EMPLOYEE MANAGEMENT SYSTEM

USED LIBRARY FILES

- #include<stdio.h> // standard input output
- #include<string.h> // strings
- #include<stdlib.h> // standard library

LIST OF FUNCTIONS USED IN CODE

- Int empty()
- Void insert()
- Int checkid()
- Int validate()
- int age()
- Int checkexp()
- Int contact()
- Void update()
- Void disp()
- Void searchid()
- Int avlid()
- Void searchdept()
- Int avldept()
- Void sort()
- Int sortid()
- Void deletedata()

DESCRIPTION OF UDF(S)

- **Int empty()**
- Step 1: open file in reading mode
- Step 2: if there is record then return c=1
- Step 3: else return c=0
- Step 4: close the file

DESCRIPTION OF UDF(S)

- **Void insert()**
- Step 1: open file in append mode
- Step 2: take input of employee details
- Step 3: write the record into file
- Step 4 : close the file

DESCRIPTION OF UDF(S)

- **Int checkid()**
- Step 1: open file in reading mode.
- Step 2: compare inserted id with exist id .
- Step 3: if it is valid then return 0 otherwise 1.
- Step 4: close the file.

DESCRIPTION OF UDF(S)

- **Int validate()**
- Step 1: take input of date
- Step 2: check whether the date is valid or not
- Step 3: if date is not valid then ask again
- Step 4: when the date is valid return it

DESCRIPTION OF UDF(S)

- **Int age()**
- Step1: take age of employee
- Step 2: check if it is between 21-65 then it is valid
- Step 3: else again ask the age

DESCRIPTION OF UDF(S)

- **Int checkexp()**
- Step 1: open file in reading mode.
- Step 2: compare inserted experience with age.
- Step 3: if it is valid then return experience otherwise ask again to insert valid experience.
- Step 4: close the file.

DESCRIPTION OF UDF(S)

- **Int contact()**
- Step 1: take input of contact number
- Step 2: check if it 10 of digits or not
- Step 3: if it is not valid ask again for valid contact number

DESCRIPTION OF UDF(S)

- **Void disp()**
- Step 1: open file in reading mode
- Step 2: print the record of file one by one by using loop
- Step 3: close the file

DESCRIPTION OF UDF(S)

- **Void searchid()**
- Step 1: take input of id
- Step 2: check if the id is available or not (by calling function “avlid()”)
- Step 3: if the id is available then “display” the details
- Step 4 : if id is not available ask user if he/she want to search another id.

DESCRIPTION OF UDF(S)

- **Int avlid()**
- Step 1: open file in reading mode
- Step 2: compare input id and check in looping whether id is available or not
- Step 3: close the file

DESCRIPTION OF UDF(S)

- **Void searchdept()**
- Step 1: take input of department name to search
- Step 2: check if the dept.name is available or not (by calling function “avldept()”)
- Step 3: if the dept. name is available then “display” the details
- Step 4 : if dept. name is not available then ask user if he/she want to search another dept. name.

DESCRIPTION OF UDF(S)

- **int avldept()**
- Step 1: open file in reading mode
- Step 2: compare input dept and check in looping whether id is available or not
- Step 3: close the file

DESCRIPTION OF UDF(S)

- **Void sort()**
- Step 1: open file in reading mode
- Step 2: copy one-one salary into array
- Step 3: compare value of array to salary and print the details in sort by salary
- Step 4: close the file

DESCRIPTION OF UDF(S)

- **Void sortid()**
- Step 1: open file in reading mode
- Step 2: copy one-one id into array
- Step 3: compare value of array to id and print the details in sort by id
- Step 4: close the file

DESCRIPTION OF UDF(S)

- **Void deletedata()**
- Step 1: take input of id which you want to delete and check it is available or not
- Step 2: if it is available then copy all the record into tamparary file excepting input id's record .
- Step 3: copy temporary file into Record file
- Step 4: close all files

DESCRIPTION OF UDF(S)

- **Void update()**
- Step 1: take id from user
- Step 2: give choice to the user where he/she want to update
- Step 3: according to choice update record
- Step 4: ask user if he/she want to continue updating