Programming Project

Names:

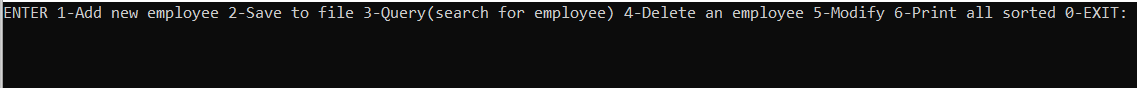
1. Yasser Mohamed Khairy Salama ID: 7598
2. Mohamed Ahmed Mohamed Mohamed Morsy ID: 7514
3. Asem Mohamed Saad Aboumadawi ID: 7755

1.Introduction:

This program will create and maintain a company management system to keep track of Employees inside the company.

2.User Manual:

In this program we have a lot of options and functions to do, after this window appear.



1. In order to add new employee: the user should enter 1, then he has to input the data of the new employee (ID, Last Name, First Name, Salary, Date(dd/mm/yyyy), Address, Phone Number and Email), after he finishes, he will be able to add new one or do other functions. (Return to the first sentence).
2. In order to save any change to the file: the user will have to enter 2.
3. In order to search for an employee: the user will have to enter 3, then he should input the last name of employee then all employee with this last name will appear in the screen with details.
4. In order to delete an employee: the user will have to enter 4, then he should input the last name and first name, anyone has these two names together will be deleted from the file.
5. In order to modify the information of an employee: the user will have to enter 5, then he should input the ID of this employee then the user can modify any of these information (Last Name, First Name, Salary, Date(dd/mm/yyyy), Address, Phone Number and Email) then he must input 0 after finishing modification.
6. In order to print all employees sorted: the user will have to enter 6. Then he should input what kind of sort he wants (According to Last Name, Salary or Date of Birth).
7. In order to exit: the user will have to enter 0, then the program will notify the user that he is exiting without saving his changes if he had made any.

**3.Code description:**

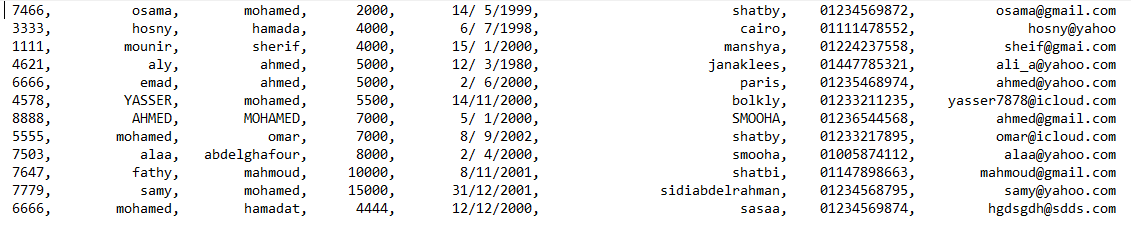
We declared an array of employees (of kind struct) that contain all the employees read by load function which is called by default.

We used a while loop to keep asking the user for the function he want to do until he finishes and exit.

Create file function is called by default that takes the file name and check if the file exists or not and create one if not exist.

We declared two stucts one for the employees and one for the date of birth, a struct of struct was used while declaring date of birth in the employee struct.

**1-LOAD function:** This function takes the file name that is declared in the main, the mode and the array of employees, this function is used to take data from the file and put it in the array of employees, in this function the file is opened with the mode “r”. A while loop is used until the end of file and we took every line to a string then used **strtok** function to divide the line into (ID, Last Name, First Name, Salary, Date(dd/mm/yyyy), Address, Phone Number and Email), then another while loop to assign each part from the string to an employee by using switch, **atoi** function is used to assign any integer (ID, salary, Date) and **strcpy** function to assign any string (Last Name, First Name, Address, Phone Number and Email) and a counter increase for each line is returned which means (number of employees = counter-1).



**2-ADD function:** In this function a do while loop is used for each input to keep asking for this input until entering a valid one by making the while condition to be the return of the **is\_valid** functions which we will discuss later. This function return employee of type struct.

Text

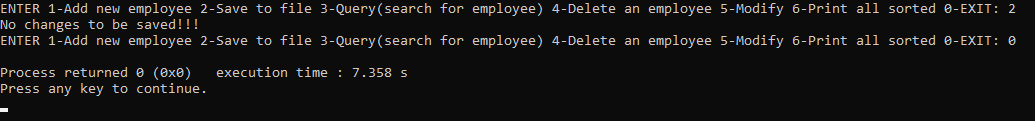
Description automatically generated

A picture containing graphical user interface

Description automatically generated

The employee is added successfully.

**3-SAVE function:** This function takes the file name, the mode “w”, the array of employees and their number. we used a for loop with index starting from 0 to less than the number of the employees in which we used **sprintf** function(which stores output on char buffer instead of printing on console) to give it the format we want to save with then **fprintf** function to print the stored string in the buffer to the file. But if we called the save function without making any changes it will print “no data to be saved”. When this function is called an variable called j in the main is assigned to zero which indicates that no more data to be saved.



**4-QUERY function:** This function takes the array of employees, last name that we search for entered by user and number of employees. We used **linear search** to find the last name we are searching for by making a for loop with index starting from 0 to less than the number of the employees to compare the last name entered by user with each employee’s last name ,a variable found is initialized by zero and increase by 1 when we find any employee with this last name and this one is printed with details by call the print function we made and if no one matches with this last name we are searching for the found will stay zero it will say that there is no employee with this last name.

**Linear search algorism:**

arr[100],x

for i=0 to n // stop if i=n

if arr[i]==x

return i

return -1

**Text

Description automatically generated**

The code printed all the employees that have the same entered last name by the user

**5-DELETE function:** This function takes the array of employees, last name and first name of employee the user want to delete ,takes a pointer to the number of employees and a pointer to the variable that indicates that there was some changes happened needed to be saved we start with a for loop with index starting from 0 to less than the number of the employees to compare and search for the employee that has these last and first names and if found the variable found which is initialized by 0 increases by 1 and we enter another for loop from the index of the employee we want to delete to (less than the number of the employees-1) and shifts every employee after that one to the one before him.

Text

Description automatically generated

Text

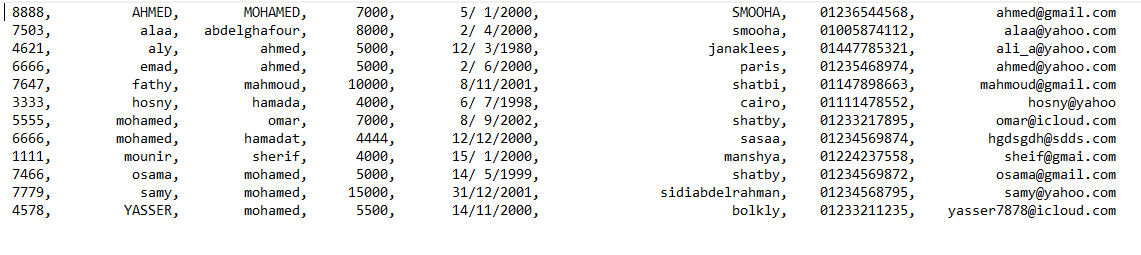
Description automatically generated with medium confidence

All employees with last name Mohamed and first name Ahmed are all deleted successfully.

**6-MODIFY function:** This function takes array of employees, id of employee the user search for, number of employees and a pointer to the variable that indicates that there were some changes happened needed to be saved (p), if the ID that we enter matches one of the IDs of the employees in the file then the program will enter the while loop to modify certain details of the employee according to the user choice until the user choose to stop modifying. If the employee modified any data, the content of pointer (p) increases by one.

Text

Description automatically generated



The salary of employee whose ID is 7466 is successfully modified from 2000 to 5000.

**7-PRINT function:** This function is used to print all the employees by calling it in a for loop from 0 to a smaller number of employees, it takes an employee from the array of employees and print him with details, but all of this happen after the user choose to print them sorted according to what (last name, salary, date of birth). They are sorted using bubble sort.

**Bubble Sort algorism:**

i,j

arr[100],temp

for i=0 to size //stop if i=size

for j=0 to n-size-1 //stop if j=size-i-1

if (arr[j]>arr[j+1])

temp=arr[j]

arr[j]=arr[j+1]

arr[j+1]=temp

**note:** the last three lines in this algorithm are for swapping,

1. **Sorted with last name:**

This function takes the array of employees and their number. Then the bubble sort algorithm is used but in the if condition **strcasecmp** function is used to compare the element comes first with the one after it so if it returns positive number the two elements get swapped with each other.

Text

Description automatically generated with medium confidence

Table

Description automatically generated with medium confidence

1. **Sorted with salary:**

This function takes the array of employees and their number. Then the bubble sort algorithm is used to sort them ascendingly according to their salary value.

Graphical user interface

Description automatically generated with low confidence

Table

Description automatically generated with low confidence

1. **Sorted with date of birth:**

This function takes the array of employees and their number. Then the bubble sort algorithm is used to sort them ascendingly according to their date of birth from the youngest employee by knowing that with if conditions that compare the year at first then the year with month then year, month, and day.

**Text

Description automatically generated with low confidence**

**Text

Description automatically generated with medium confidence**

**8-EXIT:** When the user enters 0 to exit there is a variable j that was initialized by 0 and increases by 1 when making any modification and is assigned again to zero after calling save to file function so if j not equal to zero the program warn the user by a message and a peep sound that he is exiting without saving some data and ask him to continue or exit without saving.

Text

Description automatically generated

**9-IS\_VALID functions:** In these functions the program checks if the user has entered valid input or not these functions return 1 if valid and 0 if not valid.

So, that’s why a do while loops were used in the add and modify functions to let the user input data and it will keep asking for this data until the user enter valid option.

1. **Is\_valid\_int:**

This one was used to check that the input are integers, so it was saved at first as a string and sent to this function and check if all the string characters are between ‘0’ and ‘9’. This function was used in the input of ID and phone number but in phone number we added one more condition that its length must be 11 by using **strlen** function.

ID is then changed to integers by **atoi** function to assign it to the struct of the employee.

1. **Is\_valid\_name:**

This function was used for last name and first name to check that they are only letters, so the function takes and use a loop was used until we get null character of this string and check if all its characters are between ‘a’ and ‘z’ or ‘A’ and ‘Z’ and if not, it returns 0.

1. **Is\_valid\_date:**

This function takes day, month and year entered by user and check that the day is between 1 and 31, month between 1 and 12 and if the year is less than 2018 if not return 0.

1. **Is\_valid\_email:**

This function takes email entered by user and check if it contains “@” and “.” by declaring 2 pointers, the first one takes the return of the **strstr** function that search for “@” in the string and if exists it returns a pointer to its place so if the first pointer is not equal to null character ‘\0’ so the “@” was found, so after that the second pointer takes the return of the **strstr** function that search for “.” If it was found too(second pointer not equal to ‘\0’),it returns 1, But if any pointer of these 2 was equal to null character the function return 0.

1. **Is\_valid\_float:**

This one was used to check that the input is a float number, so it was saved at first as a string and sent to this function and check if all the string characters are between ‘0’ and ‘9’ or ‘.’. Then if the input is valid then it is converted into float using **atof** function to assign it to the struct of the employee.

Text

Description automatically generated

That’s how the program worked when the user entered some invalid data.