

CSE 2001: Data Structure & Algorithms

Programming Assignment-VII

(Dynamic Memory Allocation & Structure)

1. Write a C program to add two integers using dynamic memory allocation.
2. Write a C program using dynamic memory allocation to read an array of integers and find the maximum and minimum element of the array.
3. Write a C program to create a structure named as *Student* having field *regd_no*, *stud_name*, *mark1*, *mark2*, *mark3* of three subjects. Calculate the total and average of marks obtained by each student. Output your result under suitable headings. For each student, print the *regd_no*, name, the three marks, the average and a message “*Pass*” or “*Fail*” (A student passes if the average is greater than or equal to 50).
4. A way to represent a *point* in C with two doubles using a structure is as follows.

```
struct Point {  
    double x, y;  
};
```

- (a) Write a C program that would assign values to the individual members and display two points in the following form.

(2.5, 5.7) (1.5, 3.2)

- (b) Then, pass the structure as parameter through a user defined function and calculate the *Euclidean distance* between them. The signature of the function is as follows.

double getDistance (struct Point a, struct Point b)

Hints: According to the Euclidean distance formula, the distance (AB) between two points in the plane with coordinates A(x_1 , y_1) and B(x_2 , y_2) is given by

$$AB = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

5. Modify the above program (Q. 4) using pointers to structure and dynamic memory allocation.

6. Write a C program using structure with necessary declarations and functions to implement the addition, subtraction, multiplication and division of two complex numbers.

The structure declaration should be as follows.

```
struct complex
{
    float real;
    float imag;
} a, b;
```

The function definition for addition of two complex number should be as follows.

```
struct complex add (struct complex a, struct complex b)
{
    struct complex tmp;
    tmp.real = a.real + b.real;
    tmp.imag = a.imag + b.imag;
    return(tmp);
}
```

Similarly define other functions to perform different operations over complex numbers.

7. Modify the above program (Q. 6) using *typedef*

Hints: One may define a structure data-type with a single name

```
typedef struct newtype {
    member-variable1;
    member-variable2;
    .
    member-variableN;
} mytype;
```

mytype is the name of the new data-type

8. Modify the above program (Q. 6) using pointers to structure and dynamic memory allocation.
