

Lab Assignment -9

1. Define a structure **Student** with members **roll no**, **name** and **average_mark**.
 - a) Re-define the structure name with Stud.
 - b) Create a variable Std of type Stud.
 - c) Read values for the members of Std
 - d) Print all the details.
2. Define a structure **Book** with members **book_id**, **book_title**, **author**, **price**.
 - a) Define a variable of type Book.
 - b) Using a call-by-reference function **read()**, store details of the book.
 - c) Using a call-by-value function **print()**, print the book details.
3. Define a structure **Bank** with members **account_no**, **name**, **type_account**(Savings/Current), **balance**.
 - a) Store 5 customer details [Use structure array].
 - b) Using a function **search()** that takes structure array as argument and **account_no**, search if that **account_no** exist or not. If so print the details.
 - c) Using another function **sort_balance()**, sort the structure array on the basis of **balance** and print the details.
4. Define a structure **Employee** with members **empno**, **emp_name**, **position**(Manager(M),Supervisor(S),Ordinary(O)), **basic_pay**.
 - a) Read a value **n** from the user and store **n** number of employee details.
 - b) Use function **Read()** to read the employee details.
 - c) Use function **Display()** print the employee details.
 - d) Use a function **Search()** which should give user a varieties of choices on the basis of which will display the employee details. i.e, **empno** or **emp_name** or **position** or **basic pay**. If **position** is selected all the employees belonging to that category should be displayed. If **basic pay** is selected all the employees within that **basic pay** should be displayed.
5. Define a structure **Date_Admit** with the **day**, **month** and **year** as members.
Define another structure **Patient** with members **ipno**, **name**, **dateofadmit** which is of type **Date_Admit** structure.
 - a) Store **n** patient details.
 - b) Using a function **detail()** show all the patient details admitted within a period of time as per users input. [Ex: 04 01 2013 and 01 11 2013. If the user inputs date print the patient details admitted in between these periods.]
6. (Use dynamic memory allocation) Define a structure to represent a two dimensional point:

```
typedef struct point {  
    int x, y;  
} Point;
```

- A polygon can be stored in an array of points, one point for each vertex of the polygon. Write a program which does the following:
- Reads in the number of vertices in a polygon from the user.
- Dynamically allocates an array of the appropriate number of point structures.
- Reads the coordinates of each vertex into the array of points.
- Computes and prints the total length of all the edges of the polygon. To compute the length of a single edge, compute the distance between the two end points of the edge using the following formula: $\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

7. Cash Register (Pointer to Structure)

Implement a very simple cash register using a structure to store the data. Write several functions which operate on the cash register structure. Each function will take a pointer to a cash register structure as one of its parameters.

The structure and some sample function prototypes are shown below. The `init()` function should load the specified bills into the cash register. The `add_notes()` and `remove_notes()` functions add or remove the given numbers of notes. `value()` computes the total amount stored, and `inventory()` function prints out the number of each type of bill and the total value. Have the `inventory()` function make use of `value()` instead of doing the calculation itself.

```
typedef struct CashRegister {
    int tens;
    int fives;
    int ones;
} CashRegister;
```

```
void Init(CashRegister *cr, int tens, int fives, int ones); void
add_notes(CashRegister *cr, int tens, int fives, int ones); void
remove_notes(CashRegister *cr, int tens, int fives, int ones); int
value(CashRegister *cr);
void inventory(CashRegister *cr);
```