Advanced C Lab assignment (Teams)

Name: Sankalp Mukim

Registration Number: 20BDS0128

Course Code: CSE2010

Slot: L27+L28

Table of Contents

Question 1

a. Create a structure to specify data of customers in a bank. The data to be stored is: Account number, Name, Balance in account. Assume maximum of 100 customers in the bank.

i) Write a function to print the Account number and name of each customer with balance below Rs. 100.

ii) If a customer requests for withdrawal or deposit, the form contains the fields: Acct. no, amount, code (1 for deposit, 0 for withdrawal)
Write a program to give a message, "The balance is insufficient for the specified withdrawal", if on withdrawal the balance falls below Rs. 500.

Code

```c
#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
```

```c
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

```
sanka@Sankalps-HP ~ OneDrive Documents .SEM4 CSE2010 - Advanced C Programming Lab impromptuassignment main
> gcc -o q1 q1.c
sanka@Sankalps-HP ~ OneDrive Documents .SEM4 CSE2010 - Advanced C Programming Lab impromptuassignment main
> ./q1
Account number: 1
Name: John
Balance: 50.00
Account number: 2
Name: Mary
Balance: 75.00
Account number: 3
Name: Peter
Balance: 300.00
Account number: 4
Name: Paul
Balance: 400.00
Account number: 5
Name: Mary
Balance: 500.00
Account number: 1
Name: John
Balance: 50.00
Account number: 2
Name: Mary
Balance: 75.00
Account number: 2
Name: Mary
Balance: 75.00
Withdrawal failed
Account number: 2
Name: Mary
Balance: 75.00
Account number: 2
Name: Mary
Balance: 575.00
sanka@Sankalps-HP ~ OneDrive Documents .SEM4 CSE2010 - Advanced C Programming Lab impromptuassignment main
>
```

Question 2

b. Write a function that compares two given dates. To store a date use a structure that contains three members namely day, month and year. If the dates are equal the function should return 0, otherwise it should return 1.

Code

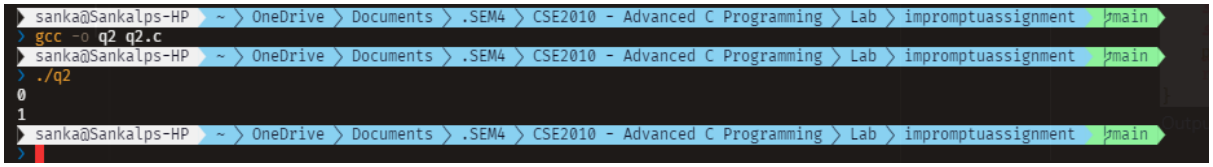
```c
// write a function that compares two given dates
#include <stdio.h>
// to store a date, use a structure that contains the day, month and year

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


sanka@Sankalps-HP ~ OneDrive Documents .SEM4 CSE2010 - Advanced C Programming Lab impromptuassignment main
gcc -o q2 q2.c
sanka@Sankalps-HP ~ OneDrive Documents .SEM4 CSE2010 - Advanced C Programming Lab impromptuassignment main
./q2
0
1
sanka@Sankalps-HP ~ OneDrive Documents .SEM4 CSE2010 - Advanced C Programming Lab impromptuassignment main