

EXPERIMENT : 7 STRUCTURES & UNION

- 1). Write a C Program that uses functions to perform the following operations:
- Reading a Complex number.
  - Writing a Complex number.
  - Addition and subtraction of two complex numbers.

NOTE: Represent complex number using a structure

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

```
struct Complex {  
    float real;  
    float imag;  
};
```

```
struct Complex readComplex() {  
    struct Complex c;  
    printf("Enter real part: ");  
    scanf("%f", &c.real);  
    printf("Enter imaginary part: ");  
    scanf("%f", &c.imag);  
    return c;  
}
```

```
void writeComplex(struct Complex c) {  
    if (c.imag >= 0)  
        printf("%.2f + %.2fi\n", c.real,  
            c.imag);  
}
```

```
else
    printf ("%2f - %2fi\n", c.real, c.imag);
}

struct Complex addComplex (struct Complex c1,
struct complex c2) {
    struct complex result;
    result.real = c1.real + c2.real;
    result.imag = c1.imag + c2.imag;
    return result;
}

struct complex subComplex (struct Complex c1,
struct complex c2) {
    struct complex result;
    result.real = c1.real - c2.real;
    result.imag = c1.imag - c2.imag;
    return result;
}

int main () {
    struct complex num1, num2, sum, diff;

    printf ("Enter first complex number: \n");
    num1 = readcomplex ();

    printf ("Enter second complex number: \n");
    num2 = readcomplex ();

    sum = addComplex (num1, num2);
```



Experiment No. \_\_\_\_\_

Name: \_\_\_\_\_

PAGE NO.:

DATE: / /

```
diff = sub(complex (num1, num2));
```

```
printf ("\n First complex Number: ");  
write complex (num1);
```

```
printf ("\n Second complex Number: ");  
write complex (num2);
```

```
printf ("\n Sum: ");  
write complex (sum);
```

```
printf ("\n Difference: ");  
write complex (diff);
```

```
return 0;  
}
```

Output :- Enter first complex number:

Enter real part: 3

Enter imaginary part: 4

Enter Second complex number:

Enter real part: 1

Enter imaginary part: 2

First Complex Number:  $3.00 + 4.00i$

Second Complex Number:  $1.00 + 2.00i$

Sum:  $4.00 + 6.00i$

Difference:  $2.00 + 2.00i$

Teacher's Signature: \_\_\_\_\_



```
1 #include <stdio.h>
2
3 struct Complex {
4     float real;
5     float imag;
6 };
7
8 struct Complex readComplex() {
9     struct Complex c;
10    printf("Enter real part: ");
11    scanf("%f", &c.real);
12    printf("Enter imaginary part: ");
13    scanf("%f", &c.imag);
14    return c;
15 }
16
17 void writeComplex(struct Complex c) {
18     if (c.imag >= 0)
19         printf("%.2f + %.2fi\n", c.real, c.imag);
20     else
21         printf("%.2f - %.2fi\n", c.real, -c.imag);
22 }
23
24 struct Complex addComplex(struct Complex c1, struct Complex c2) {
25     struct Complex result;
26     result.real = c1.real + c2.real;
27     result.imag = c1.imag + c2.imag;
28     return result;
29 }
30
```

Enter first complex number:

Enter real part: 3

Enter imaginary part: 4

Enter second complex number:

Enter real part: 1

Enter imaginary part: 2

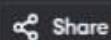
First Complex Number: 3.00 + 4.00i

Second Complex Number: 1.00 + 2.00i

Sum: 4.00 + 6.00i

Difference: 2.00 + 2.00i

=== Code Execution Successful ===



```
29 }
30
31 struct Complex subComplex(struct Complex c1, struct Complex c2) {
32     struct Complex result;
33     result.real = c1.real - c2.real;
34     result.imag = c1.imag - c2.imag;
35     return result;
36 }
37
38 int main() {
39     struct Complex num1, num2, sum, diff;
40
41     printf("Enter first complex number:\n");
42     num1 = readComplex();
43
44     printf("\nEnter second complex number:\n");
45     num2 = readComplex();
46
47     sum = addComplex(num1, num2);
48     diff = subComplex(num1, num2);
49
50     printf("\nFirst Complex Number: ");
51     writeComplex(num1);
52
53     printf("Second Complex Number: ");
54     writeComplex(num2);
55
56     printf("\nSum: ");
57     writeComplex(sum);
58 }
```

```
Enter first complex number:
Enter real part: 3
Enter imaginary part: 4
```

```
Enter second complex number:
Enter real part: 1
Enter imaginary part: 2
```

```
First Complex Number: 3.00 + 4.00i
Second Complex Number: 1.00 + 2.00i
```

```
Sum: 4.00 + 6.00i
Difference: 2.00 + 2.00i
```

```
=== Code Execution Successful ===
```



```
35     return result;
36 }
37
38 int main() {
39     struct Complex num1, num2, sum, diff;
40
41     printf("Enter first complex number:\n");
42     num1 = readComplex();
43
44     printf("\nEnter second complex number:\n");
45     num2 = readComplex();
46
47     sum = addComplex(num1, num2);
48     diff = subComplex(num1, num2);
49
50     printf("\nFirst Complex Number: ");
51     writeComplex(num1);
52
53     printf("Second Complex Number: ");
54     writeComplex(num2);
55
56     printf("\nSum: ");
57     writeComplex(sum);
58
59     printf("Difference: ");
60     writeComplex(diff);
61
62     return 0;
63 }
64
```

Enter first complex number:  
Enter real part: 3  
Enter imaginary part: 4

Enter second complex number:  
Enter real part: 1  
Enter imaginary part: 2

First Complex Number: 3.00 + 4.00i  
Second Complex Number: 1.00 + 2.00i

Sum: 4.00 + 6.00i  
Difference: 2.00 + 2.00i

=== Code Execution Successful ===





2. Write a C program to compute the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employee's name and gross salary.

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

```
#define EMP_COUNT 10
```

```
struct Employee {  
    char name[50];  
    float basic-pay;  
    float da;  
    float gross-salary;  
};
```

```
int main () {
```

```
    struct Employee emp[EMP_COUNT];
```

```
    int i;
```

```
    printf("Enter details of %d employees: \n",  
           EMP_COUNT);
```

```
    for (i = 0; i < EMP_COUNT; i++) {
```

```
        printf("\nEmployee %d\n", i+1);
```

```
        printf("Enter name: ");
```

```
        scanf("%s", emp[i].name);
```

```
        printf("Enter basic pay: ");
```

```
        scanf("%f", &emp[i].basic-pay);
```

Teacher's Signature: \_\_\_\_\_

```
emp[i].da = 0.52f * emp[i].basic - pay;
```

```
emp[i].gross - salary = emp[i].basic - pay +  
emp[i].da;
```

```
}
```

```
printf("\n%-20s%-15s\n", "Employee  
Name", "Gross Salary");
```

```
printf("\n%-20s%-15s\n", "Employee  
Name", "Gross Salary\n");
```

```
for(i = 0; i < EMP - COUNT; i++) {
```

```
printf("%-20sRs. %.2f\n", emp[i].name,  
emp[i].gross - salary);
```

```
}
```

```
return 0;
```

```
}
```



Experiment No. \_\_\_\_\_ Name: \_\_\_\_\_

PAGE NO.:

DATE: / /

Output :- Enter details of 100 Employees:

Employee 1

Enter name : Rahul

Enter basic pay : 20000

Employee 2

Enter Name : Aryan

Enter basic pay : 25000

Employee 3

Enter Name : Rena

Enter basic pay : 18000

...

Employee Name

Gross Salary

-----

Rahul

Rs. 30400.00

Aryan

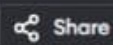
Rs. 38000.00

Rena

Rs. 27360.00

...

Teacher's Signature: \_\_\_\_\_



```
1 #include <stdio.h>
2
3 #define EMP_COUNT 3
4
5 struct Employee {
6     char name[50];
7     float basic_pay;
8     float da;
9     float gross_salary;
10 };
11
12 int main() {
13     struct Employee emp[EMP_COUNT];
14     int i;
15
16     printf("Enter details of %d employees:\n", EMP_COUNT);
17
18     for (i = 0; i < EMP_COUNT; i++) {
19         printf("\nEmployee %d\n", i + 1);
20         printf("Enter name: ");
21         scanf("%s", emp[i].name);
22         printf("Enter basic pay: ");
23         scanf("%f", &emp[i].basic_pay);
24
25         emp[i].da = 0.52f * emp[i].basic_pay;
26
27         emp[i].gross_salary = emp[i].basic_pay + emp[i].da;
28     }
29
30     printf("\n%-20s%-15s\n", "Employee Name", "Gross Salary");
```

Enter details of 3 employees:

Employee 1

Enter name: Rahul

Enter basic pay: 20000

Employee 2

Enter name: Aryan

Enter basic pay: 25000

Employee 3

Enter name: Rena

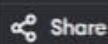
Enter basic pay: 18000

Employee Name	Gross Salary
---------------	--------------

Rahul	Rs. 30400.00
Aryan	Rs. 38000.00
Rena	Rs. 27360.00

=== Code Execution Successful ===





```
9   float gross_salary;
10 };
11
12 int main() {
13     struct Employee emp[EMP_COUNT];
14     int i;
15
16     printf("Enter details of %d employees:\n", EMP_COUNT);
17
18     for (i = 0; i < EMP_COUNT; i++) {
19         printf("\nEmployee %d\n", i + 1);
20         printf("Enter name: ");
21         scanf("%s", emp[i].name);
22         printf("Enter basic pay: ");
23         scanf("%f", &emp[i].basic_pay);
24
25         emp[i].da = 0.52f * emp[i].basic_pay;
26
27         emp[i].gross_salary = emp[i].basic_pay + emp[i].da;
28     }
29
30     printf("\n%-20s%-15s\n", "Employee Name", "Gross Salary");
31     printf("-----\n");
32     for (i = 0; i < EMP_COUNT; i++) {
33         printf("%-20sRs. %.2f\n", emp[i].name, emp[i].gross_salary);
34     }
35
36     return 0;
37 }
38
```

Enter details of 3 employees:

Employee 1

Enter name: Rahul

Enter basic pay: 20000

Employee 2

Enter name: Aryan

Enter basic pay: 25000

Employee 3

Enter name: Rena

Enter basic pay: 18000

Employee Name	Gross Salary
---------------	--------------

Rahul	Rs. 30400.00
-------	--------------

Aryan	Rs. 38000.00
-------	--------------

Rena	Rs. 27360.00
------	--------------

=== Code Execution Successful ===



Search





- 3). Create a Book structure containing book-id, title, author name and price. Write a C program to pass a structure as a function argument and print the book details.

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

```
struct Book {
```

```
    int book-id;
```

```
    char title[50];
```

```
    char author[50];
```

```
    float price;
```

```
};
```

```
void displayBook (struct Book b) {
```

```
    printf ("\n Book details : \n");
```

```
    printf ("Book ID : %d\n", b.book-id);
```

```
    printf ("Title : %s\n", b.title);
```

```
    printf ("Author : %s\n", b.author);
```

```
    printf ("Price : %.2f\n", b.price);
```

```
}
```

```
int main () {
```

```
    struct Book b1;
```

```
    printf ("Enter Book ID : ");
```

```
    scanf ("%d\n", &b1.book-id);
```

```
    getch ();
```

Experiment No. \_\_\_\_\_

Name: \_\_\_\_\_

PAGE NO.:

DATE: / /

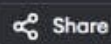
```
printf ("Enter Book title : ");  
fgets (b1.title, sizeof (b1.title), stdin);  
  
printf ("Enter Author Name : ");  
fgets (b1.author, sizeof (b1.author), stdin);  
  
printf ("Enter book price : ");  
scanf ("%f", &b1.price);  
  
display Book (b1);  
return 0;  
}
```

Output :- Enter Book ID : 101  
Enter Book Title : C Programming  
Enter Author Name : Dennis Ritchie  
Enter Book Price : 500.00

Book Details :

Book ID : 101  
Title : C Programming  
Author : Dennis Ritchie  
Price : 500.00

Teacher's Signature: \_\_\_\_\_



```
1  #include <stdio.h>
2
3  struct Book {
4      int book_id;
5      char title[50];
6      char author[50];
7      float price;
8  };
9
10 void displayBook(struct Book b) {
11     printf("\nBook Details:\n");
12     printf("Book ID : %d\n", b.book_id);
13     printf("Title : %s\n", b.title);
14     printf("Author : %s\n", b.author);
15     printf("Price : %.2f\n", b.price);
16 }
17
18 int main() {
19     struct Book b1;
20
21     printf("Enter Book ID: ");
22     scanf("%d", &b1.book_id);
23     getchar();
24
25     printf("Enter Book Title: ");
26     fgets(b1.title, sizeof(b1.title), stdin);
27
28     printf("Enter Author Name: ");
29     fgets(b1.author, sizeof(b1.author), stdin);
30 }
```

```
Enter Book ID: 101
Enter Book Title: C Programming
Enter Author Name: Dennis Ritchie
Enter Book Price: 500.00
```

Book Details:

Book ID : 101

Title : C Programming

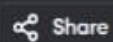
Author : Dennis Ritchie

Price : 500.00

=== Code Execution Successful ===







```
9
10 void displayBook(struct Book b) {
11     printf("\nBook Details:\n");
12     printf("Book ID   : %d\n", b.book_id);
13     printf("Title      : %s\n", b.title);
14     printf("Author     : %s\n", b.author);
15     printf("Price      : %.2f\n", b.price);
16 }
17
18 int main() {
19     struct Book b1;
20
21     printf("Enter Book ID: ");
22     scanf("%d", &b1.book_id);
23     getchar();
24
25     printf("Enter Book Title: ");
26     fgets(b1.title, sizeof(b1.title), stdin);
27
28     printf("Enter Author Name: ");
29     fgets(b1.author, sizeof(b1.author), stdin);
30
31     printf("Enter Book Price: ");
32     scanf("%f", &b1.price);
33
34     displayBook(b1);
35
36     return 0;
37 }
38
```

```
Enter Book ID: 101
Enter Book Title: C Programming
Enter Author Name: Dennis Ritchie
Enter Book Price: 500.00
```

Book Details:

```
Book ID   : 101
Title      : C Programming

Author     : Dennis Ritchie

Price      : 500.00
```

=== Code Execution Successful ===



- 4). Create a union containing 6 strings: name, home-address, hostel-address, city, state and zip. Write a C program to display your present address

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

```
union Address {
    char name [50];
    char home-address [100];
    char hostel-address [100];
    char city [50];
    char state [50];
    char zip [10];
};
```

```
int main ( ) {
    union Address addr;
```

```
strcpy (addr.hostel-address, "Room No. 1001,
    Girls Hostel, University Campus");
```

```
printf ("Present address: %s\n", addr.hostel-
    address);
```

```
return 0;
}
```

Experiment No. \_\_\_\_\_ Name: \_\_\_\_\_

PAGE NO.:

DATE: / /

Output:- Present Address :

Room No. 1001, Girls Hostel,  
University campus

Teacher's Signature: \_\_\_\_\_





```
1 #include <stdio.h>
2 #include <string.h>
3
4 union Address {
5     char name[50];
6     char home_address[100];
7     char hostel_address[100];
8     char city[50];
9     char state[50];
10    char zip[10];
11 };
12
13 int main() {
14     union Address addr;
15
16     strcpy(addr.hostel_address, "Room No. 1001, Girls Hostel, University Campus");
17     printf("Present Address: %s\n", addr.hostel_address);
18
19     return 0;
20 }
21
```

Present Address: Room No. 1001, Girls Hostel, University Campus

=== Code Execution Successful ===



Search

