Lab Assignment 7

Akshat Mittal - 20107

July 2021

Contents

1

1.1 Code

```c
#include <stdio.h>
#include <stdlib.h>
struct account
{
    int account_number;
    char name[20];
    float balance;
};
struct account customer[5] = {{8300, "Jonas", 450},
                              {2600, "Martha", 40},
                              {8700, "Klaus", 200},
                              {4500, "Stefen", 300},
                              {8600, "Caroline", 400}}; //values for 5 customers
void print_account(struct account *cust)                //input details of customer
{
    printf("\n Account number: %d", cust->account_number);
    printf("\n Name: %s", cust->name);
    printf("\n Balance: %f", cust->balance);
}
void print_below_100(struct account *cust)
{
    if (cust->balance < 100)
    {
        printf("\n Account number: %d", cust->account_number);
        printf("\n Name: %s", cust->name);
    }
}
int main()
{
    printf("\n\n\n");
    int i, j, choice, acc, am;
    printf("\n Below listed accounts have balance low than 100");
    for (i = 0; i < 5; ++i)
        print_below_100(&customer[i]); //if balance<100 call print_below();
    printf("\n\n\n Enter account number: ");
    scanf("%d", &acc);
    printf("\n Enter balance: ");
    scanf("%d", &am);
    printf("\n How can we serve you? ");
    printf("\n  0. Withdrawal");
    printf("\n  1. Deposit\n ");
    scanf("%d", &choice);
    printf("\n The balance is insufficinet for the specified withdrawal/deposit");
    printf("\n\n\n");
    return 0;
}
```

1.2 Output

```
Below listed accounts have balance low than 100
Account number: 2600
Name: Martha


Enter account number: 8700

Enter balance: 150

How can we serve you?
 0. Withdrawal
 1. Deposit
0

The balance is insufficinet for the specified withdrawal/deposit
```

2

2.1 Code

```c
#include <stdio.h>
struct full_date
{
    int date, month, year;
} Date[2];
void enter_date(struct full_date *DATE) //to input date
{
    printf("\n Enter date: ");
    scanf("%d", &(DATE->date));
    printf(" Enter month: ");
    scanf("%d", &(DATE->month));
    printf(" Enter year: ");
    scanf("%d", &(DATE->year));
}
void print_date(struct full_date *DATE) //to output date
{
    printf("%d / %d / %d", (DATE->date), (DATE->month), (DATE->year));
}
int main()
{
    printf("\n\n\n");
    printf("\n Enter 1st date: \n");
    enter_date(&Date[0]);
    printf("\n Enter 2nd date: \n");
    enter_date(&Date[1]);
    //input for both date have been taken
    printf("\n\n");
    print_date(&Date[0]);
    printf(" and ");
    print_date(&Date[1]);
    //comparing date, month and year of both date
    if ((Date[0].date == Date[1].date) &&
        (Date[0].month == Date[1].month) &&
        (Date[0].year == Date[1].year))
        printf(" are equal.");
    else
        printf(" are not equal.");
    printf("\n\n\n");
    return 0;
}
```

2.2 Output

```
 Enter 1st date:

 Enter date: 25
 Enter month: 6
 Enter year: 2002

 Enter 2nd date:

 Enter date: 19
 Enter month: 7
 Enter year: 2021


25 / 6 / 2002 and 19 / 7 / 2021 are not equal.
```

3

3.1 Code

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct student
{
    int roll;
    char name[20], course[10], department[10];
    int year;
}; //declaring values for 5 students
struct student Students[5] = {{1, "Jonas", "M.Tech.", "IT", 2017},
                              {2, "Martha", "BBA", "Arts", 2017},
                              {3, "Klaus", "Ph.D.", "Occult", 2001},
                              {4, "Stefen", "B.Tech.", "CSE", 2020},
                              {5, "Caroline", "B.Pharma.", "Neurology", 2020}};
void print_student(struct student *stud) //to print details of a student
{
    printf("\n\n Roll Number: %d", stud->roll);
    printf("\n Name: %s", stud->name);
    printf("\n Course: %s", stud->course);
    printf("\n Department: %s", stud->department);
    printf("\n Course: %s", stud->course);
    printf("\n Year of joining: %d", stud->year);
}
void print_join_year(int y) //print details of student with a particular year
{
    for (int i = 0; i < 5; ++i)
        if (Students[i].year == y)
            print_student(&Students[i]);
}
void print_student_roll(int r) //print details of student with a roll no.
{
    for (int i = 0; i < 5; ++i)
        if (Students[i].roll == r)
            print_student(&Students[i]);
}
int main()
{
    int y, r;
    printf("\n Enter a year: ");
    scanf("%d", &y);
    print_join_year(y);
    printf("\n\n Enter a roll no.: ");
    scanf("%d", &r);
    print_student_roll(r);
    return 0;
}
```

3.2 Output

```
Enter a year: 2020


Roll Number: 4
Name: Stefen
Course: B.Tech.
Department: CSE
Course: B.Tech.
Year of joining: 2020

Roll Number: 5
Name: Caroline
Course: B.Pharma.
Department: Neurology
Course: B.Pharma.
Year of joining: 2020

Enter a roll no.: 3


Roll Number: 3
Name: Klaus
Course: Ph.D.
Department: Occult
Course: Ph.D.
Year of joining: 2001
```

4

4.1 Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct library
{
    int accession_number;
    char title[30];
    char author[20];
    float price;
    int flag;
};
// bubble sort to sort in ascending order
void bubble_sort(int a[], int n)
{
    int i, j, t;
    for (i = 0; i < n; ++i)
        for (j = 0; j < n - 1; ++j)
            if (a[j] > a[j + 1])
            {
                t = a[j];
                a[j] = a[j + 1];
                a[j + 1] = t;
            }
}
// to print details of a book
void print_book(struct library *book)
{
    printf("\n");
    printf("\n Accesion Number: %d", book->accession_number);
    printf("\n Book Title: %s", book->title);
    printf("\n Author Name: %s", book->author);
    printf("\n Price: %f", book->price);
    printf("\n Issued: %s", book->flag ? "Yes" : "No");
}
// declaring values for 5 students
struct library books[5] = {{2005, "Harry Potter", "Rowling", 800, 1},
                           {2009, "The Fault in Our Stars", "Green", 300, 1},
                           {2003, "A Brief History Of Time", "Hawkings", 1000, 0},
                           {2015, "The Theory of Everything", "Hawkings", 1500, 1},
                           {2001, "Turtles all the way down", "Green", 200, 0}};

int main()
{
    printf("\n\n\n");
    int ch, count = 5, acc, accn[5];
```

```
char aut[20];
printf("\n \t\t LIBRARY \n");
printf("\n 1. Add Book Information");
printf("\n 2. Display book information");
printf("\n 3. List all books of given author");
printf("\n 4. List the title of specified book");
printf("\n 5. List the count of books in the library");
printf("\n 6. List the books in the order of accession number");
printf("\n 7. Exit");
printf("\n Enter your choice: ");
scanf("%d", &ch);
switch (ch)
{
case 1: // Add Book Information
    printf("\nEnter book information:- \n");
    printf(" Accesion Number: ");
    scanf("%d", &books[count].accession_number);
    printf(" Book Title: ");
    scanf("%s", &books[count].title);
    printf(" Author Name: ");
    scanf("%s", &books[count].author);
    printf(" Price: ");
    scanf("%f", &books[count].price);
    books[count].flag = 0;
    count++;
    break;
case 2: // Display book information
    printf("\n Accesion Number: ");
    scanf("%d", &acc);
    for (int i = 0; i < count; ++i)
        if (books[i].accession_number == acc)
            print_book(&books[i]);
    break;
case 3: // List all books of given author
    printf("\n Author Name: ");
    scanf("%s", &aut);
    for (int i = 0; i < count; ++i)
        if (!strcmp(books[i].author, aut))
            print_book(&books[i]);
    break;
case 4: // List the title of specified book
    printf("\n Accesion Number: ");
    scanf("%d", &acc);
    for (int i = 0; i < count; ++i)
        if (books[i].accession_number == acc)
            printf("\n Book Title: %s", books[i].title);
    break;
case 5: // List the count of books in the library
    printf("\n The no. of books; %d", count);
    break;
```

```
    case 6: // List the books in the order of accession number
        for (int i = 0; i < count; ++i)
            accn[i] = books[i].accession_number;
        bubble_sort(accn, count); // to arrange accession no. in ascending order
        for (int i = 0; i < count; ++i)
            for (int j = 0; j < count; ++j)
                if (books[j].accession_number == accn[i]) // printing in ascending order
                    print_book(&books[j]);
        break;
    case 7: // Exit
        exit(0);
        break;
    default:
        break;
    }
    printf("\n\n\n");
    return 0;
}
```

4.2 Output

```
                LIBRARY

1. Add Book Information
2. Display book information
3. List all books of given author
4. List the title of specified book
5. List the count of books in the library
6. List the books in the order of accession number
7. Exit
Enter your choice: 6


Accesion Number: 2001
Book Title: Turtles all the way down
Author Name: Green
Price: 200.000000
Issued: No

Accesion Number: 2003
Book Title: A Brief History Of Time
Author Name: Hawkings
Price: 1000.000000
Issued: No

Accesion Number: 2005
Book Title: Harry Potter
Author Name: Rowling
Price: 800.000000
Issued: Yes

Accesion Number: 2009
Book Title: The Fault in Our Stars
Author Name: Green
Price: 300.000000
Issued: Yes

Accesion Number: 2015
Book Title: The Theory of Everything
Author Name: Hawkings
Price: 1500.000000
Issued: Yes
```

5

5.1 Code

```c
#include <stdio.h>
struct length_fi
{
    int feet;
    float inch;
};
struct length_mcm
{
    int m;
    float cm;
};
struct length_mcm change(struct length_fi *len_obj)
{
    struct length_mcm test_obj;
    float centi;
    int meter;
    // centimeters = ( total inches ) * 2.54
    centi = ((len_obj->feet * 12) + len_obj->inch) * 2.54;
    meter = (centi / 100);
    centi = centi - (meter * 100);
    test_obj.cm = centi;
    test_obj.m = meter;
    return test_obj;
}
int main()
{
    printf("\n\n\n");
    struct length_fi old_len;
    struct length_mcm new_len;
    printf("\nEnter the length in feet inch:- \n");
    printf(" Feet: ");
    scanf("%d", &old_len.feet);
    printf(" Inches: ");
    scanf("%f", &old_len.inch);
    new_len = change(&old_len); // conversion from feet-inch to m-cm
    printf("\nNew Length:- ");
    printf("\n Meters: %d", new_len.m);
    printf("\n Centimeters: %f", new_len.cm);
    printf("\n\n\n");
    return 0;
}
```

5.2 Output

```
Enter the length in feet inch:-
 Feet: 5
 Inches: 10

New Length:-
 Meters: 1
 Centimeters: 77.800003
```

6

6.1 Code

```
#include <stdio.h>
struct party_items
{
    char name[20];
    float price;
    int quantity;
} item[50];
void read_item(struct party_items *item)    // to input details of new item
{
    printf("\nEnter details:- \n");
    printf(" Name: ");
    scanf("%s", &(item->name));
    printf(" Price: ");
    scanf("%f", &(item->price));
    printf(" Enter quantity: ");
    scanf("%d", &(item->quantity));
}
int main()
{
    printf("\n\n\n");
    int count;
    printf("Enter the no. of items (<=50): ");
    scanf("%d", &count);
    for (int i = 0; i < count; ++i)
        read_item(&item[i]);
    float total = 0;
    for (int i = 0; i < count; ++i)
        total += item[i].price * item[i].quantity;  // to calculate total amount
    printf("\n\n The total amount to be paid is: %f", total);
    printf("\n\n\n");
    return 0;
}
```

6.2 Output

```
Enter the no. of items (<=50): 3

Enter details:-
 Name: Momos
 Price: 5
 Enter quantity: 20

Enter details:-
 Name: Pizza
 Price: 150
 Enter quantity: 2

Enter details:-
 Name: Decors
 Price: 20
 Enter quantity: 3


 The total amount to be paid is: 460.000000
```

7

7.1 Code

```c
#include <stdio.h>
struct Date
{
    int date, month, year;
}current_date;
struct employee
{
    int code;
    char name[20];
    struct Date doj;
};
void print_employee(struct employee *employes)
{
    printf("\n Employee Code: %d", employes->code);
    printf("\n Employee Name: %s", employes->name);
    printf("\n Employee Date of joining: %d / %d / %d",
            employes->doj.date, employes->doj.month, employes->doj.year);
}//declaring values for 5 employes
struct employee employes[5] = {{8300, "Jonas", {18, 11, 2018}},
                               {2600, "Martha", {25, 3, 2018}},
                               {6900, "Klaus", {1, 1, 2001}},
                               {4500, "Stefen", {25, 6, 2015}},
                               {8600, "Caroline", {28, 9, 2015}}};
int main()
{
    printf("\n Enter current date:- \n");
    printf(" Date: ");
    scanf("%d", &current_date.date);
    printf(" Month: ");
    scanf("%d", &current_date.month);
    printf(" Year: ");
    scanf("%d", &current_date.year);
    for (int i = 0; i < 5; ++i)
    {
        if ((current_date.year - employes[i].doj.year) > 3) print_employee(&employes[i]);
        else if ((current_date.year - employes[i].doj.year) == 3)   //year diff = 3
        {
            if ((current_date.month > employes[i].doj.month))   //month diff > 0
                print_employee(&employes[i]);
            else if ((current_date.month == employes[i].doj.month)) //no month diff
                if ((current_date.date > employes[i].doj.date))   //date diff > 0
                    print_employee(&employes[i]);
        }
    }
    return 0;
}
```

7.2 Output

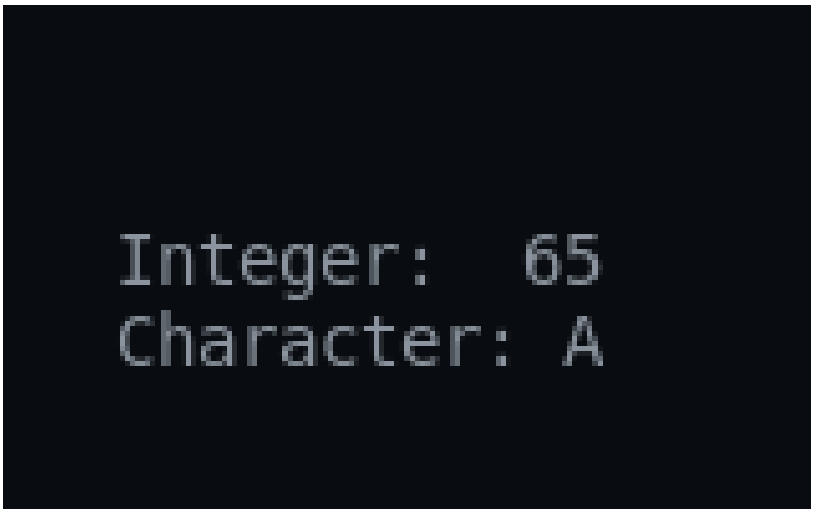
```
Enter current date:-
Date: 19
Month: 7
Year: 2021

Employee Code: 2600
Employee Name: Martha
Employee Date of joining: 25 / 3 / 2018
Employee Code: 6900
Employee Name: Klaus
Employee Date of joining: 1 / 1 / 2001
Employee Code: 4500
Employee Name: Stefen
Employee Date of joining: 25 / 6 / 2015
Employee Code: 8600
Employee Name: Caroline
Employee Date of joining: 28 / 9 / 2015
```

8

8.1 Code

```c
#include <stdio.h>
union share
{
    int i;
    char ch;
};  // union of an interger and charcater
union share cnvt;
int main()
{
    printf("\n\n\n");
    cnvt.i = 65;
    printf("\n Integer:  %d", cnvt.i);  // 65
    printf("\n Character: %c", cnvt.ch);    // char(65) = A
    printf("\n\n\n");
    return 0;
}
```

8.2 Output



9

9.1 Code

```c
#include <stdio.h>
union party_items
{
    char name[20];
    int price;
    int quantity;
} item[50];
// to input details of new item
void read_item(union party_items *item)
{
    printf("\nEnter details:- \n");
    printf(" Name: ");
    scanf("%s", &(item->name));
    printf(" Price: ");
    scanf("%d", &(item->price));
    printf(" Enter quantity: ");
    scanf("%d", &(item->quantity));
    printf("\n Price: %d \t Quantity: %d\n\n", (item->price), (item->quantity));
}
int main()
{
    printf("\n\n\n");
    int count;
    printf("Enter the no. of items (<=50): ");
    scanf("%d", &count);
    for (int i = 0; i < count; ++i)
        read_item(&item[i]);
    float total = 0;
    for (int i = 0; i < count; ++i)
        total += item[i].price * item[i].quantity; // to calculate total amount
    printf("\n\n The total amount to be paid is: %f", total);
    printf("\n\n\n");
    return 0;
}
```

9.2 Output

```
Enter the no. of items (<=50): 2

Enter details:-
 Name: momos
 Price: 5
 Enter quantity: 20

 Price: 20       Quantity: 20


Enter details:-
 Name: pizza
 Price: 150
 Enter quantity: 2

 Price: 2        Quantity: 2



 The total amount to be paid is: 404.000000
```

9.3 Code

```c
#include <stdio.h>
union length_fi
{
    int feet;
    int inch;
};
union length_mcm
{
    int m;
    int cm;
};
union length_mcm change(union length_fi *len_obj)
{
    union length_mcm test_obj;
    int centi, meter; // centimeters = ( total inches ) * 2.54
    centi = ((len_obj->feet * 12) + len_obj->inch) * 2.54;
    meter = (centi / 100);
    centi = centi - (meter * 100);
    printf("\n Before commiting to union:- \n  m: %d \t cm: %d", meter, centi);
    test_obj.m = meter, test_obj.cm = centi;
    printf("\n After commiting to union:- \n  m: %d \t cm: %d", test_obj.m, test_obj.cm);
    return test_obj;
}
```

```c
int main()
{
    printf("\n\n\n");
    union length_fi old_len;
    union length_mcm new_len;
    printf("\nEnter the length in feet inch:- \n");
    printf(" Feet: ");
    scanf("%d", &old_len.feet);
    printf(" Inches: ");
    scanf("%d", &old_len.inch);
    printf("\n Feet: %d \t Inch: %d\n", old_len.feet, old_len.inch);
    new_len = change(&old_len); // conversion from feet-inch to m-cm
    printf("\n\n New Length:- ");
    printf("\n  Meters: %d", new_len.m);
    printf("\n  Centimeters: %d", new_len.cm);
    printf("\n\n\n");
    return 0;
}
```

9.4 Output

```
Enter the length in feet inch:-
 Feet: 4
 Inches: 2

 Feet: 2          Inch: 2

 Before commiting to union:-
  m: 0    cm: 66
 After commiting to union:-
  m: 66           cm: 66

 New Length:-
  Meters: 66
  Centimeters: 66
```

10

10.1 Code

```c
#include <stdio.h>
union floatingPoint
{
    float f;
    double d;
    long double x;
} input;
int main()
{
    printf("\n\n\n");
    printf("\n Enter float value: ");
    scanf("%f", &input.f);
    printf("\n Enter double value: ");
    scanf("%lf", &input.d);
    printf("\n Enter long double value: ");
    scanf("%Lf", &input.x);
    //After this, all variables store same values, different format
    printf("\n Float value (f): %f", input.f);
    printf("\n Double value (d): %lf", input.d);
    printf("\n Long Double value (x): %Lf", input.x);
    printf("\n\n\n");
    return 0;
}
```

10.2 Output

```
Enter float value: 5.4

Enter double value: 5.4

Enter long double value: 5.4

Float value (f): -107374184.000000
Double value (d): -0.000000
Long Double value (x): 5.400000
```