially, after the program display all these three things on the window command prompt, it will ask the users to input an integer parameter for their choices whether they wanted to start the databases or choosing other options in the main menu as can be shown below.

My Birthday and Anniversary Databases is consisting of:

- 3 Structures:

Name: Mean Diamand
Neptun Code: KSG25Z

1. **name** used for defining a data type for storing first name and surname of the users by declaring character string data type structure elements called f_name(for the first name) and s_name(for the surname).
2. **date** used for defining new data types for storing the year, month, day of the date by declaring integer data type structure elements called year, month, day.
3. **list** used for defining a list of the name of the users or the husband and wife if they are couple, category, type, and date of the users by declaring a structure element called 'n' and 'w'(for the first name and surname) as the name data type from the name structure, character string data type structure elements called category(for the category of the users. Ex: Friend, Relative, Family, Colleague…etc.), type (for the type of day. Ex: Birthday, Nameday, Anniversary, Marriage…etc.), and a structure element called 'd' (for the year, month, and day) as the date data type from the date structure. This structure also stores the address of the list as well.

- 16 Functions:
  1. **read_the_file()** is a void function that used for reading the text file that wanted to be display by using file handling method for reading the text file.
  2. **introduction()** is a int function that return the input choice of the users when they wanted to select the option in the menu and it also used for reading a text file that contains the introduction of my program as shown below. In this function, I implemented a file handling method for reading a text file.

Name: Mean Diamand
Neptun Code: KSG25Z

3. **databases()** is a void function that used to choose the options which are displayed on the databases menu that will be shown when the users start the databases. In this function, I implemented the same switch case in the conditional statement method just like in the options function except that I added decoration for the databases menu by printing the title and the options for this menu.

4. **data_entry()** is a void function that receives one address type of list, I used for entering the input information from the users into the database by using the variables which are declared by the list which is also connected to the date and name structures. In this function, I declare a list type pointer called p and allocate the memory space of it. Then, let the users input all the information into the node.

5. **read_database()** is a function that receives one address type of list and returns the address type of list, I used for reading the data that were already inside the text file into the list. For example, if we use a text file that already contained all the information, but that information wasn't contained in the list then this function will read all that information into the list.

6. **save_database()** is a void function that receives two address types of list, I used it for saving the data that the user input to the node to the text file by using the file handling method which is writing into the file.

7. **insert_data()** is a function receives two address type of list then return an address of the list, which I used for inserting the node into the list. First, I declare a list pointer type called u. Then, I give a condition if the head is NULL which means if the list is empty then the program will call the push_front() function to add the first node into this empty list. After that, I create a loop to move u to the last node of the list. Then, I allocate the memory space for the node after the last node. After that, I use the strcpy() function to copy the string from a node that we need to copy to the new node in the list, and same for the year, month, and day, we initialize the new node to be equal to the node

that we need to copy. Finally, I let the last node point to NULL since it doesn't have any node to point to.

8. **push_front()** is a function that receives two addresses of the list and return an address of the list which I used for adding the first node into an empty list. First, I declare a list type pointer called p and allocate the memory space of it. After that, I use the strcpy() function to copy the string from a node that we need to copy to the new node in the list, and same for the year, month, and day, we initialize the new node to be equal to the node that we need to copy. Then, I let the pointer of the last node point to NULL since it doesn't have any node to point to, and the head will be equal to p. For example, when we start to input the first information into the database which is empty without any information inside.

9. **printPerson()** is a void function that receives one address type of list, I used for printing all the information of the node.

10. **data_search()** is a void function that receives one address type of list, I used for the users to choose the options which are displayed on the search menu that will be shown when the users choose the searching for the data in the databases option in the databases menu. In this function, I implemented the same switch case in the conditional statement method just like the other options functions.

11. **search_category()** is a void function that receives one address type of list I used for the users to choose the category of data that they want to search for in the database. In this function, I implement the switch case to let the user choose which category that the user wants to display, and I declare a list pointer type variable called p, initialize it equal to head which is the first node. Then, I make a loop and give a condition to compare the string str with the category inside the p and if it is the same then I used the printPerson()

function to print the node. After that, I moved to the next node, and this keeps on through every node until it goes through all the nodes.

12. **search_day()** is a void function that receives one address type of list I used for the users to choose the type of day of the data that they want to search for in the database. In this function, I did the same way as the search_category() function but I changed the options to choose the type of day instead.

13. **last_next()** is a void function that receives one address type of list I used for displaying the day that occurred last month and the day that will occur next month. In this function, I check the month by using the time.h library to use the time_h data type, time() function, struct tm structure, and *localtime() function for extracting the real month from the computer and use it for checking to select the day that occurred last month and next month. Since the months in a year has 12, for last month, I decided to create an if-else statement to check if the month is January then the last month should be December and besides that just decrement the month by 1 then the program will display the day that occurred last month and for next month, I decided to create another if-else statement to check if the month is December then the next month should be January and besides that just increment the month by 1 then the program will display the day that will occur next month. After that, I declare a list pointer type called p and let it point to the address of the head which is the first node that stores the information of the first person in the database. Then, I make a loop to and give the if conditional statement if the month data inside that node is equal to the month1 which is the last month then print that node using the printPerson(p) function with the address of p that I already declared, then p will go to the next node and check the month of the next node and print it again if the month is last month. This loop will keep on until it goes through all the nodes. The same goes for the next month, I also used the same method for checking and printing that.

Name: Mean Diamand
Neptun Code: KSG25Z

14. **print_databases()** is a void function that receives a one address type of list which I used for printing out the data containing inside the link list which is the structure how the databases.

15. For the two functions that responsible for deleting data, which is data_delete(), the function that I'm planning to use for receiving the input name from the users and pass it to delete_person() to delete the person are still in working process.

In the main function, I have the option switch case which will take the return value from the introduction() function and jump into the case according to that value. I also called the read_the_file() function to use file handling method on the help, credit and history text file.