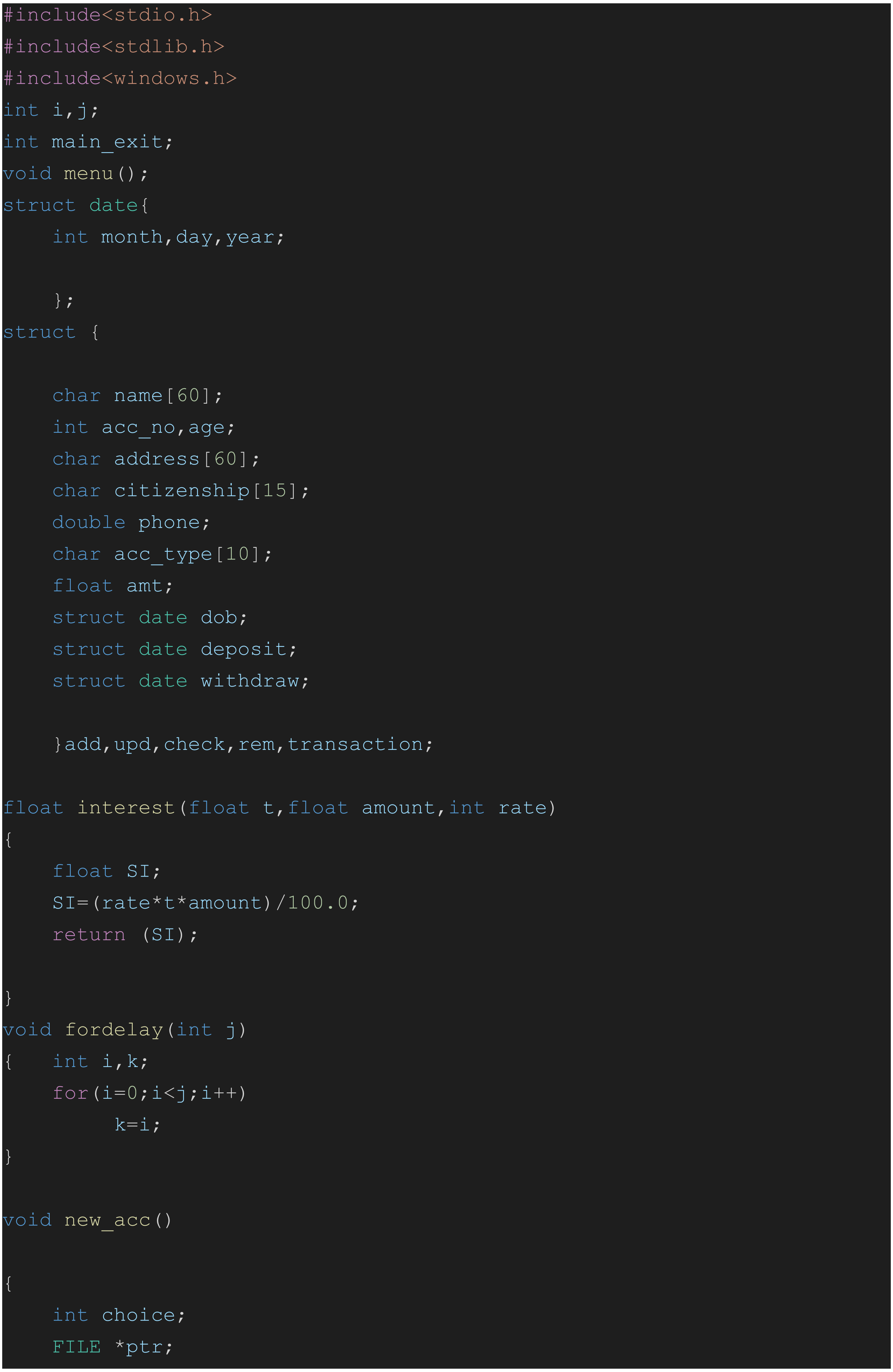
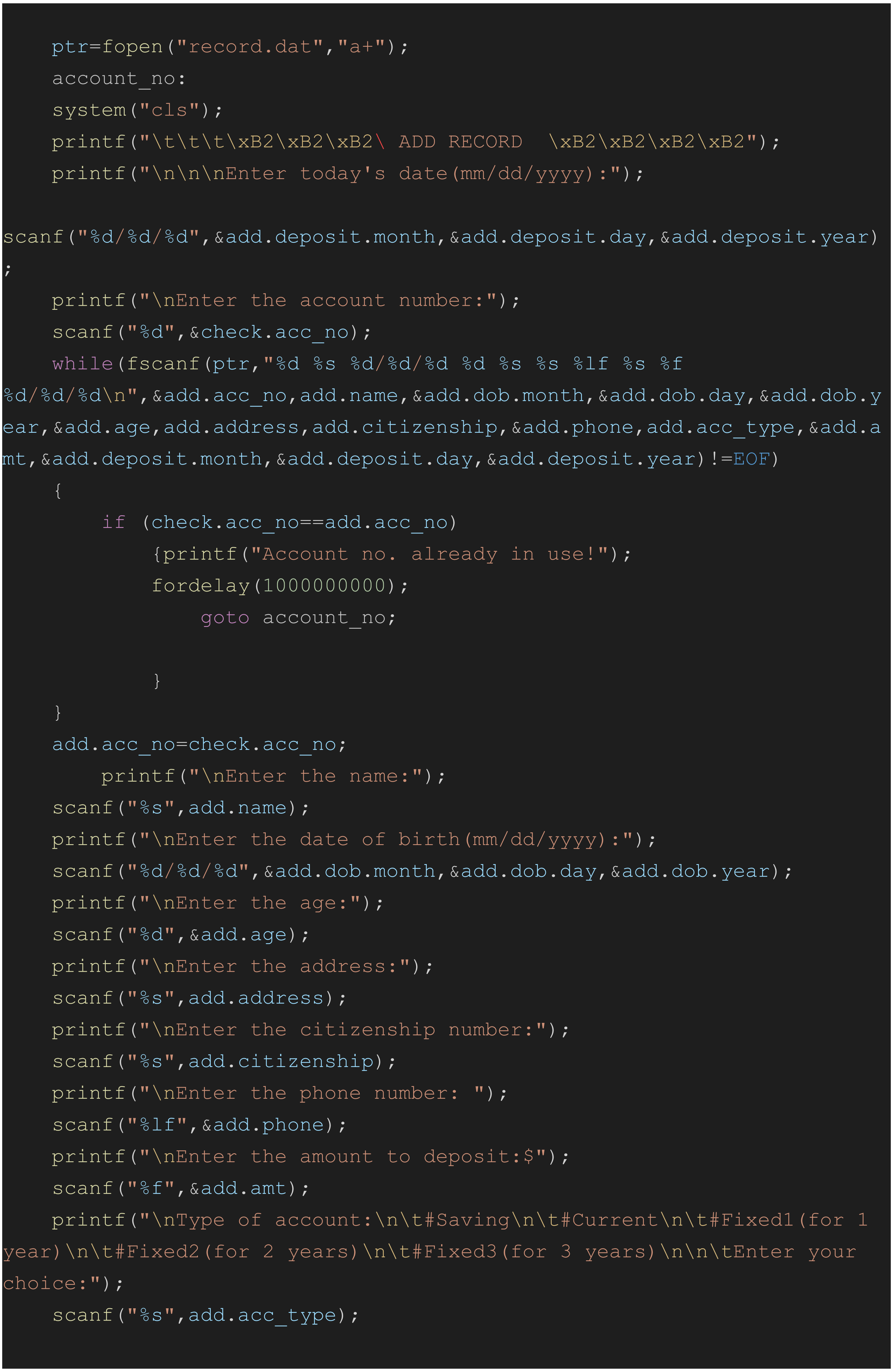


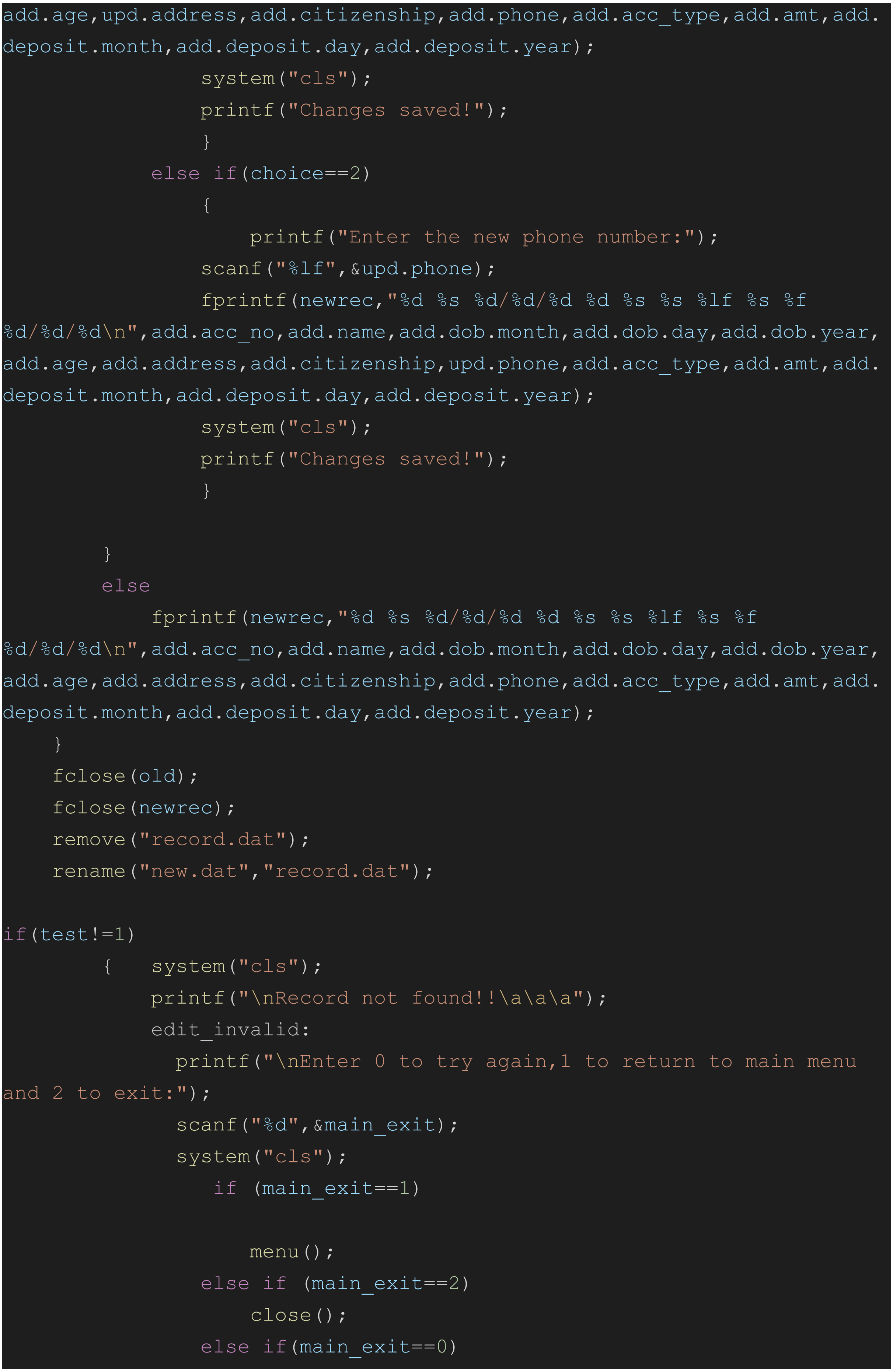
```c
#include<stdio.h>
#include<stdlib.h>
#include<windows.h>
int i,j;
int main_exit;
void menu();
struct date{
    int month,day,year;

    };
struct {

    char name[60];
    int acc_no,age;
    char address[60];
    char citizenship[15];
    double phone;
    char acc_type[10];
    float amt;
    struct date dob;
    struct date deposit;
    struct date withdraw;

    }add,upd,check,rem,transaction;

float interest(float t,float amount,int rate)
{
    float SI;
    SI=(rate*t*amount)/100.0;
    return (SI);

}
void fordelay(int j)
{   int i,k;
    for(i=0;i<j;i++)
         k=i;
}

void new_acc()

{
    int choice;
    FILE *ptr;
```



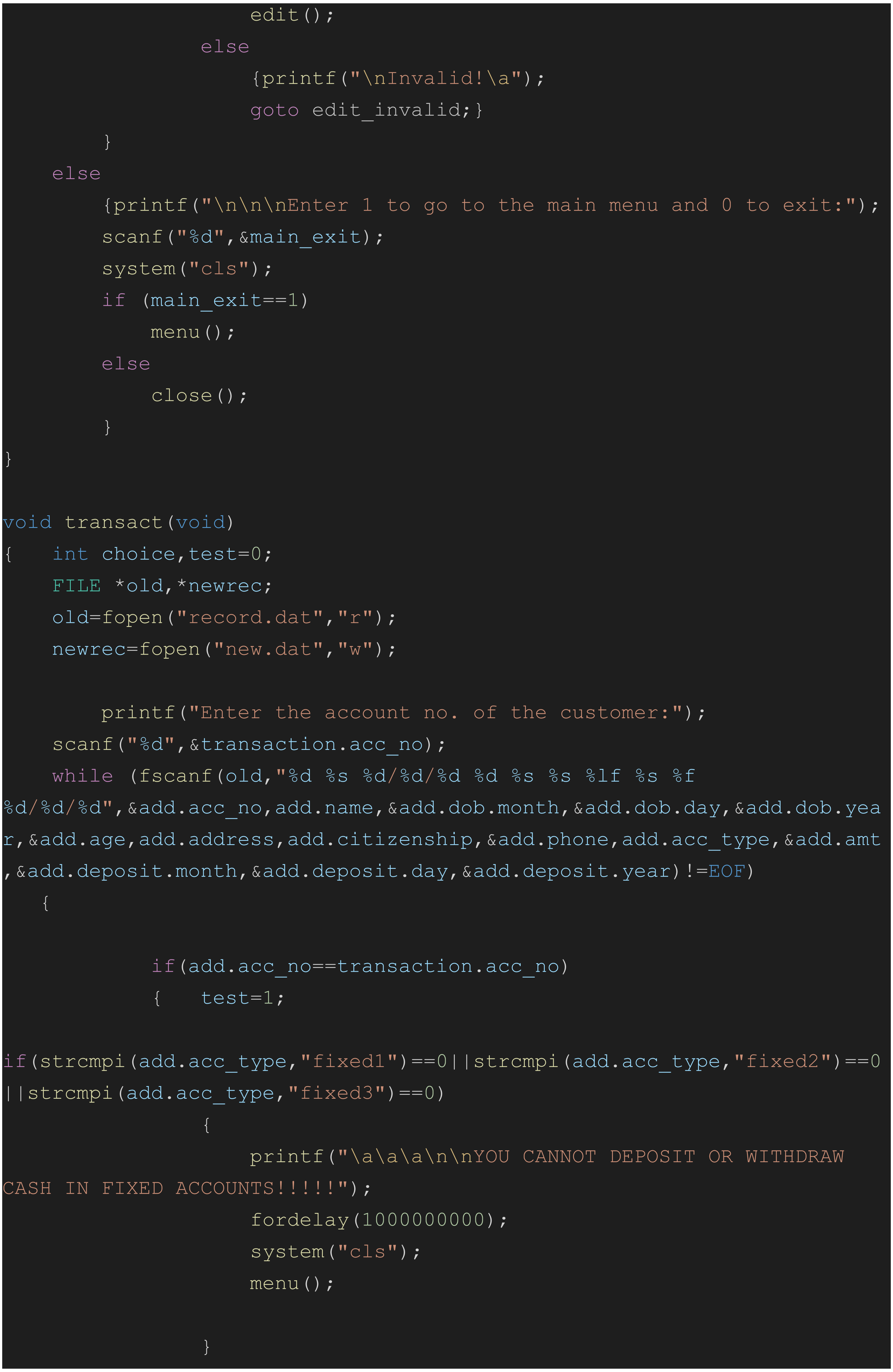
```
    ptr=fopen("record.dat","a+");
    account_no:
    system("cls");
    printf("\t\t\t\xB2\xB2\xB2\ ADD RECORD  \xB2\xB2\xB2\xB2");
    printf("\n\n\nEnter today's date(mm/dd/yyyy):");

scanf("%d/%d/%d",&add.deposit.month,&add.deposit.day,&add.deposit.year)
;
    printf("\nEnter the account number:");
    scanf("%d",&check.acc_no);
    while(fscanf(ptr,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&add.dob.y
ear,&add.age,add.address,add.citizenship,&add.phone,add.acc_type,&add.a
mt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)
    {
        if (check.acc_no==add.acc_no)
            {printf("Account no. already in use!");
            fordelay(1000000000);
                goto account_no;

            }
    }
    add.acc_no=check.acc_no;
        printf("\nEnter the name:");
    scanf("%s",add.name);
    printf("\nEnter the date of birth(mm/dd/yyyy):");
    scanf("%d/%d/%d",&add.dob.month,&add.dob.day,&add.dob.year);
    printf("\nEnter the age:");
    scanf("%d",&add.age);
    printf("\nEnter the address:");
    scanf("%s",add.address);
    printf("\nEnter the citizenship number:");
    scanf("%s",add.citizenship);
    printf("\nEnter the phone number: ");
    scanf("%lf",&add.phone);
    printf("\nEnter the amount to deposit:$");
    scanf("%f",&add.amt);
    printf("\nType of account:\n\t#Saving\n\t#Current\n\t#Fixed1(for 1
year)\n\t#Fixed2(for 2 years)\n\t#Fixed3(for 3 years)\n\n\tEnter your
choice:");
    scanf("%s",add.acc_type);
```

```c
        fprintf(ptr,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);


    fclose(ptr);
    printf("\nAccount created successfully!");
    add_invalid:
    printf("\n\n\n\t\tEnter 1 to go to the main menu and 0 to exit:");
    scanf("%d",&main_exit);
    system("cls");
    if (main_exit==1)
        menu();
    else if(main_exit==0)
            close();
    else
        {
            printf("\nInvalid!\a");
            goto add_invalid;
        }
}
void view_list()
{
    FILE *view;
    view=fopen("record.dat","r");
    int test=0;
    system("cls");
    printf("\nACC. NO.\tNAME\t\t\tADDRESS\t\t\tPHONE\n");

    while(fscanf(view,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&add.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc_type,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)
       {
           printf("\n%6d\t %10s\t\t\t%10s\t\t%.0lf",add.acc_no,add.name,add.address,add.phone);
           test++;
       }

    fclose(view);
    if (test==0)
        {   system("cls");
```

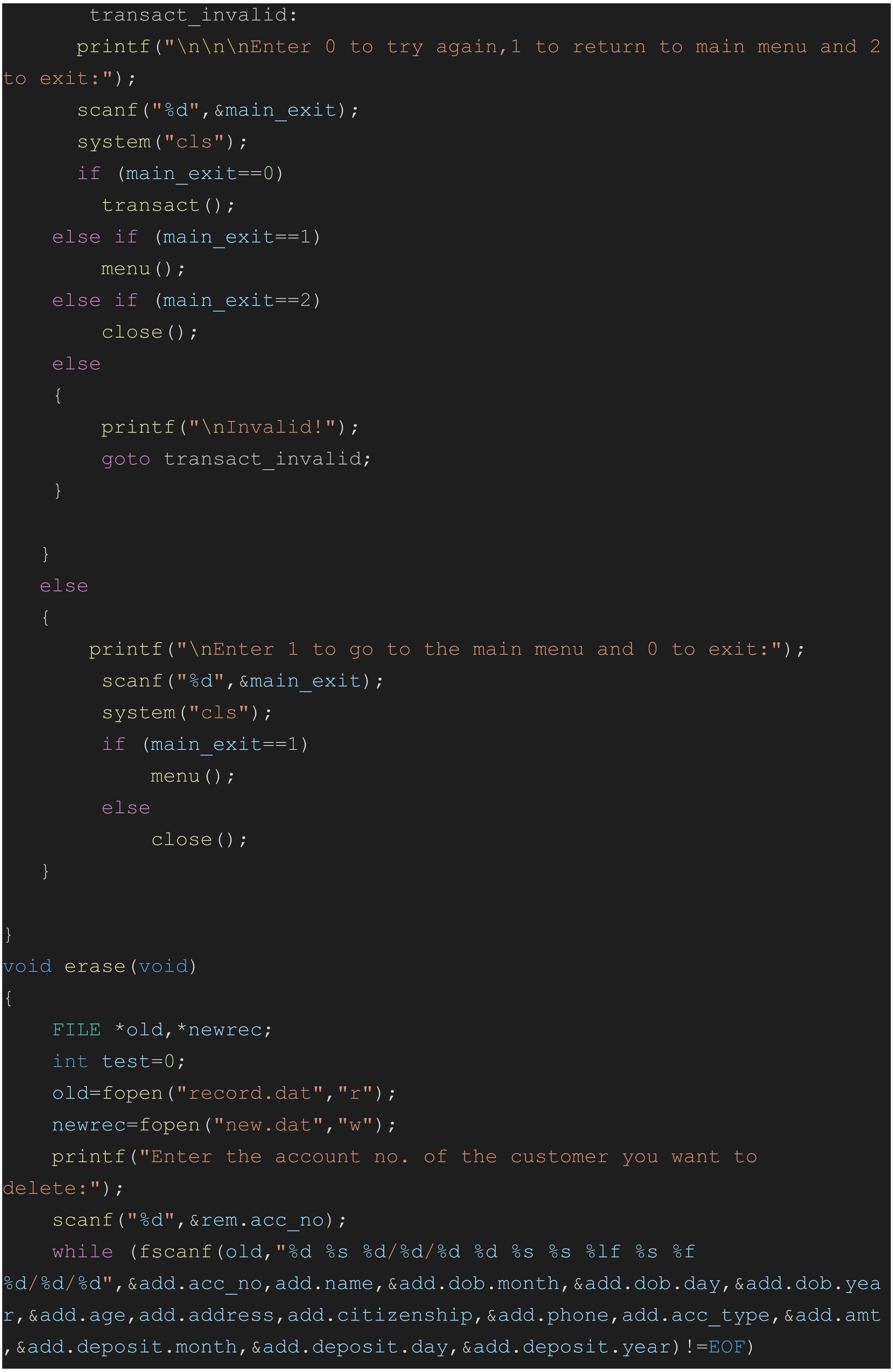
```
            printf("\nNO RECORDS!!\n");}

    view_list_invalid:
     printf("\n\nEnter 1 to go to the main menu and 0 to exit:");
        scanf("%d",&main_exit);
        system("cls");
        if (main_exit==1)
            menu();
        else if(main_exit==0)
            close();
        else
        {
            printf("\nInvalid!\a");
            goto view_list_invalid;
        }
}
void edit(void)
{
    int choice,test=0;
    FILE *old,*newrec;
    old=fopen("record.dat","r");
    newrec=fopen("new.dat","w");

    printf("\nEnter the account no. of the customer whose info you want
to change:");
    scanf("%d",&upd.acc_no);
    while(fscanf(old,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&add.dob.yea
r,&add.age,add.address,add.citizenship,&add.phone,add.acc_type,&add.amt
,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)
    {
        if (add.acc_no==upd.acc_no)
        {   test=1;
            printf("\nWhich information do you want to
change?\n1.Address\n2.Phone\n\nEnter your choice(1 for address and 2
for phone):");
            scanf("%d",&choice);
            system("cls");
            if(choice==1)
                {printf("Enter the new address:");
                scanf("%s",upd.address);
                fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.year,
```



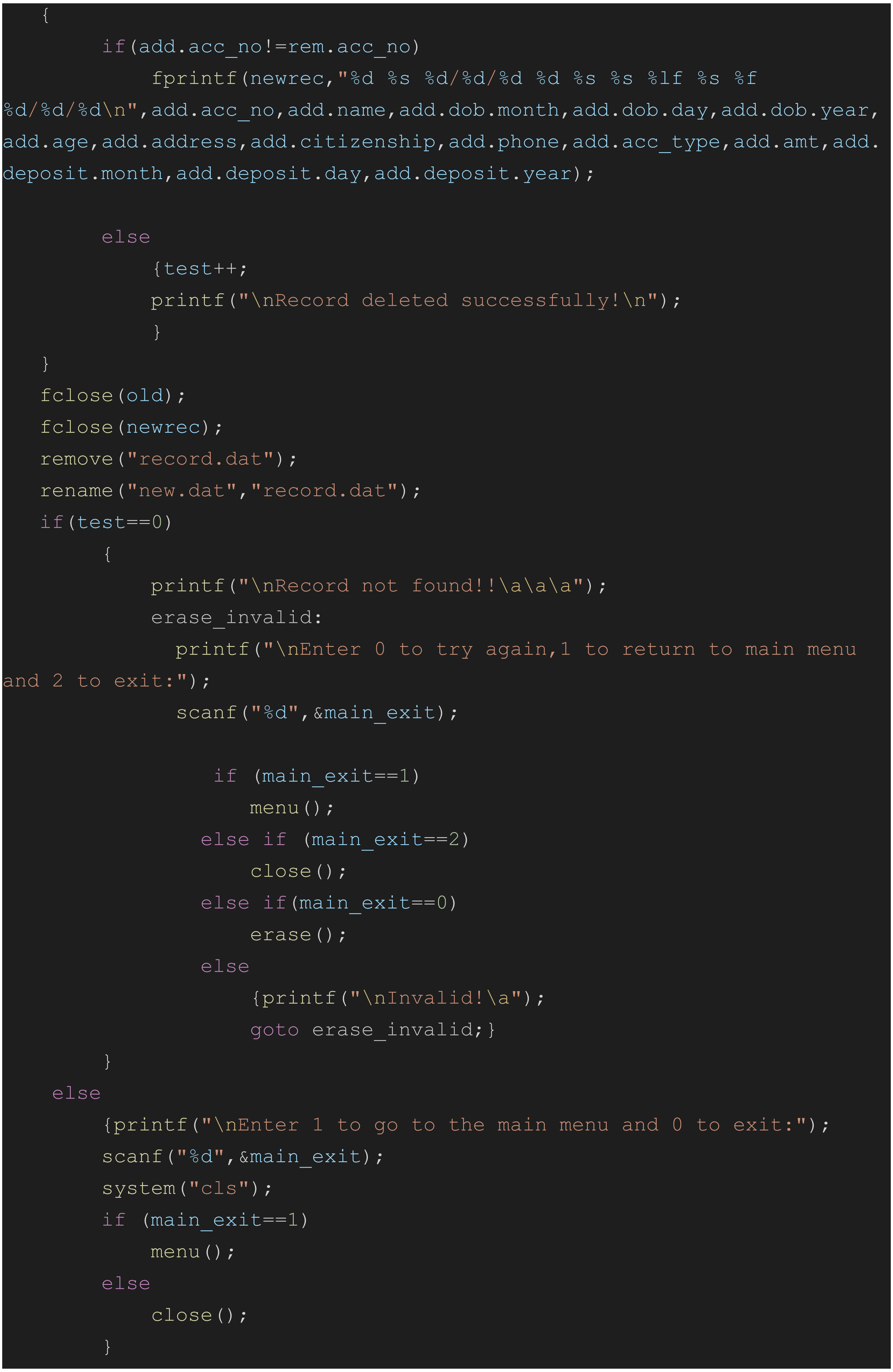
```
add.age,upd.address,add.citizenship,add.phone,add.acc_type,add.amt,add.
deposit.month,add.deposit.day,add.deposit.year);
                system("cls");
                printf("Changes saved!");
                }
            else if(choice==2)
                {
                    printf("Enter the new phone number:");
                scanf("%lf",&upd.phone);
                fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.year,
add.age,add.address,add.citizenship,upd.phone,add.acc_type,add.amt,add.
deposit.month,add.deposit.day,add.deposit.year);
                system("cls");
                printf("Changes saved!");
                }

        }
        else
            fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.year,
add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,add.
deposit.month,add.deposit.day,add.deposit.year);
    }
    fclose(old);
    fclose(newrec);
    remove("record.dat");
    rename("new.dat","record.dat");

if(test!=1)
        {   system("cls");
            printf("\nRecord not found!!\a\a\a");
            edit_invalid:
              printf("\nEnter 0 to try again,1 to return to main menu
and 2 to exit:");
              scanf("%d",&main_exit);
              system("cls");
                 if (main_exit==1)

                    menu();
                else if (main_exit==2)
                    close();
                else if(main_exit==0)
```



```
                    edit();
                else
                    {printf("\nInvalid!\a");
                    goto edit_invalid;}
        }
    else
        {printf("\n\n\nEnter 1 to go to the main menu and 0 to exit:");
        scanf("%d",&main_exit);
        system("cls");
        if (main_exit==1)
            menu();
        else
            close();
        }
}

void transact(void)
{   int choice,test=0;
    FILE *old,*newrec;
    old=fopen("record.dat","r");
    newrec=fopen("new.dat","w");

        printf("Enter the account no. of the customer:");
    scanf("%d",&transaction.acc_no);
    while (fscanf(old,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&add.dob.yea
r,&add.age,add.address,add.citizenship,&add.phone,add.acc_type,&add.amt
,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)
   {

            if(add.acc_no==transaction.acc_no)
            {   test=1;

if(strcmpi(add.acc_type,"fixed1")==0||strcmpi(add.acc_type,"fixed2")==0
||strcmpi(add.acc_type,"fixed3")==0)
                {
                    printf("\a\a\a\n\nYOU CANNOT DEPOSIT OR WITHDRAW
CASH IN FIXED ACCOUNTS!!!!!");
                    fordelay(1000000000);
                    system("cls");
                    menu();

                }
                printf("\n\nDo you want to\n1.Deposit\n2.Withdraw?\n\nEnter your choice(1 for deposit and 2 for withdraw):");
                scanf("%d",&choice);
                if (choice==1)
                {
                    printf("Enter the amount you want to deposit:$ ");
                    scanf("%f",&transaction.amt);
                    add.amt+=transaction.amt;
                    fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);
                    printf("\n\nDeposited successfully!");
                }
                else
                {
                    printf("Enter the amount you want to withdraw:$ ");
                    scanf("%f",&transaction.amt);
                    add.amt-=transaction.amt;
                    fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);
                    printf("\n\nWithdrawn successfully!");
                }

            }
            else
            {
               fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);
            }
   }
   fclose(old);
   fclose(newrec);
   remove("record.dat");
   rename("new.dat","record.dat");
   if(test!=1)
   {
       printf("\n\nRecord not found!!");
```

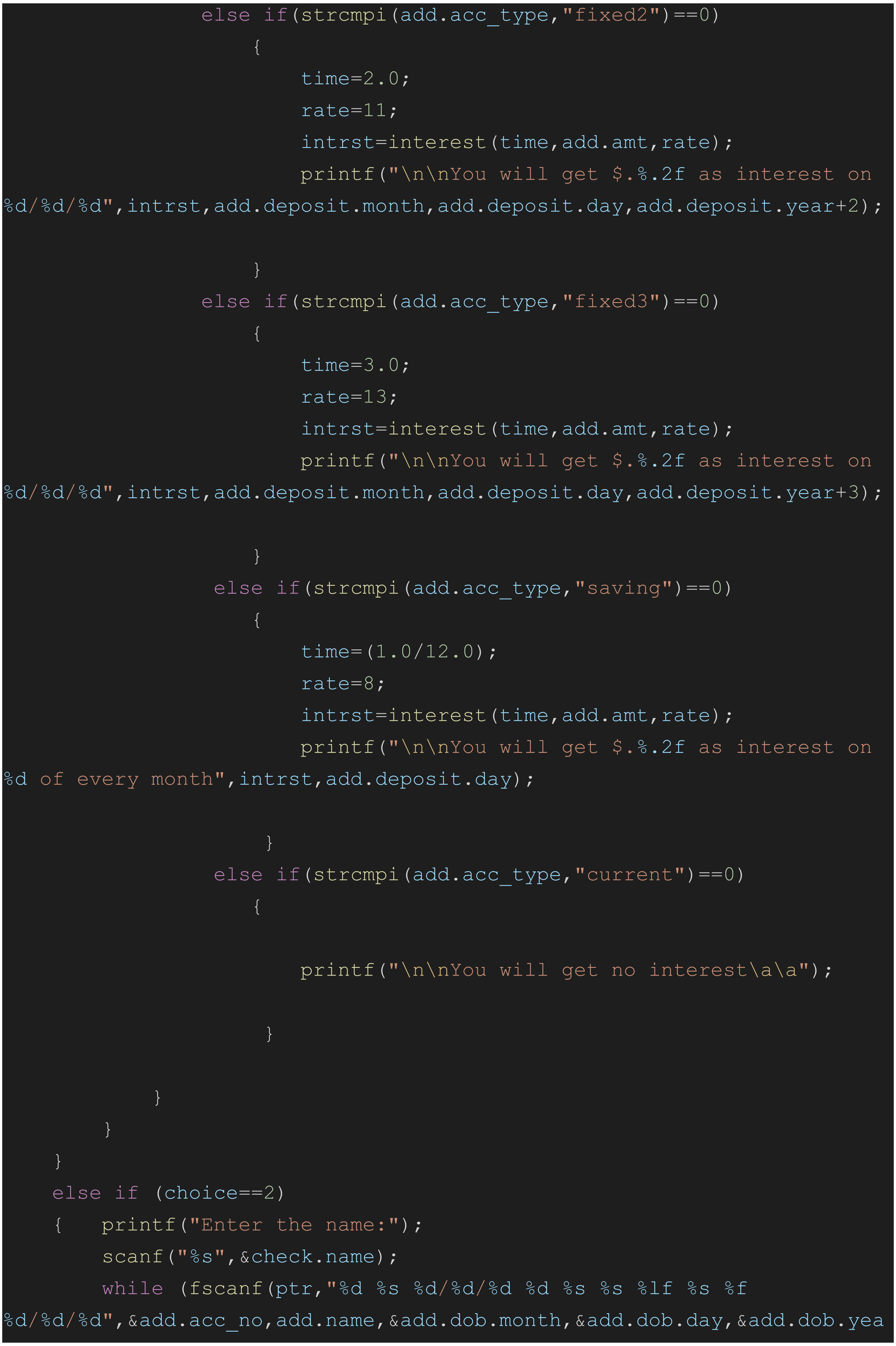


```
       transact_invalid:
      printf("\n\n\nEnter 0 to try again,1 to return to main menu and 2
to exit:");
      scanf("%d",&main_exit);
      system("cls");
      if (main_exit==0)
        transact();
    else if (main_exit==1)
        menu();
    else if (main_exit==2)
        close();
    else
    {
        printf("\nInvalid!");
        goto transact_invalid;
    }

   }
   else
   {
       printf("\nEnter 1 to go to the main menu and 0 to exit:");
        scanf("%d",&main_exit);
        system("cls");
        if (main_exit==1)
            menu();
        else
            close();
   }

}
void erase(void)
{
    FILE *old,*newrec;
    int test=0;
    old=fopen("record.dat","r");
    newrec=fopen("new.dat","w");
    printf("Enter the account no. of the customer you want to
delete:");
    scanf("%d",&rem.acc_no);
    while (fscanf(old,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&add.dob.yea
r,&add.age,add.address,add.citizenship,&add.phone,add.acc_type,&add.amt
,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)
```



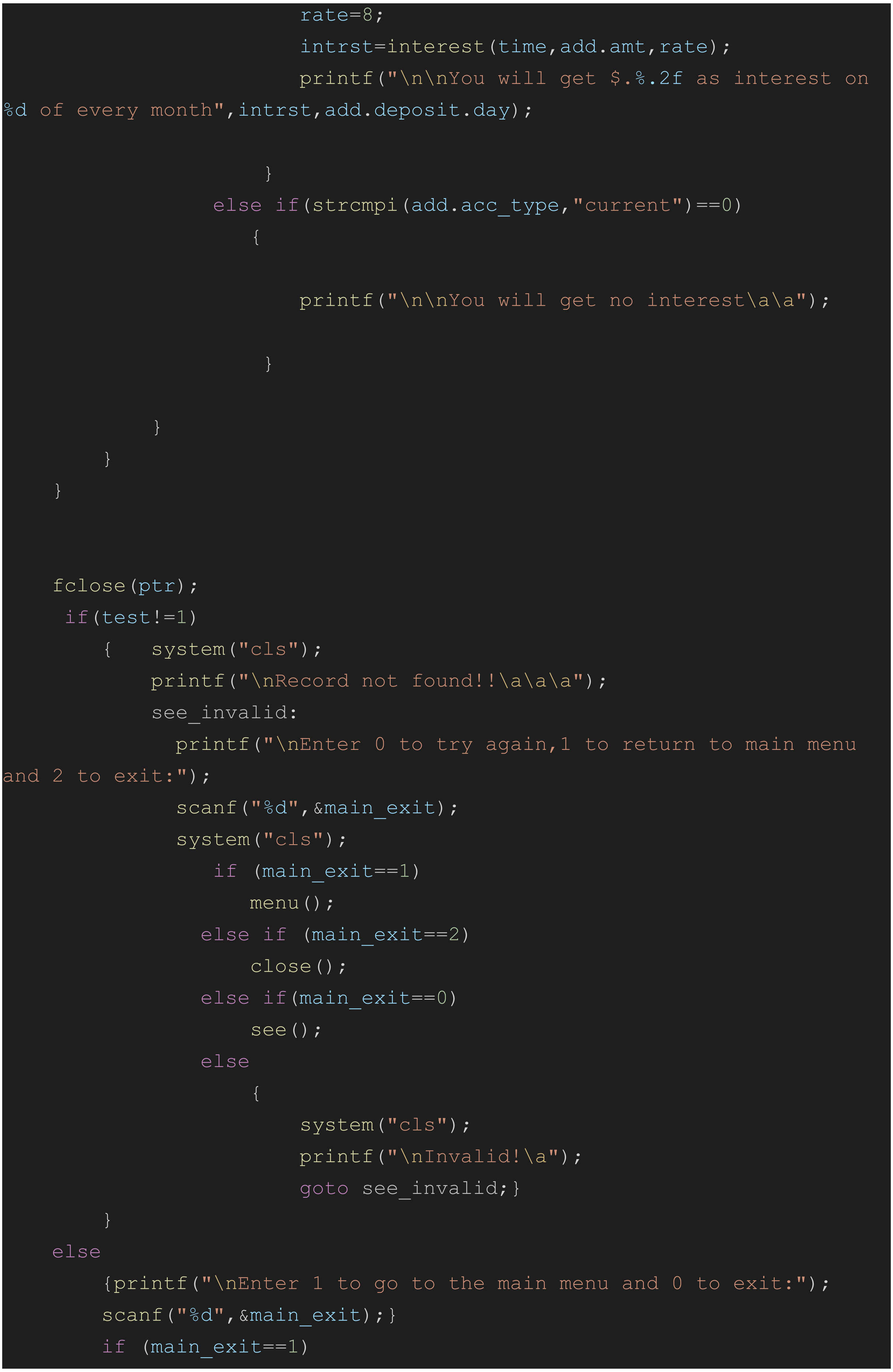
```
   {
        if(add.acc_no!=rem.acc_no)
            fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.year,
add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,add.
deposit.month,add.deposit.day,add.deposit.year);

        else
            {test++;
            printf("\nRecord deleted successfully!\n");
            }
   }
   fclose(old);
   fclose(newrec);
   remove("record.dat");
   rename("new.dat","record.dat");
   if(test==0)
        {
            printf("\nRecord not found!!\a\a\a");
            erase_invalid:
              printf("\nEnter 0 to try again,1 to return to main menu
and 2 to exit:");
              scanf("%d",&main_exit);

                 if (main_exit==1)
                    menu();
                else if (main_exit==2)
                    close();
                else if(main_exit==0)
                    erase();
                else
                    {printf("\nInvalid!\a");
                    goto erase_invalid;}
        }
    else
        {printf("\nEnter 1 to go to the main menu and 0 to exit:");
        scanf("%d",&main_exit);
        system("cls");
        if (main_exit==1)
            menu();
        else
            close();
        }
```

```
}

void see(void)
{
    FILE *ptr;
    int test=0,rate;
    int choice;
    float time;
    float intrst;
    ptr=fopen("record.dat","r");
    printf("Do you want to check by\n1.Account no\n2.Name\nEnter your
choice:");
    scanf("%d",&choice);
    if (choice==1)
    {   printf("Enter the account number:");
        scanf("%d",&check.acc_no);

        while (fscanf(ptr,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&add.dob.yea
r,&add.age,add.address,add.citizenship,&add.phone,add.acc_type,&add.amt
,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)
        {
            if(add.acc_no==check.acc_no)
            {   system("cls");
                test=1;

                printf("\nAccount NO.:%d\nName:%s \nDOB:%d/%d/%d
\nAge:%d \nAddress:%s \nCitizenship No:%s \nPhone number:%.0lf \nType
Of Account:%s \nAmount deposited:$ %.2f \nDate Of
Deposit:%d/%d/%d\n\n",add.acc_no,add.name,add.dob.month,add.dob.day,add
.dob.year,add.age,add.address,add.citizenship,add.phone,

add.acc_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year
);
                if(strcmpi(add.acc_type,"fixed1")==0)
                    {
                        time=1.0;
                        rate=9;
                        intrst=interest(time,add.amt,rate);
                        printf("\n\nYou will get $%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+1);
                    }
```

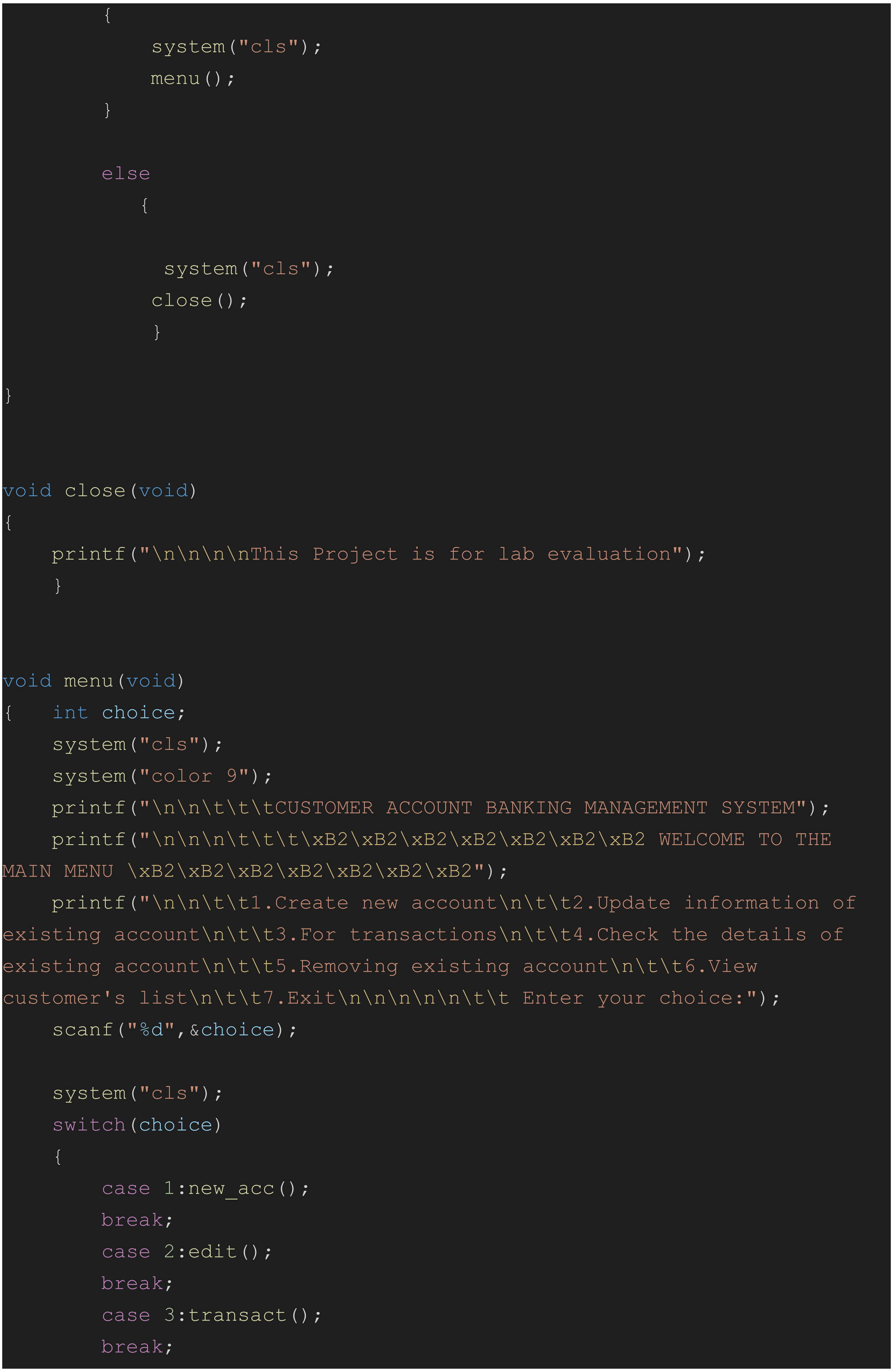


```
                else if(strcmpi(add.acc_type,"fixed2")==0)
                    {
                        time=2.0;
                        rate=11;
                        intrst=interest(time,add.amt,rate);
                        printf("\n\nYou will get $.%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+2);

                    }
                else if(strcmpi(add.acc_type,"fixed3")==0)
                    {
                        time=3.0;
                        rate=13;
                        intrst=interest(time,add.amt,rate);
                        printf("\n\nYou will get $.%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+3);

                    }
                 else if(strcmpi(add.acc_type,"saving")==0)
                    {
                        time=(1.0/12.0);
                        rate=8;
                        intrst=interest(time,add.amt,rate);
                        printf("\n\nYou will get $.%.2f as interest on
%d of every month",intrst,add.deposit.day);

                     }
                 else if(strcmpi(add.acc_type,"current")==0)
                    {

                        printf("\n\nYou will get no interest\a\a");

                     }

            }
        }
    }
    else if (choice==2)
    {   printf("Enter the name:");
        scanf("%s",&check.name);
        while (fscanf(ptr,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&add.dob.yea
```

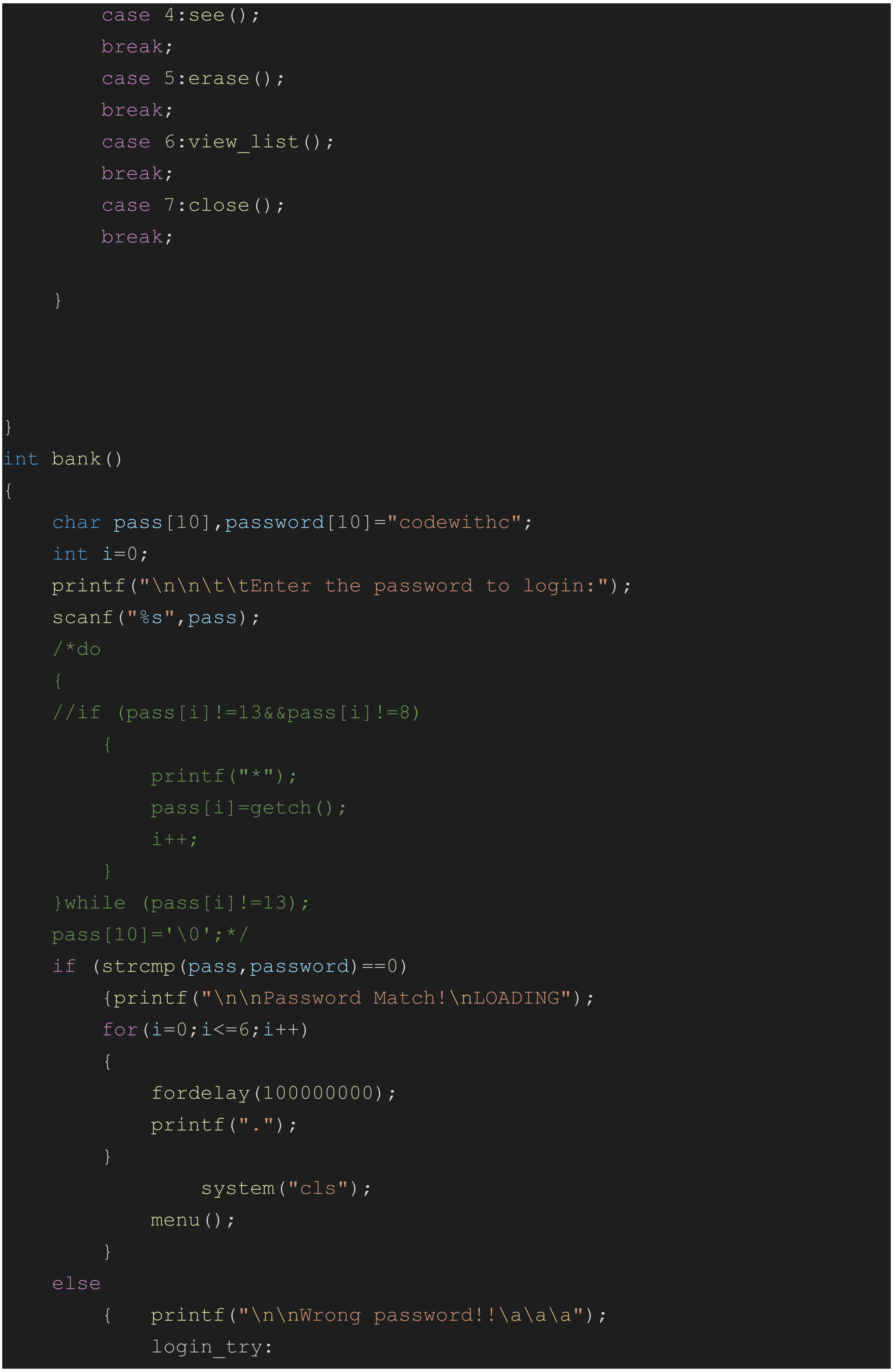
```
r,&add.age,add.address,add.citizenship,&add.phone,add.acc_type,&add.amt
,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)
        {
            if(strcmpi(add.name,check.name)==0)
            {   system("cls");
                test=1;
                printf("\nAccount No.:%d\nName:%s \nDOB:%d/%d/%d
\nAge:%d \nAddress:%s \nCitizenship No:%s \nPhone number:%.0lf \nType
Of Account:%s \nAmount deposited:$%.2f \nDate Of
Deposit:%d/%d/%d\n\n",add.acc_no,add.name,add.dob.month,add.dob.day,add
.dob.year,add.age,add.address,add.citizenship,add.phone,

add.acc_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year
);
                if(strcmpi(add.acc_type,"fixed1")==0)
                    {
                        time=1.0;
                        rate=9;
                        intrst=interest(time,add.amt,rate);
                        printf("\n\nYou will get $.%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+1);
                    }
                else if(strcmpi(add.acc_type,"fixed2")==0)
                    {
                        time=2.0;
                        rate=11;
                        intrst=interest(time,add.amt,rate);
                        printf("\n\nYou will get $.%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+2);

                    }
                else if(strcmpi(add.acc_type,"fixed3")==0)
                    {
                        time=3.0;
                        rate=13;
                        intrst=interest(time,add.amt,rate);
                        printf("\n\nYou will get $.%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+3);

                    }
                 else if(strcmpi(add.acc_type,"saving")==0)
                    {
                        time=(1.0/12.0);
```



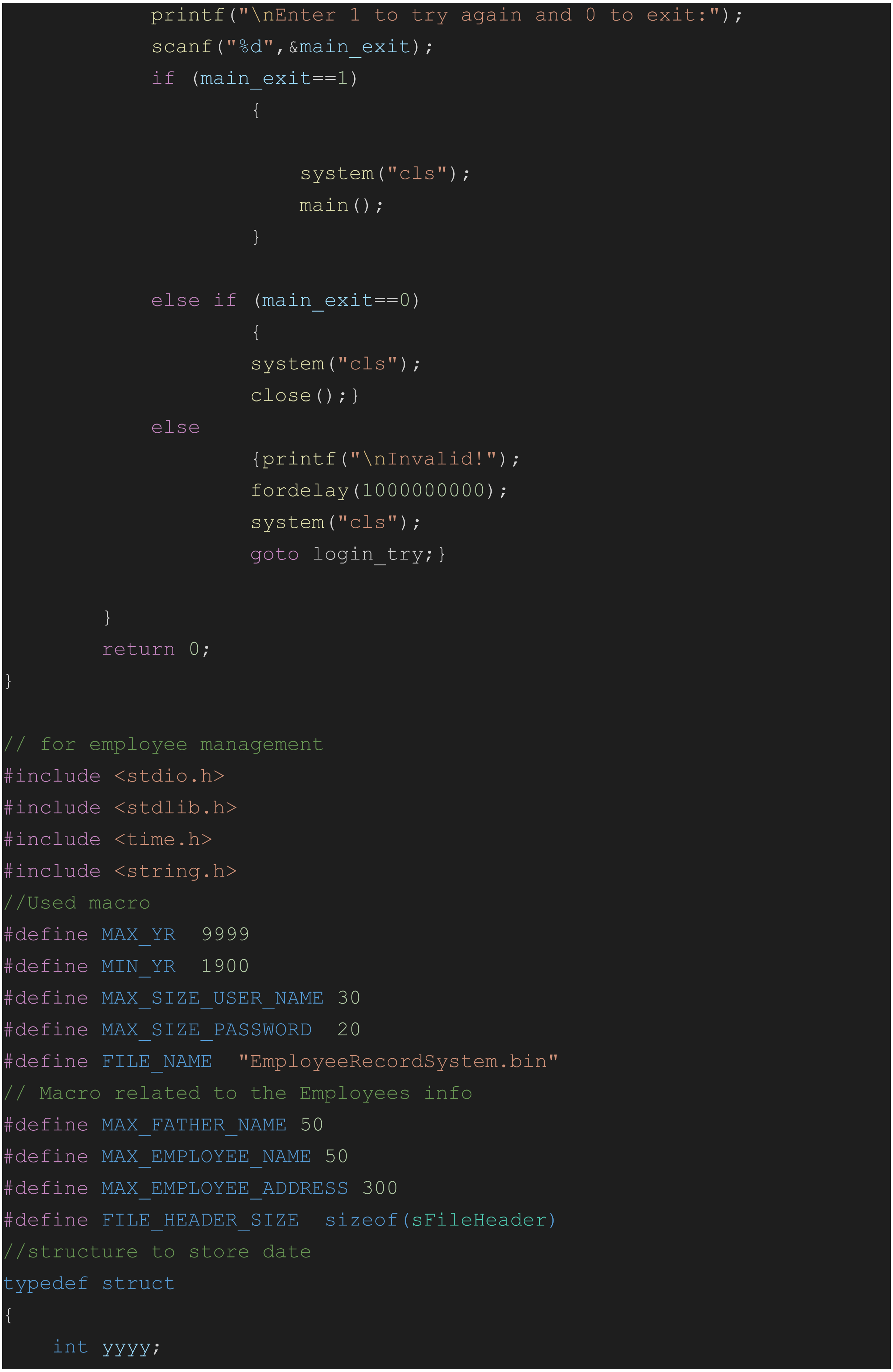
```
                        rate=8;
                        intrst=interest(time,add.amt,rate);
                        printf("\n\nYou will get $.%.2f as interest on
%d of every month",intrst,add.deposit.day);

                     }
                 else if(strcmpi(add.acc_type,"current")==0)
                    {

                        printf("\n\nYou will get no interest\a\a");

                     }

            }
        }
    }


    fclose(ptr);
     if(test!=1)
        {   system("cls");
            printf("\nRecord not found!!\a\a\a");
            see_invalid:
              printf("\nEnter 0 to try again,1 to return to main menu
and 2 to exit:");
              scanf("%d",&main_exit);
              system("cls");
                 if (main_exit==1)
                    menu();
                else if (main_exit==2)
                    close();
                else if(main_exit==0)
                    see();
                else
                    {
                        system("cls");
                        printf("\nInvalid!\a");
                        goto see_invalid;}
        }
    else
        {printf("\nEnter 1 to go to the main menu and 0 to exit:");
        scanf("%d",&main_exit);}
        if (main_exit==1)
```



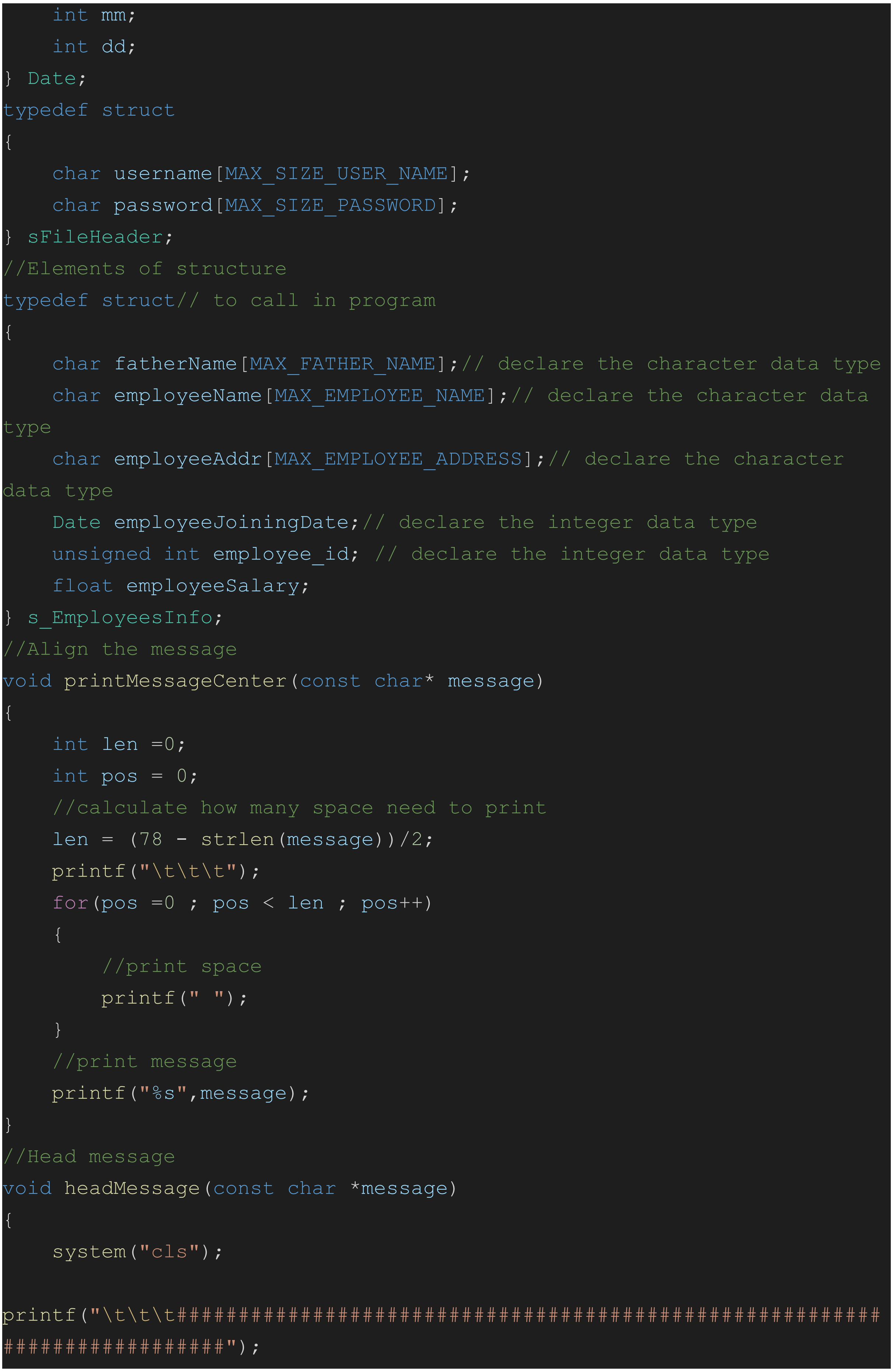
```c
        {
            system("cls");
            menu();
        }

        else
           {

             system("cls");
            close();
            }

}



void close(void)
{
    printf("\n\n\n\nThis Project is for lab evaluation");
    }



void menu(void)
{   int choice;
    system("cls");
    system("color 9");
    printf("\n\n\t\t\tCUSTOMER ACCOUNT BANKING MANAGEMENT SYSTEM");
    printf("\n\n\n\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2 WELCOME TO THE MAIN MENU \xB2\xB2\xB2\xB2\xB2\xB2\xB2");
    printf("\n\n\t\t1.Create new account\n\t\t2.Update information of existing account\n\t\t3.For transactions\n\t\t4.Check the details of existing account\n\t\t5.Removing existing account\n\t\t6.View customer's list\n\t\t7.Exit\n\n\n\n\n\t\t Enter your choice:");
    scanf("%d",&choice);

    system("cls");
    switch(choice)
    {
        case 1:new_acc();
        break;
        case 2:edit();
        break;
        case 3:transact();
        break;
```



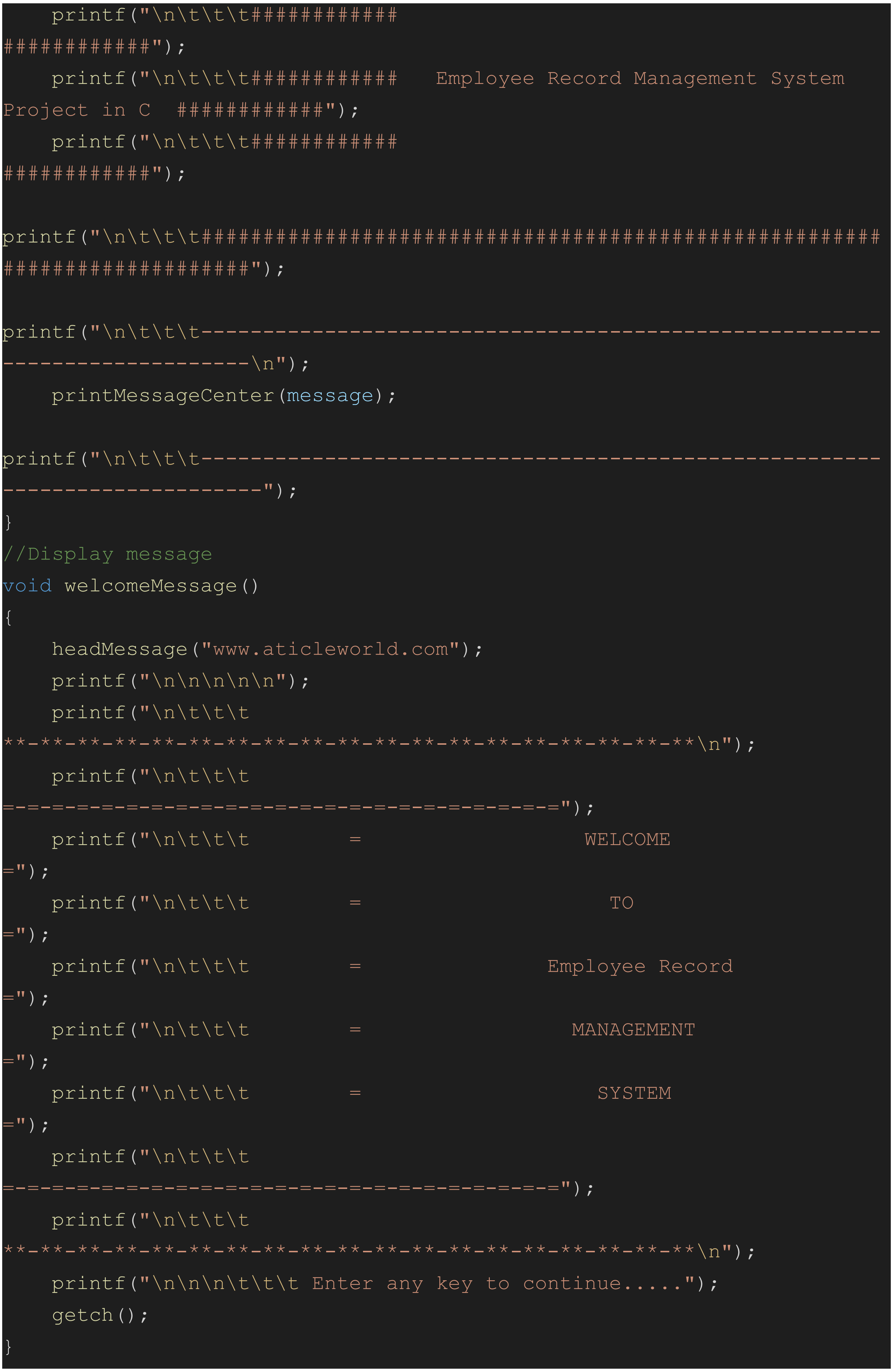
```
        case 4:see();
        break;
        case 5:erase();
        break;
        case 6:view_list();
        break;
        case 7:close();
        break;

    }




}
int bank()
{
    char pass[10],password[10]="codewithc";
    int i=0;
    printf("\n\n\t\tEnter the password to login:");
    scanf("%s",pass);
    /*do
    {
    //if (pass[i]!=13&&pass[i]!=8)
        {
            printf("*");
            pass[i]=getch();
            i++;
        }
    }while (pass[i]!=13);
    pass[10]='\0';*/
    if (strcmp(pass,password)==0)
        {printf("\n\nPassword Match!\nLOADING");
        for(i=0;i<=6;i++)
        {
            fordelay(100000000);
            printf(".");
        }
                system("cls");
            menu();
        }
    else
        {   printf("\n\nWrong password!!\a\a\a");
            login_try:
```



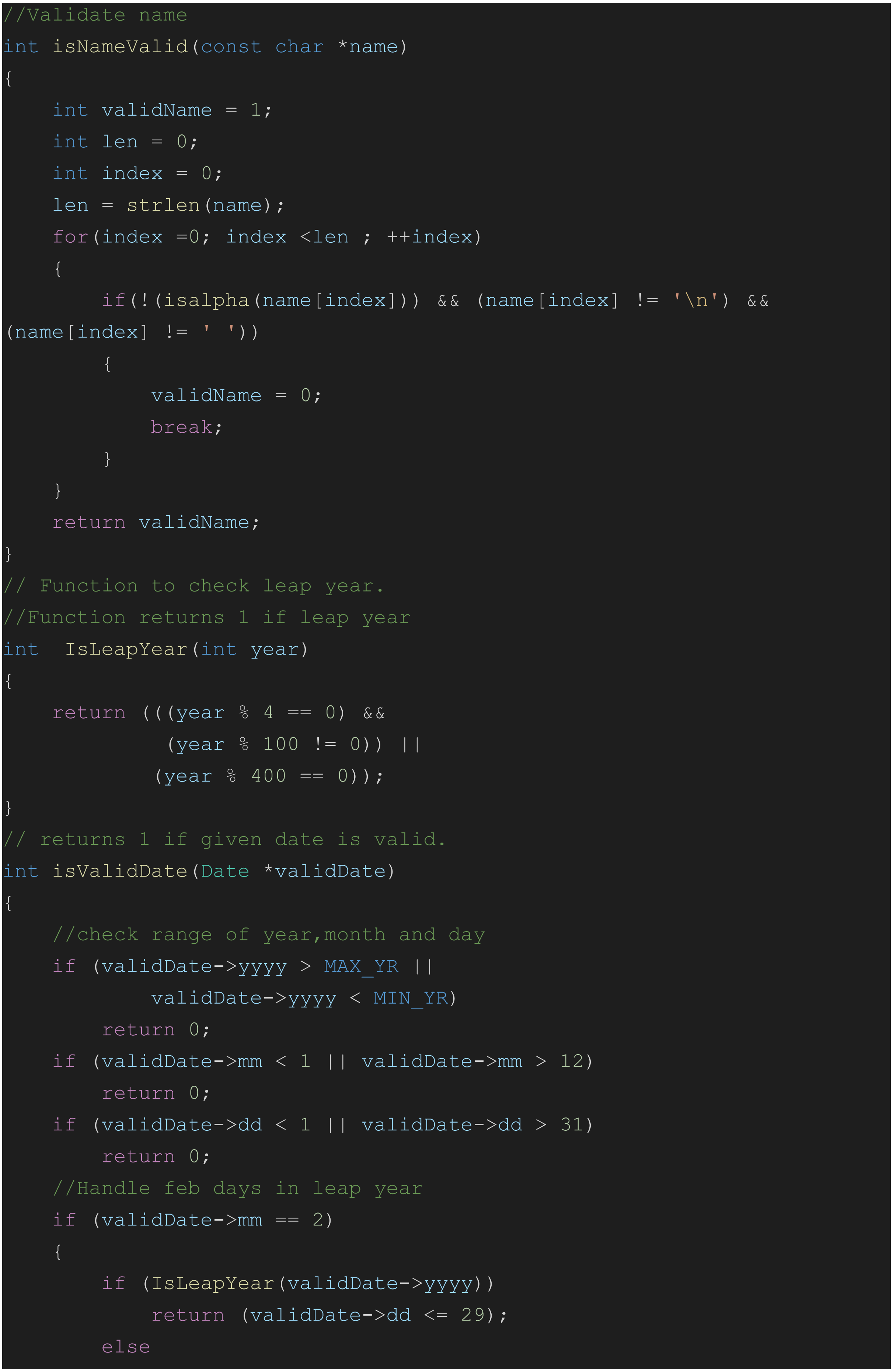
```
            printf("\nEnter 1 to try again and 0 to exit:");
            scanf("%d",&main_exit);
            if (main_exit==1)
                    {

                        system("cls");
                        main();
                    }

            else if (main_exit==0)
                    {
                    system("cls");
                    close();}
            else
                    {printf("\nInvalid!");
                    fordelay(1000000000);
                    system("cls");
                    goto login_try;}

        }
        return 0;
}

// for employee management
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
//Used macro
#define MAX_YR  9999
#define MIN_YR  1900
#define MAX_SIZE_USER_NAME 30
#define MAX_SIZE_PASSWORD  20
#define FILE_NAME  "EmployeeRecordSystem.bin"
// Macro related to the Employees info
#define MAX_FATHER_NAME 50
#define MAX_EMPLOYEE_NAME 50
#define MAX_EMPLOYEE_ADDRESS 300
#define FILE_HEADER_SIZE  sizeof(sFileHeader)
//structure to store date
typedef struct
{
    int yyyy;
```



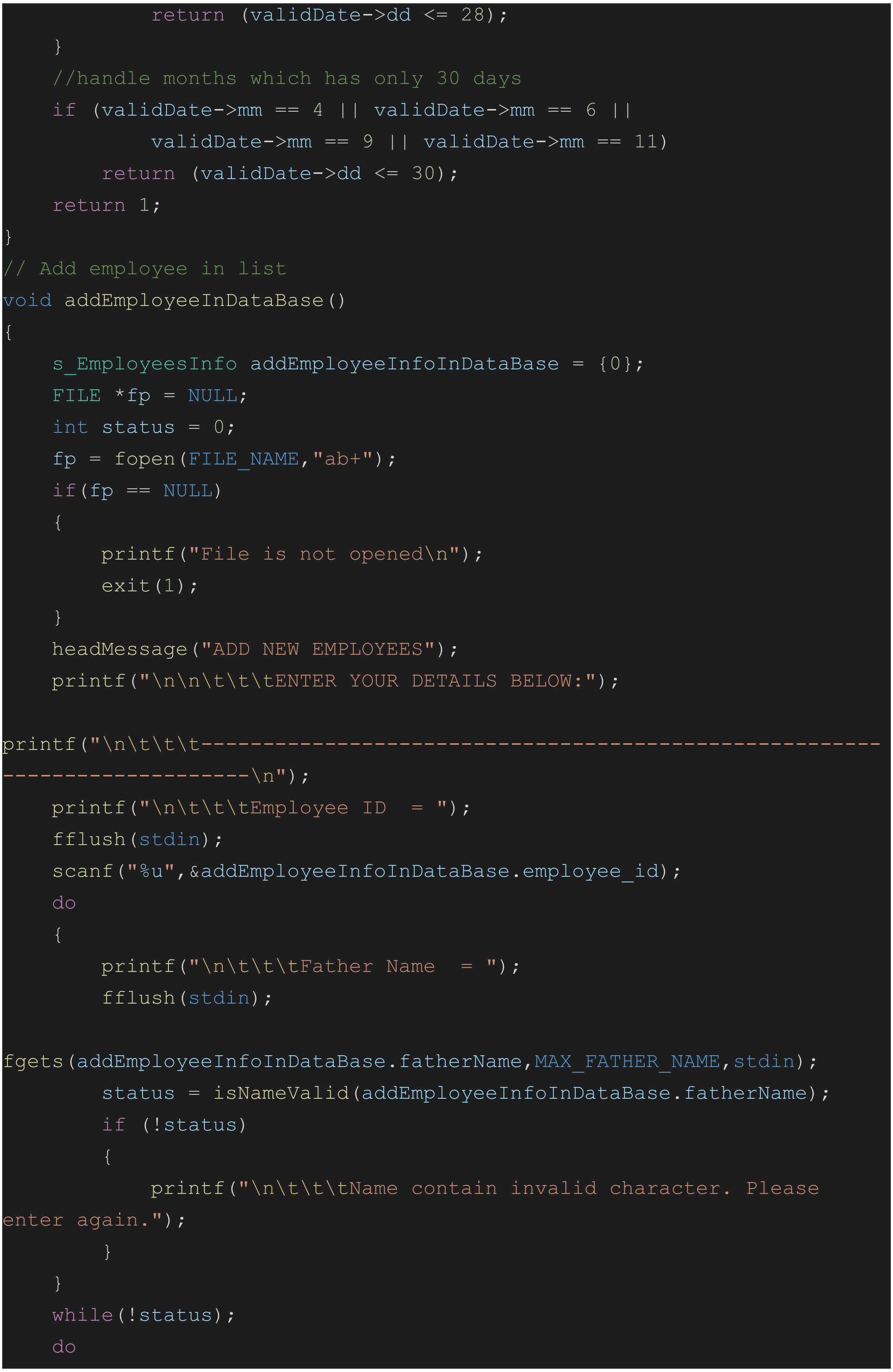
```c
    int mm;
    int dd;
} Date;
typedef struct
{
    char username[MAX_SIZE_USER_NAME];
    char password[MAX_SIZE_PASSWORD];
} sFileHeader;
//Elements of structure
typedef struct// to call in program
{
    char fatherName[MAX_FATHER_NAME];// declare the character data type
    char employeeName[MAX_EMPLOYEE_NAME];// declare the character data type
    char employeeAddr[MAX_EMPLOYEE_ADDRESS];// declare the character data type
    Date employeeJoiningDate;// declare the integer data type
    unsigned int employee_id; // declare the integer data type
    float employeeSalary;
} s_EmployeesInfo;
//Align the message
void printMessageCenter(const char* message)
{
    int len =0;
    int pos = 0;
    //calculate how many space need to print
    len = (78 - strlen(message))/2;
    printf("\t\t\t");
    for(pos =0 ; pos < len ; pos++)
    {
        //print space
        printf(" ");
    }
    //print message
    printf("%s",message);
}
//Head message
void headMessage(const char *message)
{
    system("cls");

printf("\t\t\t###########################################################################");
```



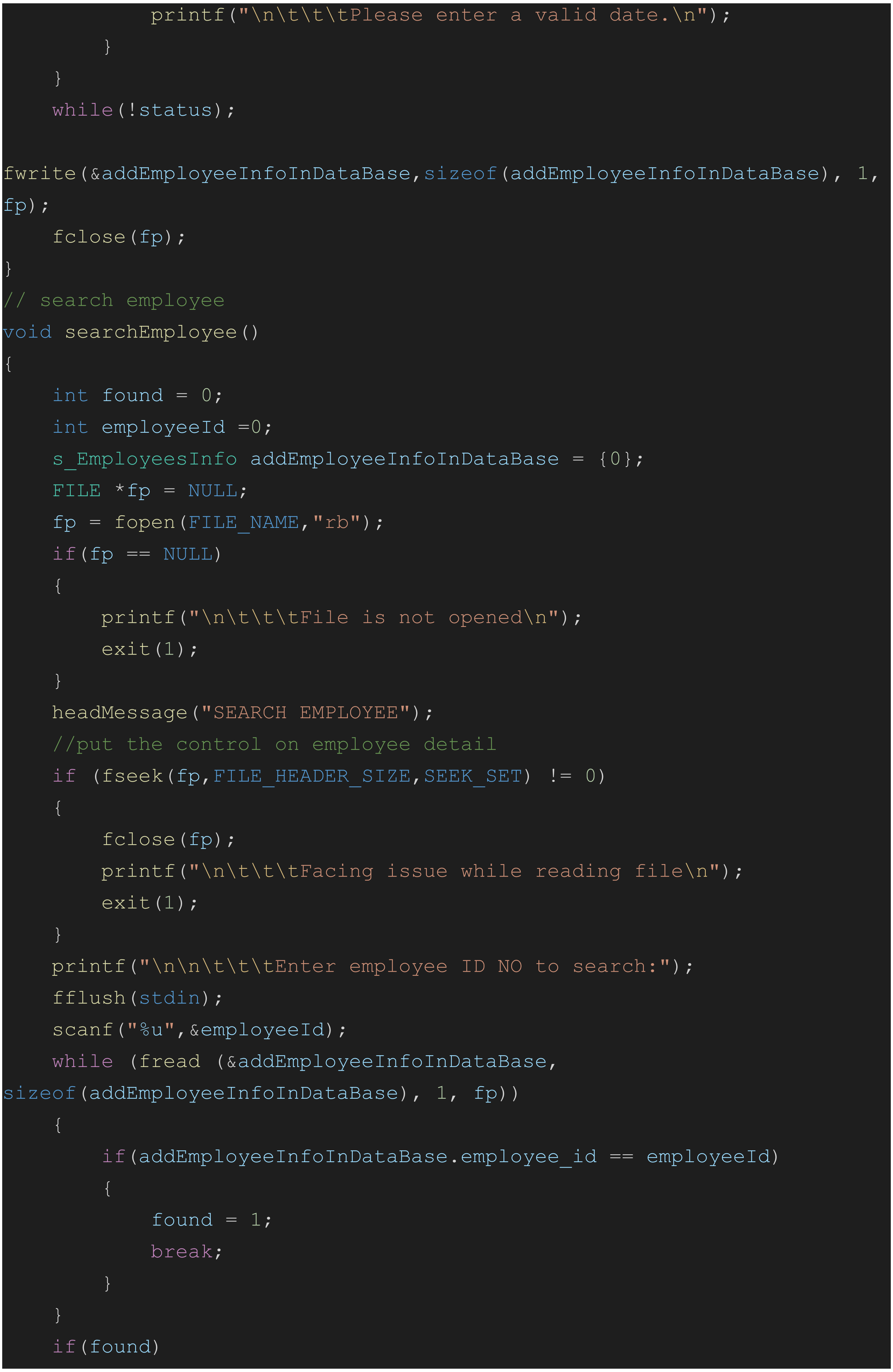
```
    printf("\n\t\t\t############
############");
    printf("\n\t\t\t############   Employee Record Management System
Project in C  ############");
    printf("\n\t\t\t############
############");

printf("\n\t\t\t#######################################################
####################");

printf("\n\t\t\t-------------------------------------------------------
--------------------\n");
    printMessageCenter(message);

printf("\n\t\t\t-------------------------------------------------------
---------------------");
}
//Display message
void welcomeMessage()
{
    headMessage("www.aticleworld.com");
    printf("\n\n\n\n\n");
    printf("\n\t\t\t
**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**\n");
    printf("\n\t\t\t
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=");
    printf("\n\t\t\t        =                  WELCOME
=");
    printf("\n\t\t\t        =                    TO
=");
    printf("\n\t\t\t        =               Employee Record
=");
    printf("\n\t\t\t        =                 MANAGEMENT
=");
    printf("\n\t\t\t        =                   SYSTEM
=");
    printf("\n\t\t\t
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=");
    printf("\n\t\t\t
**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**\n");
    printf("\n\n\n\t\t\t Enter any key to continue.....");
    getch();
}
```



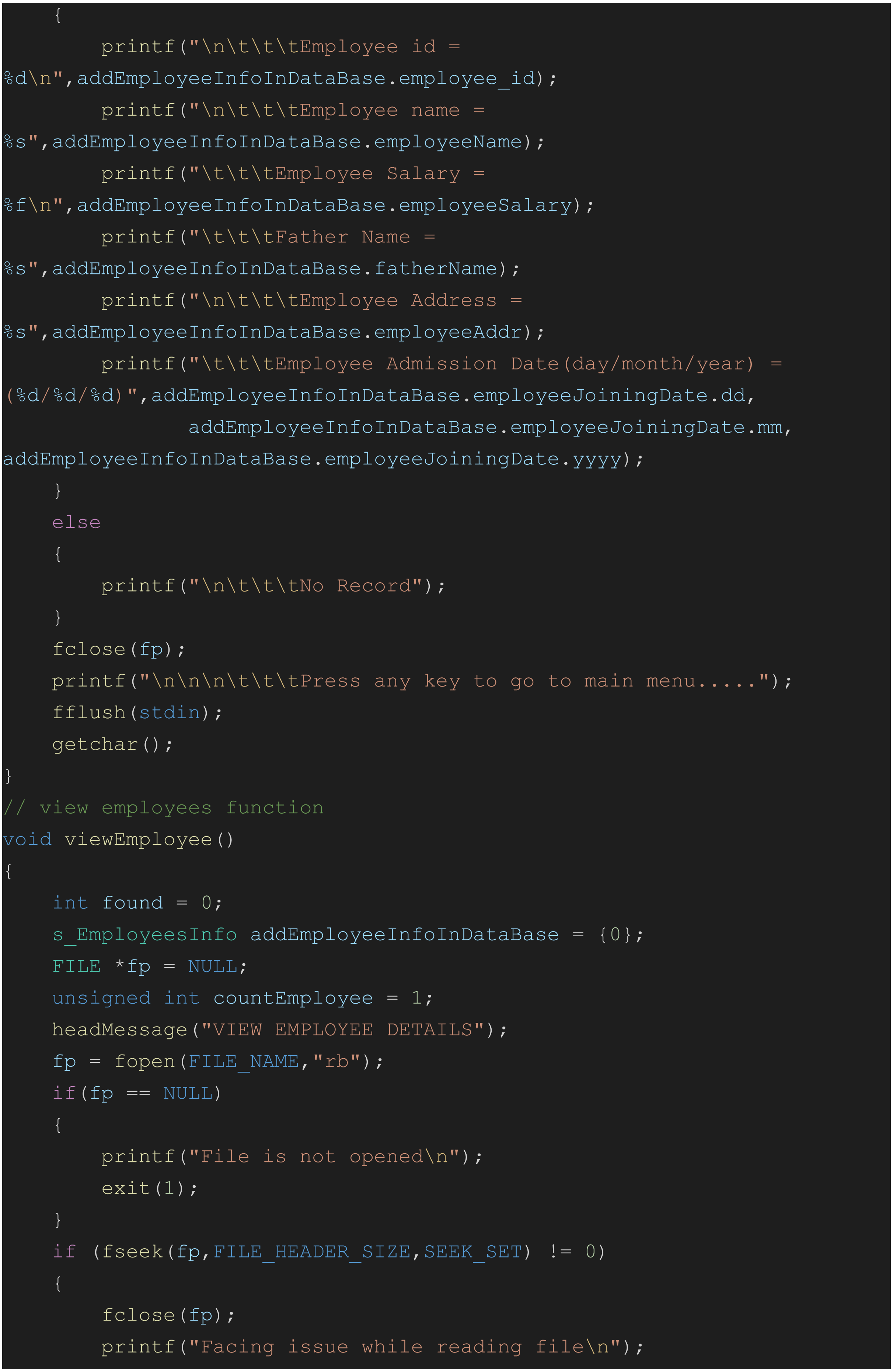
```
//Validate name
int isNameValid(const char *name)
{
    int validName = 1;
    int len = 0;
    int index = 0;
    len = strlen(name);
    for(index =0; index <len ; ++index)
    {
        if(!(isalpha(name[index])) && (name[index] != '\n') &&
(name[index] != ' '))
        {
            validName = 0;
            break;
        }
    }
    return validName;
}
// Function to check leap year.
//Function returns 1 if leap year
int  IsLeapYear(int year)
{
    return (((year % 4 == 0) &&
             (year % 100 != 0)) ||
            (year % 400 == 0));
}
// returns 1 if given date is valid.
int isValidDate(Date *validDate)
{
    //check range of year,month and day
    if (validDate->yyyy > MAX_YR ||
            validDate->yyyy < MIN_YR)
        return 0;
    if (validDate->mm < 1 || validDate->mm > 12)
        return 0;
    if (validDate->dd < 1 || validDate->dd > 31)
        return 0;
    //Handle feb days in leap year
    if (validDate->mm == 2)
    {
        if (IsLeapYear(validDate->yyyy))
            return (validDate->dd <= 29);
        else
```



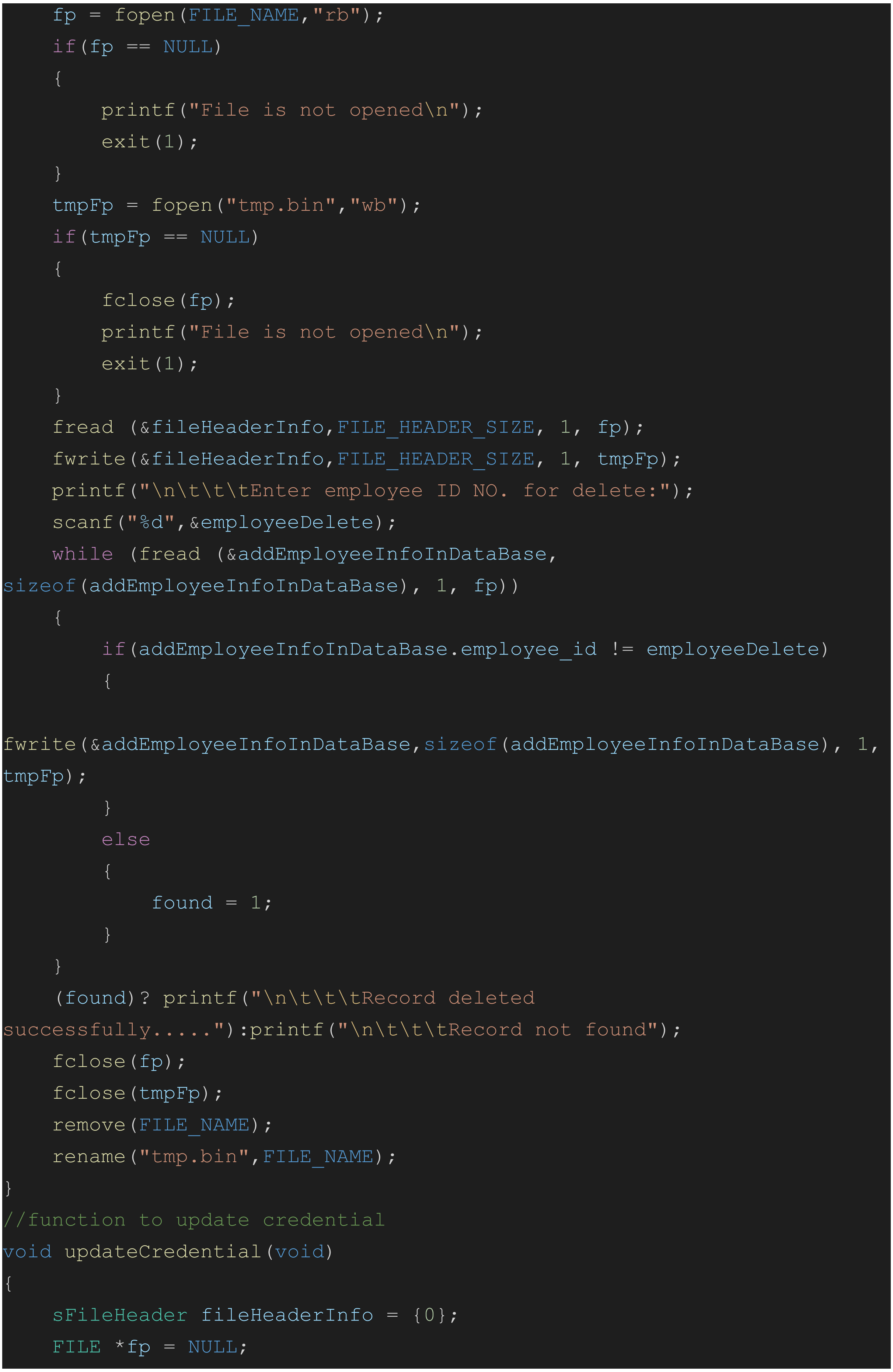
```
            return (validDate->dd <= 28);
    }
    //handle months which has only 30 days
    if (validDate->mm == 4 || validDate->mm == 6 ||
            validDate->mm == 9 || validDate->mm == 11)
        return (validDate->dd <= 30);
    return 1;
}
// Add employee in list
void addEmployeeInDataBase()
{
    s_EmployeesInfo addEmployeeInfoInDataBase = {0};
    FILE *fp = NULL;
    int status = 0;
    fp = fopen(FILE_NAME,"ab+");
    if(fp == NULL)
    {
        printf("File is not opened\n");
        exit(1);
    }
    headMessage("ADD NEW EMPLOYEES");
    printf("\n\n\t\t\tENTER YOUR DETAILS BELOW:");

printf("\n\t\t\t-------------------------------------------------------
--------------------\n");
    printf("\n\t\t\tEmployee ID  = ");
    fflush(stdin);
    scanf("%u",&addEmployeeInfoInDataBase.employee_id);
    do
    {
        printf("\n\t\t\tFather Name  = ");
        fflush(stdin);

fgets(addEmployeeInfoInDataBase.fatherName,MAX_FATHER_NAME,stdin);
        status = isNameValid(addEmployeeInfoInDataBase.fatherName);
        if (!status)
        {
            printf("\n\t\t\tName contain invalid character. Please
enter again.");
        }
    }
    while(!status);
    do
    {
        printf("\n\t\t\tEmployee Name  = ");
        fflush(stdin);

fgets(addEmployeeInfoInDataBase.employeeName,MAX_EMPLOYEE_NAME,stdin);
        status = isNameValid(addEmployeeInfoInDataBase.employeeName);
        if (!status)
        {
            printf("\n\t\t\tName contain invalid character. Please
enter again.");
        }
    }
    while(!status);
    do
    {
        printf("\n\t\t\tEmployee Address  = ");
        fflush(stdin);

fgets(addEmployeeInfoInDataBase.employeeAddr,MAX_FATHER_NAME,stdin);
        status = isNameValid(addEmployeeInfoInDataBase.employeeAddr);
        if (!status)
        {
            printf("\n\t\t\tName contain invalid character. Please
enter again.");
        }
    }
    while(!status);
    printf("\n\t\t\tEmployee Salary  = ");
    fflush(stdin);
    scanf("%f",&addEmployeeInfoInDataBase.employeeSalary);
    do
    {
        //get date year,month and day from user
        printf("\n\t\t\tEnter date in format (day/month/year): ");

scanf("%d/%d/%d",&addEmployeeInfoInDataBase.employeeJoiningDate.dd,&add
EmployeeInfoInDataBase.employeeJoiningDate.mm,&addEmployeeInfoInDataBas
e.employeeJoiningDate.yyyy);
        //check date validity
        status =
isValidDate(&addEmployeeInfoInDataBase.employeeJoiningDate);
        if (!status)
        {
```



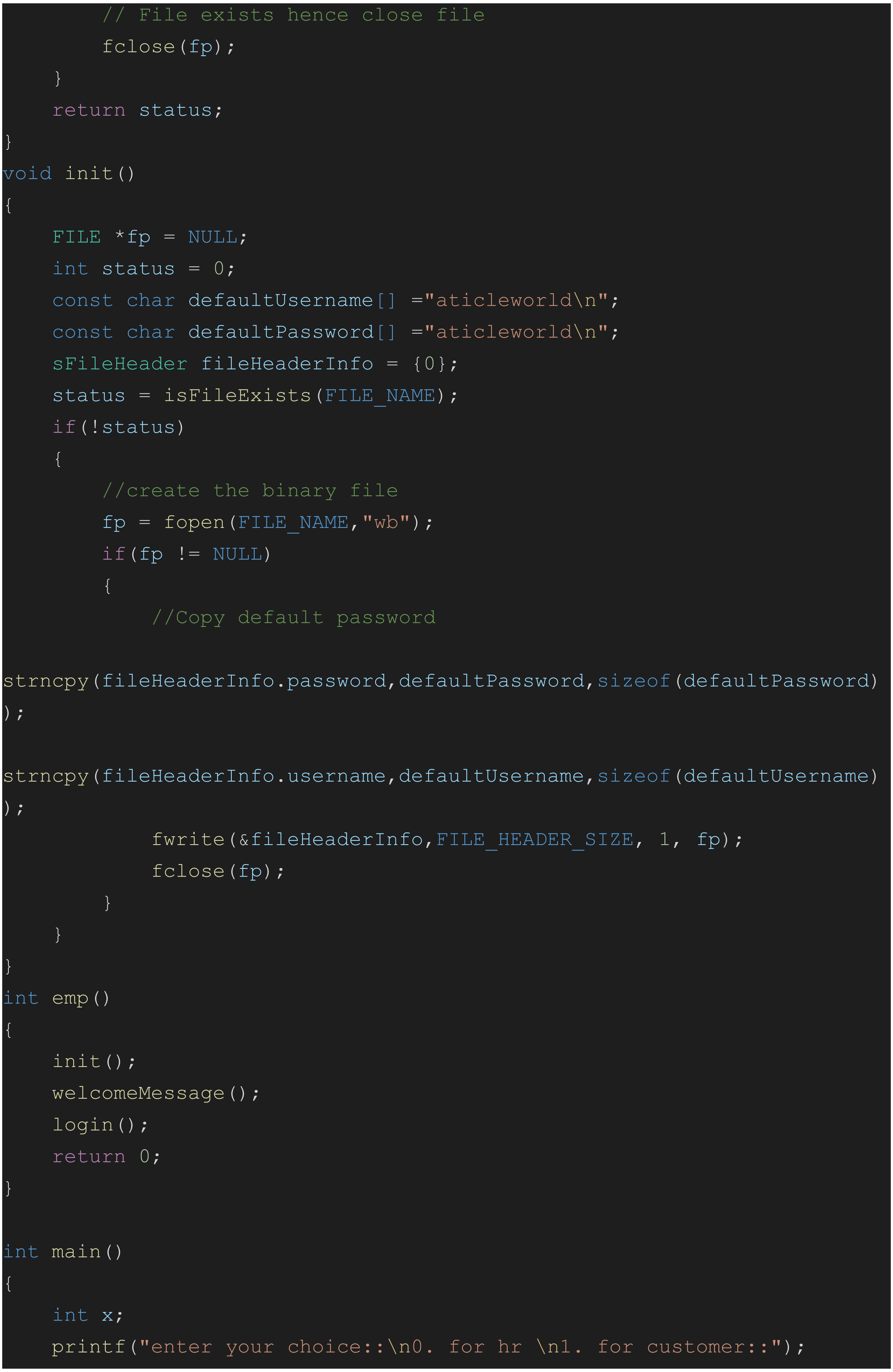
```
            printf("\n\t\t\tPlease enter a valid date.\n");
        }
    }
    while(!status);

fwrite(&addEmployeeInfoInDataBase,sizeof(addEmployeeInfoInDataBase), 1,
fp);
    fclose(fp);
}
// search employee
void searchEmployee()
{
    int found = 0;
    int employeeId =0;
    s_EmployeesInfo addEmployeeInfoInDataBase = {0};
    FILE *fp = NULL;
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)
    {
        printf("\n\t\t\tFile is not opened\n");
        exit(1);
    }
    headMessage("SEARCH EMPLOYEE");
    //put the control on employee detail
    if (fseek(fp,FILE_HEADER_SIZE,SEEK_SET) != 0)
    {
        fclose(fp);
        printf("\n\t\t\tFacing issue while reading file\n");
        exit(1);
    }
    printf("\n\n\t\t\tEnter employee ID NO to search:");
    fflush(stdin);
    scanf("%u",&employeeId);
    while (fread (&addEmployeeInfoInDataBase,
sizeof(addEmployeeInfoInDataBase), 1, fp))
    {
        if(addEmployeeInfoInDataBase.employee_id == employeeId)
        {
            found = 1;
            break;
        }
    }
    if(found)
```



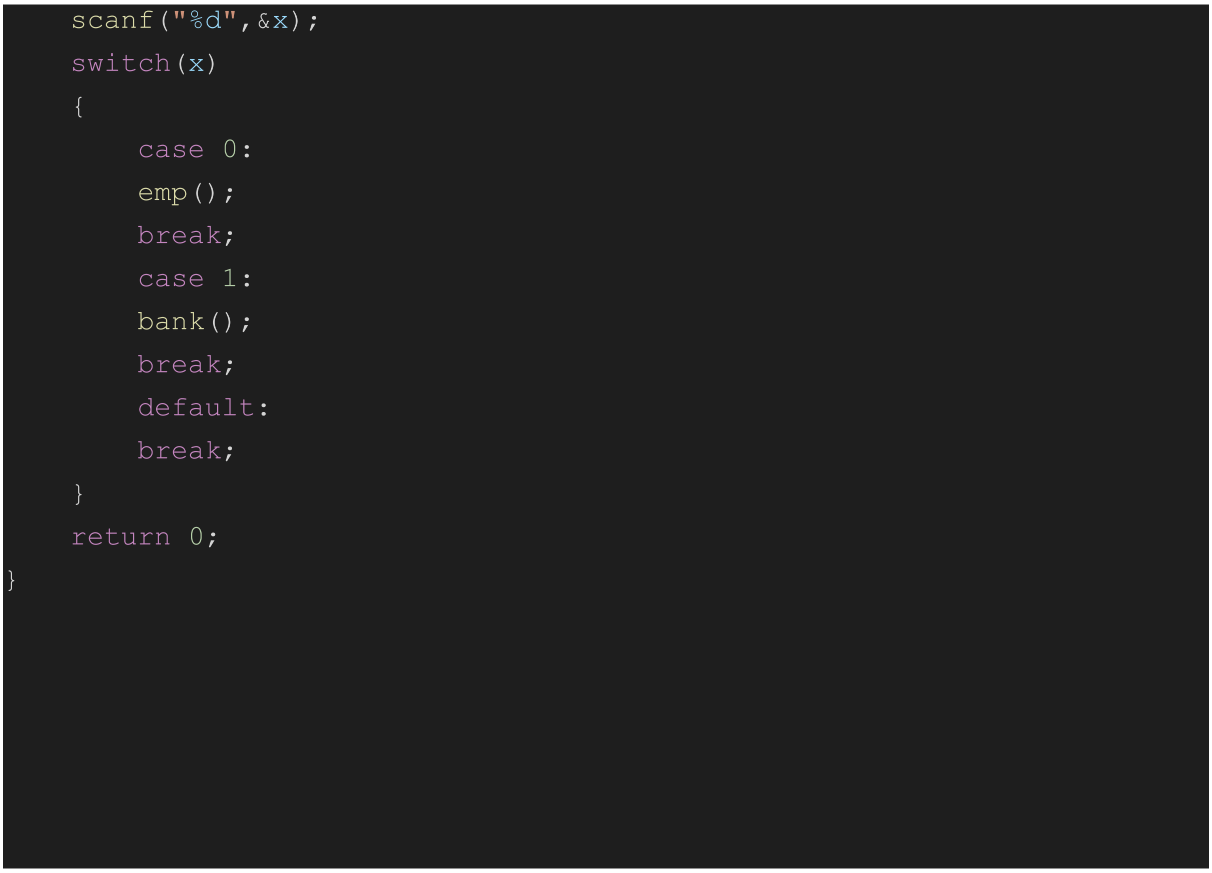
```
    {
        printf("\n\t\t\tEmployee id =
%d\n",addEmployeeInfoInDataBase.employee_id);
        printf("\n\t\t\tEmployee name =
%s",addEmployeeInfoInDataBase.employeeName);
        printf("\t\t\tEmployee Salary =
%f\n",addEmployeeInfoInDataBase.employeeSalary);
        printf("\t\t\tFather Name =
%s",addEmployeeInfoInDataBase.fatherName);
        printf("\n\t\t\tEmployee Address =
%s",addEmployeeInfoInDataBase.employeeAddr);
        printf("\t\t\tEmployee Admission Date(day/month/year) =
(%d/%d/%d)",addEmployeeInfoInDataBase.employeeJoiningDate.dd,
               addEmployeeInfoInDataBase.employeeJoiningDate.mm,
addEmployeeInfoInDataBase.employeeJoiningDate.yyyy);
    }
    else
    {
        printf("\n\t\t\tNo Record");
    }
    fclose(fp);
    printf("\n\n\n\t\t\tPress any key to go to main menu.....");
    fflush(stdin);
    getchar();
}
// view employees function
void viewEmployee()
{
    int found = 0;
    s_EmployeesInfo addEmployeeInfoInDataBase = {0};
    FILE *fp = NULL;
    unsigned int countEmployee = 1;
    headMessage("VIEW EMPLOYEE DETAILS");
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)
    {
        printf("File is not opened\n");
        exit(1);
    }
    if (fseek(fp,FILE_HEADER_SIZE,SEEK_SET) != 0)
    {
        fclose(fp);
        printf("Facing issue while reading file\n");
        exit(1);
    }
    //Print employee count
    printf("\n\t\t\tEmployee Count = %d\n\n",countEmployee);
    while (fread (&addEmployeeInfoInDataBase,
sizeof(addEmployeeInfoInDataBase), 1, fp))
    {
        printf("\t\t\tEmployee id =
%u\n",addEmployeeInfoInDataBase.employee_id);
        printf("\t\t\tEmployee Name =
%s",addEmployeeInfoInDataBase.employeeName);
        printf("\t\t\tEmployee Salary =
%f\n",addEmployeeInfoInDataBase.employeeSalary);
        printf("\t\t\tFather Name =
%s",addEmployeeInfoInDataBase.fatherName);
        printf("\t\t\tEmployee Address =
%s",addEmployeeInfoInDataBase.employeeAddr);
        printf("\t\t\tEmployee Admission Date(day/month/year) =
(%d/%d/%d)\n\n",addEmployeeInfoInDataBase.employeeJoiningDate.dd,
               addEmployeeInfoInDataBase.employeeJoiningDate.mm,
addEmployeeInfoInDataBase.employeeJoiningDate.yyyy);
        found = 1;
        ++countEmployee;
    }
    fclose(fp);
    if(!found)
    {
        printf("\n\t\t\tNo Record");
    }
    printf("\n\n\t\t\tPress any key to go to main menu.....");
    fflush(stdin);
    getchar();
}
// Delete employee entry
void deleteEmployee()
{
    int found = 0;
    int employeeDelete = 0;
    sFileHeader fileHeaderInfo = {0};
    s_EmployeesInfo addEmployeeInfoInDataBase = {0};
    FILE *fp = NULL;
    FILE *tmpFp = NULL;
    headMessage("Delete employee Details");
```



```
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)
    {
        printf("File is not opened\n");
        exit(1);
    }
    tmpFp = fopen("tmp.bin","wb");
    if(tmpFp == NULL)
    {
        fclose(fp);
        printf("File is not opened\n");
        exit(1);
    }
    fread (&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
    fwrite(&fileHeaderInfo,FILE_HEADER_SIZE, 1, tmpFp);
    printf("\n\t\t\tEnter employee ID NO. for delete:");
    scanf("%d",&employeeDelete);
    while (fread (&addEmployeeInfoInDataBase,
sizeof(addEmployeeInfoInDataBase), 1, fp))
    {
        if(addEmployeeInfoInDataBase.employee_id != employeeDelete)
        {

fwrite(&addEmployeeInfoInDataBase,sizeof(addEmployeeInfoInDataBase), 1,
tmpFp);
        }
        else
        {
            found = 1;
        }
    }
    (found)? printf("\n\t\t\tRecord deleted
successfully.....):printf("\n\t\t\tRecord not found");
    fclose(fp);
    fclose(tmpFp);
    remove(FILE_NAME);
    rename("tmp.bin",FILE_NAME);
}
//function to update credential
void updateCredential(void)
{
    sFileHeader fileHeaderInfo = {0};
    FILE *fp = NULL;
    unsigned char userName[MAX_SIZE_USER_NAME] = {0};
    unsigned char password[MAX_SIZE_PASSWORD] = {0};
    headMessage("Update Credential");
    fp = fopen(FILE_NAME,"rb+");
    if(fp == NULL)
    {
        printf("File is not opened\n");
        exit(1);
    }
    fread (&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
    if (fseek(fp,0,SEEK_SET) != 0)
    {
        fclose(fp);
        printf("\n\t\t\tFacing issue while updating password\n");
        exit(1);
    }
    printf("\n\n\t\t\tNew Username:");
    fflush(stdin);
    fgets(userName,MAX_SIZE_USER_NAME,stdin);
    printf("\n\n\t\t\tNew Password:");
    fflush(stdin);
    fgets(password,MAX_SIZE_PASSWORD,stdin);
    strncpy(fileHeaderInfo.username,userName,sizeof(userName));
    strncpy(fileHeaderInfo.password,password,sizeof(password));
    fwrite(&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
    fclose(fp);
    printf("\n\t\t\tYour Password has been changed successfully");
    printf("\n\t\t\ttLogin Again:");
    fflush(stdin);
    getchar();
    exit(1);
}
//Display menu
void menu1()
{
    int choice = 0;
    do
    {
        headMessage("MAIN MENU");
        printf("\n\n\n\t\t\t1.Add employee");
        printf("\n\t\t\t2.Search employee");
        printf("\n\t\t\t3.View employee");
        printf("\n\t\t\t4.Delete employee");
        printf("\n\t\t\t5.Update Password");
        printf("\n\t\t\t0.Exit");
        printf("\n\n\n\t\t\tEnter choice => ");
        scanf("%d",&choice);
        switch(choice)
        {
        case 1:
            addEmployeeInDataBase();
            break;
        case 2:
            searchEmployee();
            break;
        case 3:
            viewEmployee();
            break;
        case 4:
            deleteEmployee();
            break;
        case 5:
            updateCredential();
            break;
        case 0:
            printf("\n\n\n\t\t\t\tThank you!!!\n\n\n\n\n");
            exit(1);
            break;
        default:
            printf("\n\n\n\t\t\tINVALID INPUT!!! Try again...");
        }                                             //Switch Ended
    }
    while(choice!=0);                                        //Loop
Ended
}
//login password
void login()
{
    unsigned char userName[MAX_SIZE_USER_NAME] = {0};
    unsigned char password[MAX_SIZE_PASSWORD] = {0};
    int L=0;
    sFileHeader fileHeaderInfo = {0};
    FILE *fp = NULL;
    headMessage("Login");
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)
    {
        printf("File is not opened\n");
        exit(1);
    }
    fread (&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
    fclose(fp);
    do
    {
        printf("\n\n\n\t\t\t\tUsername:");
        fgets(userName,MAX_SIZE_USER_NAME,stdin);
        printf("\n\t\t\t\tPassword:");
        fgets(password,MAX_SIZE_PASSWORD,stdin);
        if((!strcmp(userName,fileHeaderInfo.username)) &&
(!strcmp(password,fileHeaderInfo.password)))
        {
            menu1();
        }
        else
        {
            printf("\t\t\t\tLogin Failed Enter Again Username &
Password\n\n");
            L++;
        }
    }
    while(L<=3);
    if(L>3)
    {
        headMessage("Login Failed");
        printf("\t\t\t\tSorry,Unknown User.");
        getch();
        system("cls");
    }
}
//Check file exist or not
int isFileExists(const char *path)
{
    // Try to open file
    FILE *fp = fopen(path, "rb");
    int status = 0;
    // If file does not exists
    if (fp != NULL)
    {
        status = 1;
```



```
        // File exists hence close file
        fclose(fp);
    }
    return status;
}
void init()
{
    FILE *fp = NULL;
    int status = 0;
    const char defaultUsername[] ="aticleworld\n";
    const char defaultPassword[] ="aticleworld\n";
    sFileHeader fileHeaderInfo = {0};
    status = isFileExists(FILE_NAME);
    if(!status)
    {
        //create the binary file
        fp = fopen(FILE_NAME,"wb");
        if(fp != NULL)
        {
            //Copy default password

strncpy(fileHeaderInfo.password,defaultPassword,sizeof(defaultPassword)
);

strncpy(fileHeaderInfo.username,defaultUsername,sizeof(defaultUsername)
);
            fwrite(&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
            fclose(fp);
        }
    }
}
int emp()
{
    init();
    welcomeMessage();
    login();
    return 0;
}

int main()
{
    int x;
    printf("enter your choice::\n0. for hr \n1. for customer::");
```



```
    scanf("%d",&x);
    switch(x)
    {
        case 0:
        emp();
        break;
        case 1:
        bank();
        break;
        default:
        break;
    }
    return 0;
}
```