 INTERNSHIP

PROJECT-1

EMPLOYEE MANAGEMENT SYSTEM

**Contents :**

1.Introduction

2.Description

3.Problem Statement

4.Module Description

5.Source Code

6.Outputs

7.Conclusion

**Developed by: K.V.J.RAMAN**

**GUIDE: Mr. BHARATH**

**Mr. PRITHVI**

**Introduction**

 Hence after the completion of the project we got familiar with the C programming and its features. A complete and useful library management can only be developed with lot of intensive effort and time. Due to lack of time and we are beginners in programming program that we expected can’t be developed by us. Our library management may not be must useful for library in our college but it will be the most useful for study and programming practice using C. As a whole, the project has been a good learning experience for us. We have gained knowledge about the various aspects of C programming. At the same time, we have developed a deep understanding about the file handling in C.

**Description**

**Employee management system is a project that helps us to store the employee information.Once the user login in this application so that we can add,.save,browse the data.This project was implemented in C programming.**

**High level requirements**

**- It describes the set of capabilities in which a project must achieve a set of capabilities and the expected outcomes and this has to be delivered by the project.**

**- First of all we should have a clear idea about the project and a particular attention to the capabilities and conditions.**

**The main constraints for the high level requirements are:**

**- Windows(Operating System).**

**- C Language.**

**- Design.**

**- Implementation.**

**Low level Requirements**

**- Processing.**

**- Technical details.**

**- Functionalities.**

**- Calculations.**

**- Performance.**

**# 4W's & 1H**

**Who**

**- Employee Management System**

**What**

**- Employee management system helps us to store and to know about the data given by the employee.**

**When**

**- It helps us when the employee join in company or in any organization.**

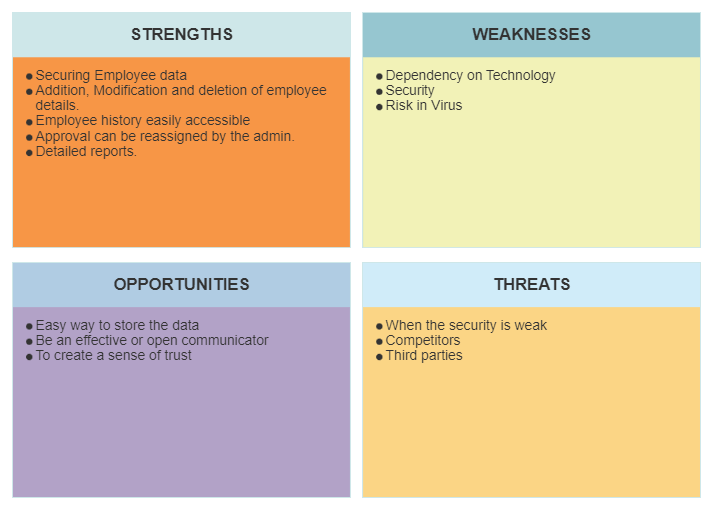
**Where**

**- This is used in any company or organization.**

**How**

**- It is implemented in c language and by creating a system which will provide required details of an employee in a system of the company or the organization.**

**SWOT ANALYSIS**

****

**Modules Description**

Analysis: This system, in order to facilitate customer input and make the program clear, so I use the module method,  
Modularize each function to make the calls between functions in the program clearer.  
**Function module description**  
Describe the functions of each module.  
**Entry module**：Enter the information of the existing employees of the factory and save it in the file to facilitate the sorting and update  
New, statistics, printing and other operations.  
**Output module**: Output all employee information or required employee information on the screen for printing or checking  
Ask for employee information.  
  
**Add module**: Add the new employee information.  
**Sorting module**: This program only sorts the salary ,name of employees.  
**Query module**: This module is divided into query by name, salary, and id.

**Complete Source Code**

**#include <stdio.h>**

**#include <string.h>**

**struct employee{**

**int id;**

**float salary;**

**char name[20];**

**}emp[2];**

**void DisplayStruct(int index){**

**for (int i=0; i<=(index-1); i++){**

**printf("\nEmployee id is %d\n",emp[i].id);**

**printf("Employee salary is %f\n",emp[i].salary);**

**printf("Employee name is %s\n",emp[i].name);**

**}**

**}**

**void EnterEmp(int index){**

**for (int i=0; i<=(index-1); i++){**

**printf("\n---Enter Details of %d Employee---\n",i);**

**printf("\nEnter Employee id : ");**

**scanf("%d",&emp[i].id);**

**printf("\nEnter Employee salary : ");**

**scanf("%f",&emp[i].salary);**

**printf("\nEnter Employee name : ");**

**scanf("%s",emp[i].name);**

**}**

**}**

**//This will returns new index**

**//that is new total no. of employees**

**int Enter\_New\_Employees(int index, int extra){**

**int end;**

**end = index + extra;**

**for (int i=index; i<=(end-1); i++){**

**printf("\n---Enter Details of %d Employee---\n",i);**

**printf("\nEnter Employee id : ");**

**scanf("%d",&emp[i].id);**

**printf("\nEnter Employee salary : ");**

**scanf("%f",&emp[i].salary);**

**printf("\nEnter Employee name : ");**

**scanf("%s",emp[i].name);**

**}**

**return end;**

**}**

**void Search\_emp\_by\_id(int x, int index){**

**int k = 0;**

**for (int i=0; i<=(index-1); i++){**

**if (x == emp[i].id){**

**printf("Employee by id %d is %s\n",x,emp[i].name);**

**}**

**k++;**

**}**

**if (k == (index-1))**

**printf("\nEmployee Not Found !!\n");**

**}**

**int main(){**

**int index = 0;**

**int y;**

**int extra;**

**printf("\n Select any one from the following \n");**

**printf("1. Enter Employees\n");**

**printf("2. Display Employees\n");**

**printf("3. No. of Employees\n");**

**printf("4. Search any Employee by id\n");**

**printf("5. Enter New Employees Data\n");**

**printf("6. Quit \n");**

**jump :**

**printf("\nEnter Choice : ");**

**scanf("%d",&y);**

**switch (y) {**

**case 1 :**

**printf("Enter Number of Employees : ");**

**scanf("%d",&index);**

**printf("\nEnter Employees Data : \n");**

**EnterEmp(index);**

**goto jump;**

**case 2 :**

**printf("\n----Employees Data is as follows---- \n");**

**DisplayStruct(index);**

**goto jump;**

**case 3 :**

**printf("\n------COUNTING--------\n");**

**printf("\nNumber of Employees are %d\n",index);**

**goto jump;**

**case 4 :**

**printf("\n---Search Employee by ID-----\n");**

**int x;**

**printf("Enter ID : ");**

**scanf("%d",&x);**

**Search\_emp\_by\_id(x,index);**

**goto jump;**

**case 5 :**

**printf("\n---New Employee Entry---\n");**

**printf("\nEnter Number of New Employees : ");**

**scanf("%d",&extra);**

**index = Enter\_New\_Employees(index,extra);**

**goto jump;**

**case 6 :**

**break;**

**default :**

**printf("\n Enter Valid Choice\n");**

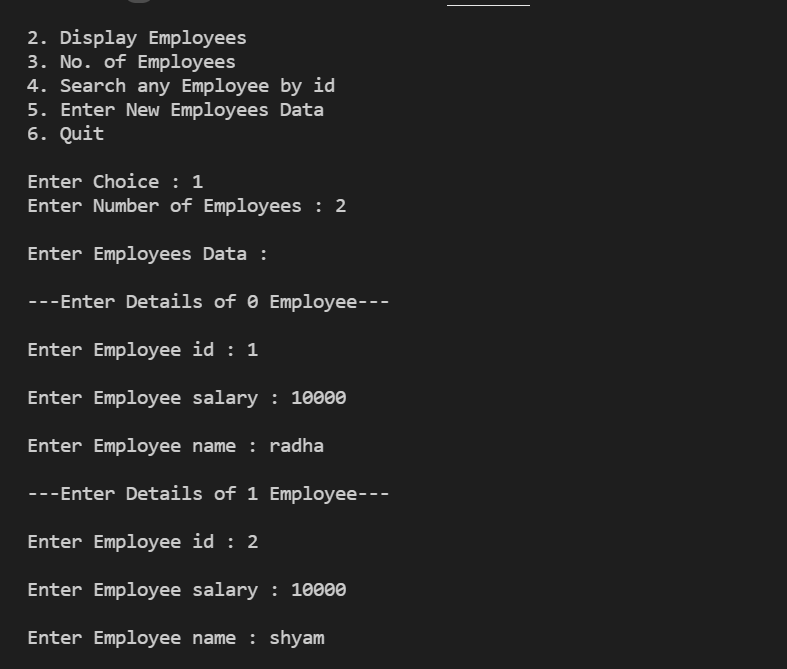
**goto jump;**

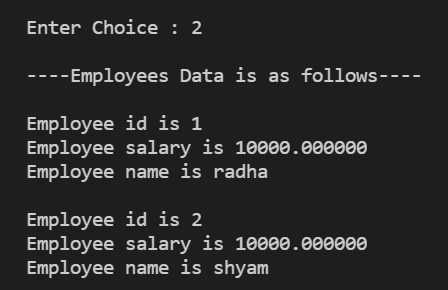
**}**

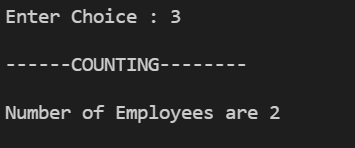
**return 0;**

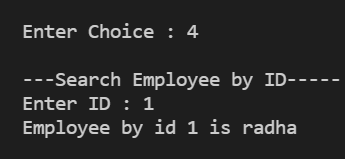
**}**

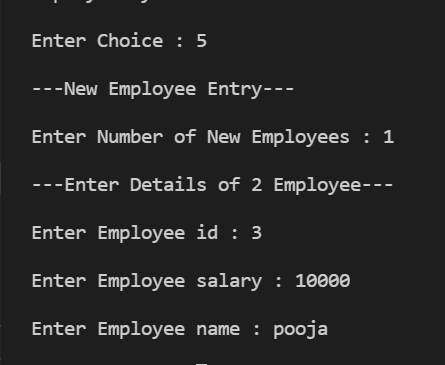
**OUTPUT:**

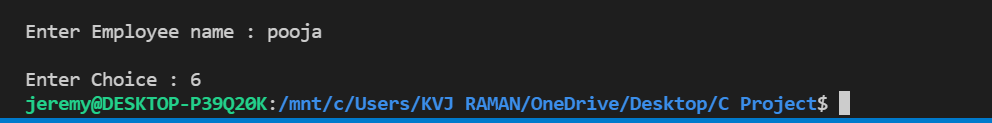
****

****

****

****

****

****

**Conclusion**

On-line algorithms were presented for complex num-

ber arithmetic using a redundant complex number system

(RCNS) to represent complex numbers as a single number.

Implementations were compared to a network of real on-li ne

adders, to demonstrate a signiﬁcant reduction in cost and in

delay.

Such algorithms can be further developed for other op-

erations such as complex square-root, complex square, and

composite algorithms can be designed for actual applica-

tions involving complex numbers, such as the Fast Fourier

Transform (FFT), recursive digit ﬁlters, and various other

applicatio

On-line algorithms were presented for complex num-

ber arithmetic using a redundant complex number system

(RCNS) to represent complex numbers as a single number.

Implementations were compared to a network of real on-li ne

adders, to demonstrate a signiﬁcant reduction in cost and in

delay.

Such algorithms can be further developed for other op-

erations such as complex square-root, complex square, and

composite algorithms can be designed for actual applica-

tions involving complex numbers, such as the Fast Fourier

Transform (FFT), recursive digit ﬁlters, and various other

applications

 Hence after the completion of the project we got familiar with the C programming and its features. A complete and useful library management can only be developed with lot of intensive effort and time. Due to lack of time and we are beginners in programming program that we expected can’t be developed by us. Our library management may not be must useful for library in our college but it will be the most useful for study and programming practice using C. As a whole, the project has been a good learning experience for us. We have gained knowledge about the various aspects of C programming. At the same time, we have developed a deep understanding about the file handling in C. On-line algorithms were presented for complex num-

ber arithmetic using a redundant complex number system

(RCNS) to represent complex numbers as a single number.

Implementations were compared to a network of real on-li ne

adders, to demonstrate a signiﬁcant reduction in cost and in

delay.

Such algorithms can be further developed for other op-

erations such as complex square-root, complex square, and

composite algorithms can be designed for actual applica-

tions involving complex numbers, such as the Fast Fourier

Transform (FFT), recursive digit ﬁlters, and various other

applicatio

On-line algorithms were presented for complex num-

ber arithmetic using a redundant complex number system

(RCNS) to represent complex numbers as a single number.

Implementations were compared to a network of real on-li ne

adders, to demonstrate a signiﬁcant reduction in cost and in

delay.

Such algorithms can be further developed for other op-

erations such as complex square-root, complex square, and

composite algorithms can be designed for actual applica-

tions involving complex numbers, such as the Fast Fourier

Transform (FFT), recursive digit ﬁlters, and various other

applicati

 This system adopts C language traditional data structure that can be expanded by itself:**Linked list implementation. Compared to using an array for storage, it is scalable!** The system realizes the**Add, check, print out, write to disk, read disk information and other basic functions.**  
The system has good scalability and can extend other functions. For example: console encryption, combined query, etc.·Multiple functions·!