ALGORITHM

int main(void)-

1. START

2. Create File Pointer FILE *data

3. Call function- head_disp()

4. OPEN File employeeRecord.txt in Read/Edit mode(r+).

5. IF file Open Operation is Unsuccessful THEN

6. PRINT “Employee Database Not Found. Press N key to create new database. To EXIT press any other key.”

7. READ create

8. IF create = ‘n’ THEN create employeeRecord.txt

9. ELSE Exit the Application

10. WHILE(1)

11. OPEN head_disp() function

12. PRINT All Choices Related to Employee management for user to choose.

13. READ choice

14. SWITCH(choice) and go to respective functions as per user’s choice

15.END

void emp_append()-

1. OPEN File employeeRecord.txt in Append mode

2.OPEN head_disp() function to display the title of software

3.READ all Employee Details present in **personal** structure

4. APPEND All Employee Record data entered in the file, using **fwrite** function.

5.IF Successful, Display Appropriate Messages.

6.CLOSE File employeeRecord.txt

7.PRINT Press 1 to Continue and 0 to Exit

8. READ exit_status

9. IF exit_status is not equal to 1 THEN

10. EXIT

void emp_modify( )-

1. OPEN head_disp() function

2. OPEN File employeeRecord.txt in r+ mode

3. PRINT “Enter Employee ID to edit Record”

4. READ id

5. WHILE End of File is not Reached

6. IF personal.emp_id equals id THEN

7. PRINT Existing Record Details

8. READ New Employee Information and Write it to the file employeeRecord.txt

9. IF Record Not Found THEN

10. PRINT Record Not Found

11. CLOSE File employeeRecord.txt

12. Ask User to Continue or not and perform necessary actions

13. END

void emp_search( )-

1. OPEN head_disp()

2. OPEN File employeeRecord.txt in r mode

3.PRINT User to enter either Employee ID or Employee name to search

4. READ search_choice

5. READ Employee Search ID or Name as per previous choice using SWITCH construct

6. WHILE End of File is not reached

7. IF Employee Id or Name matches one in file THEN

8. THEN PRINT All Employee Record

9. IF found=0 i.e. Employee Not Found THEN

10. PRINT “No Such Employee Record Found”

11. Ask user to Continue, and take appropriate actions as required.

12. END

void emp_delete( )-

1. OPEN head_disp()

2. Open File employeeRecord.txt, and also create new file temp_data.txt

3. PRINT “Enter Employee ID to delete it’s Record”

4. READ del_emp_id

5. WHILE till End of File has been reached

6. IF del_emp_id is NOT equal to personal.emp_id THEN

7. Write all records to temp_data.txt file

8. CLOSE employeeRecord.txt and temp_data.txt

9. REMOVE employeeRecord.txt from the current directory

10. RENAME temp_data.txt to employeeRecord.txt

11. PRINT “Employee Successfully Deleted”

12. Ask User to Continue or not and take appropriate actions required.

13. END

void emp_displayAll( )-

1. OPEN head_disp() function

2. OPEN File employeeRecord.txt in Read (r) mode.

3. WHILE End of File is Not Reached

4. PRINT All Employees Individual Information

5. PRINT Total Employees in File

6. Ask User to Continue or not, and take appropriate actions as required.

7. END

void emp_display( )-

1. OPEN function head_disp()

2. OPEN File employeeRecord.txt

3. PRINT “Enter Employee ID to display it’s Record”

4. READ search_emp_id

5. WHILE End of File has not been Reached

6. IF personal.emp_id equals search_emp_id

7. found =1

8. PRINT All Details of that Employee

9. IF found equals 0 THEN

10. PRINT “No Such Employee Record Found”

11. Ask User to Continue or not, and take appropriate actions required.

12. END