Advanced C Lab assignment (Teams)

Name: Sankalp Mukim

Registration Number: 20BDS0128

Course Code: CSE2010

Slot: L27+L28

Table of Contents

[Question 1 1](#_Toc96786096)

[Code 1](#_Toc96786097)

[Output 4](#_Toc96786098)

[Question 2 5](#_Toc96786099)

[Code 5](#_Toc96786100)

[Output 6](#_Toc96786101)

# Question 1

A screenshot of a computer

Description automatically generated

## Code

#include **<**stdio.h**>**

#include **<**stdlib.h**>**

#include **<**string.h**>**

**//** create a structure to specify data of customers in a bank

**//** the data to be stored is:

**//** account number, name, balance

***struct*** customer

{

***int*** account\_number;

***char*** name[20];

***float*** balance;

};

**//** create a function to print the account number and name of each customer with balance below 100

**//** the function should take a pointer to the structure as an argument

***void*** print\_customer(***struct*** customer \*c)

{

    printf(**"**Account number: %d\n**"**, c->account\_number);

    printf(**"**Name: %s\n**"**, c->name);

    printf(**"**Balance: %.2f\n**"**, c->balance);

}

***void*** print\_customer\_below\_100(***struct*** customer \*\*c, ***int*** n)

{

***int*** i;

    for (i = 0; i < n; i++)

    {

        if (c[i]->balance < 100)

        {

            print\_customer(c[i]);

        }

    }

}

**//** withdrawal function

**//** the function should take a pointer to the structure as an argument

***int*** withdraw(***struct*** customer \*c, ***int*** amount)

{

    if (c->balance - 500 < amount)

    {

        return 0;

    }

    else

    {

        c->balance -= amount;

        return 1;

    }

}

**//** deposit function

**//** the function should take a pointer to the structure as an argument

***void*** deposit(***struct*** customer \*c, ***int*** amount)

{

    c->balance += amount;

}

**//** create customer

***struct*** customer \*create\_customer(***int*** account\_number, ***char*** \*name, ***float*** balance)

{

***struct*** customer \*c = (***struct*** customer \*)malloc(sizeof(***struct*** customer));

    c->account\_number = account\_number;

    strcpy(c->name, name);

    c->balance = balance;

    return c;

}

**//** query customer by account number

***struct*** customer \*query\_customer(***struct*** customer \*\*c, ***int*** n, ***int*** account\_number)

{

***int*** i;

    for (i = 0; i < n; i++)

    {

        if (c[i]->account\_number == account\_number)

        {

            return c[i];

        }

    }

    return NULL;

}

***int*** main()

{

**//** create an array of customers

***struct*** customer \*customers[5];

**//** create customers

    customers[0] = create\_customer(1, **"**John**"**, 50);

    customers[1] = create\_customer(2, **"**Mary**"**, 75);

    customers[2] = create\_customer(3, **"**Peter**"**, 300);

    customers[3] = create\_customer(4, **"**Paul**"**, 400);

    customers[4] = create\_customer(5, **"**Mary**"**, 500);

**//** print customers

***int*** i;

    for (i = 0; i < 5; i++)

    {

        print\_customer(customers[i]);

    }

**//** print customers with balance below 100

    print\_customer\_below\_100(customers, 5);

**//** query customer by account number

***struct*** customer \*c = query\_customer(customers, 5, 2);

    if (c != NULL)

    {

        print\_customer(c);

    }

    else

    {

        printf(**"**Customer not found\n**"**);

    }

**//** withdraw

    if (withdraw(c, 500))

    {

        printf(**"**Withdrawal successful\n**"**);

    }

    else

    {

        printf(**"**Withdrawal failed\n**"**);

    }

**//** print customer

    print\_customer(c);

**//** deposit

    deposit(c, 500);

**//** print customer

    print\_customer(c);

    return 0;

}

## Output

A screenshot of a computer

Description automatically generated with medium confidence

# Question 2

Text

Description automatically generated

## Code

**//** write a function that compares two given dates

#include **<**stdio.h**>**

**//** to store a date, use a structure that contains the day, month and year

***struct*** date

{

***int*** day;

***int*** month;

***int*** year;

};

**//** if the dates are equal, return 0 else return 1

***int*** compare\_dates(***struct*** date \*d1, ***struct*** date \*d2)

{

    if (d1->day == d2->day && d1->month == d2->month && d1->year == d2->year)

    {

        return 0;

    }

    else

    {

        return 1;

    }

}

***int*** main()

{

**//** equal date case

***struct*** date d1 = {1, 1, 2000};

***struct*** date d2 = {1, 1, 2000};

    printf(**"**%d\n**"**, compare\_dates(&d1, &d2));

**//** not equal date case

***struct*** date d3 = {1, 1, 2000};

***struct*** date d4 = {1, 2, 2000};

    printf(**"**%d\n**"**, compare\_dates(&d3, &d4));

    return 0;

}

## Output

A screenshot of a computer

Description automatically generated