LAB - 5

Structures and Unions

AIM:

To understand the working of Structures and Unions.

OBJECTIVES:

- To understand the logic of Structures.
- To understand the logic of Unions.
- To realise the difference between Structures & Unions

PROGRAMS:

- Experiment - 1:

To find sum, difference, product and display complex numbers, using structure.

Input / Output:

Get two complex numbers from the user and print the sum, difference, product of the complex num.

Code:

```
struct Complex {
    float real, imag;
};
type-def struct Complex complex;

void print(complex a) {
    if (a.imag >= 0) {
        printf("%f + i%f", a.real, a.imag);
    }else {
        printf ("%f -i%f", a.real, a.imag*(0-1));
    }
}
complex add (complex *u, complex *v) {
    complex z;
    z.real = u->real + v->real;
    z.imag = u->imag + v->imag; return z; }
```

```
complex subtract (complex *u, complex *v){
        complex z;
        z.real = u->real - v->real ;
        z.imag = u->real - v->imag ;
        return z;
}
complex multiply (complex *u, complex *v){
        complex z;
        z.real = u->real * v->real;
        z.real -= u->imag * v->imag;
        z.imag = u->imag * v->~~imag~~real ;
        z.imag = u->real * v->imag ;
        return z;
}
```

Test Cases

- Input : 1, 2 | 1, 2
- Output : complex - 1 = 1.00 + i2.00
  complex -2 = 1.00 + i2.00
  Addition = 2.00 + i4.00
  Subtraction : 0.00 + i0.00
  Multiply = -3.00 + i4.00

- Experiment - 2 :

Stores data of employees in a structure and display values.

Input / Output :

Get details of employees and store in structs. Print the employees working in Chennai and average age of all employees.

Code :

```
typdef struct Employee empData ;
void print_chennaites (empData *empdata ,int emp-count){
    int count =0;
    char *Chennai = "Chennai";
    while (count ++ <emp-count){
        if(strstr (empdata->prof_data.city, Chennai)!=NULL){
            printf("( %d)", count);
            printf("%s", empdata-> Name);
            printf ("%d", empdata -> Personal_data.age);
        }
        empdata++;
        printf(" \n");
    }
}
float average_age (empdata *empdata, int emp-count){
    float avg = 0.0;
    int count = 0;
    while( count ++ <emp-count ) {
        avg += (float) (empdata-> personal_data.age);
        empdata ++;
    } avg /= empcount;
    return avg;
}
```

Test Cases :

- Input : emp.txt | with employee details.
- Output : People from Chennai, average age of employees.

• Experiment - 8 :

Working of Unions and how it is different from structures.

Explanation of Output:

This programs shows us that :

• In an Union datatype. all members share the same memory space

• The size of the union is determined by the size of the largest member.

• Unions are useful when you want to save memory and only need at least one member at a time.

• Unions are different from Structs .. structs creates seperate memory fore every member and enables it to be accessed any time.

REMARKS :

• type def are used to create an alias for existing datatypes,

CONCLUSION :

The workings of Structure and Unions are understanded by working these exersises.