

Homework Hustlers: <https://discord.gg/aJ55rZBV>

- Wizard.

Create a structure to hold any complex number $x+iy$. Write a program that uses the structure to read two complex numbers and display a third complex number.

```
~

#include <stdio.h>

struct cmplx {
    float real;
    float imag;
};

int main() {
    struct cmplx num1, num2;

    printf("enter real and imaginary parts: ");
    scanf("%f %f", &num1.real, &num1.imag);
    printf("enter real and imaginary parts: ");
    scanf("%f %f", &num2.real, &num2.imag);

    printf("= %.2f + %.2fi\n", (num1.real + num2.real), (num1.imag+num2.imag));
    return 0;
}
```

```
~

[wizard@archlinux workshop]$ ./a.out
enter real and imaginary parts: 6.9 4.20
enter real and imaginary parts: 6.9 4.20
= 13.80 + 8.40i
[wizard@archlinux workshop]$
```

Write a program that uses the above structure to input two complex numbers and pass them to function, which returns the sum of entered complex numbers in the main function and displays the sum.

~

```
#include <stdio.h>

struct cmplx {
    float real;
    float imag;
};

struct cmplx cmplx_func(struct cmplx num1, struct cmplx num2) {
    return (struct cmplx) {num1.real + num2.real, num1.imag + num2.imag};
}

int main() {
    struct cmplx num1, num2, sum;

    printf("enter real and imaginary parts: ");
    scanf("%f %f", &num1.real, &num1.imag);
    printf("enter real and imaginary parts: ");
    scanf("%f %f", &num2.real, &num2.imag);

    sum = cmplx_func(num1, num2);
    printf("Sum = %.2f + %.2fi\n", sum.real, sum.imag);

    return 0;
}
```

~

```
[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
enter real and imaginary parts: 2 4
enter real and imaginary parts: 4 2
Sum = 6.00 + 6.00i
[wizard@archlinux workshop]$
```

Create a structure named student that has a name, roll number and marks as members. Assume appropriate types and size of members. Use this structure to read and display records of 10 students. Create two functions: One is to read information of students and other to display the information.

~

```
#include <stdio.h>

typedef struct {
    char name[10];
    short roll;
    short marks;
} student;

void takeInput(student* std){
    printf("Enter name roll marks: ");
    scanf("%s %hd %hd", std->name, &std->roll, &std->marks);
}

void print(student* std){
    printf("name: %s, Roll: %hd, Marks: %hd\n", std->name, std->roll, std->marks);
}

int main() {
    student stds[10];
    for (int i = 0; i < 10; i++) { takeInput(&stds[i]); }
    for (int i = 0; i < 10; i++) { print(&stds[i]); }
    return 0;
}
```

~

```
[wizard@archlinux workshop]$ ./a.out
Enter name roll marks: a 1 100
Enter name roll marks: b 2 100
Enter name roll marks: c 3 100
Enter name roll marks: d 4 100
Enter name roll marks: e 5 100
Enter name roll marks: f 6 100
Enter name roll marks: g 7 100
Enter name roll marks: h 8 100
Enter name roll marks: i 9 100
Enter name roll marks: j 0 100
name: a, Roll: 1, Marks: 100
name: b, Roll: 2, Marks: 100
name: c, Roll: 3, Marks: 100
name: d, Roll: 4, Marks: 100
name: e, Roll: 5, Marks: 100
name: f, Roll: 6, Marks: 100
name: g, Roll: 7, Marks: 100
name: h, Roll: 8, Marks: 100
name: i, Roll: 9, Marks: 100
name: j, Roll: 0, Marks: 100
[wizard@archlinux workshop]$
```

WAP to input name, post and salary of ten employees from main() function and pass to structure type user defined function(argument of this function should also be a structure type). This function returns a structure variable which keeps the record of only those employees whose salary is greater than 10,000. This modified record is also displayed from the main() function.

~

```
#include <stdio.h>

struct employee {
    char name[10];
    char post[10];
    float salary;
};

void high_sals(struct employee employees[], struct employee high_sal_emp[], int* count) {
    for (int i = 0; i < 10; i++) {
        if (employees[i].salary > 10000) {
            high_sal_emp[*count] = employees[i];
            (*count)++;
        }
    }
}

int main() {
    struct employee employees[10], high_sal_emp[10];
    int count = 0;

    for (int i = 0; i < 10; i++) {
        printf("Enter name, post, and salary: ");
        scanf("%9s %9s %f", employees[i].name, employees[i].post, &employees[i].salary);
    }
    high_sals(employees, high_sal_emp, &count);
    printf("salary > 10000:\n");
    for (int i = 0; i < count; i++) { printf("Name: %s\n", high_sal_emp[i].name); }

    return 0;
}
```

~

```
[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
Enter name, post, and salary: a a 1
Enter name, post, and salary: b b 2
Enter name, post, and salary: c c 3
Enter name, post, and salary: d d 4
Enter name, post, and salary: e e 5
Enter name, post, and salary: f f 6
Enter name, post, and salary: g g 69
Enter name, post, and salary: h h 10000
Enter name, post, and salary: i i 10001
Enter name, post, and salary: j j 100000

Employees with salary > 10,000:
Name: i
Name: j
[wizard@archlinux workshop]$
```

Write a program that asks the user for two inputs: lower limit and upper limit. Now Write a function named display that prints all the numbers between those limits.

~

```
#include <stdio.h>

struct limits {
    int l, u;
};

void display(struct limits l) {
    for (int i = l.l; i <= l.u; i++) printf("%d ", i);
}

int main() {
    struct limits l;
    scanf("%d %d", &l.l, &l.u);
    display(l);
    return 0;
}
```

~

```
[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
1 5
1 2 3 4 5
[wizard@archlinux workshop]$
```

Write a structure to store the names, salary and hours of work per day of 5 employees in a company. Write a program to increase the salary depending on the number of hours of work per day as follows and then print the name of all the employees along with their final salaries.

Hours Worked	Salary Increase
8	\$50
10	\$100
>= 12	\$150

~

```
#include <stdio.h>

struct employee {
    char name[30];
    float salary;
    int hours;
};

int main() {
    struct employee e[5];
    for (int i = 0; i < 5; i++) {
        scanf("%s %f %d", e[i].name, &e[i].salary, &e[i].hours);

        if (e[i].hours >= 12) e[i].salary += 150;
        else if (e[i].hours >= 10) e[i].salary += 100;
        else if (e[i].hours >= 8) e[i].salary += 50;
    }
    for (int i = 0; i < 5; i++) printf("%s %.2f\n", e[i].name, e[i].salary);
    return 0;
}
```

~

```
[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
someone 150 1
someoneelse 200 9
someonealso 300 10
wizard 10000 24
idkanyomore 10 10

someone 150.00
someoneelse 250.00
someonealso 400.00
wizard 10150.00
idkanyomore 110.00
[wizard@archlinux workshop]$
```

Create a structure named Employee having members name,salary and hours of work per day. Now, write a program to dynamically create an ‘n’ number of structures of type Employee. Pass this array of structure to a function that prints the highest salary of the employee.

~

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

struct Employee {
    char name[30];
    float salary;
    int hours;
};

void highest_salary(struct Employee* e, int n) {
    float max = 0;
    for (int i = 1; i < n; i++) if (e[i].salary > max) max = e[i].salary;
    printf("Highest salary: %.2f\n", max);
}

int main() {
    int n;
    scanf("%d", &n);
    struct Employee* e = malloc(n * sizeof(struct Employee));
    for (int i = 0; i < n; i++) scanf("%s %f %d", e[i].name, &e[i].salary, &e[i].hours);
    highest_salary(e, n);
    free(e);
    return 0;
}
```

~

```
[wizard@archlinux workshop]$ gcc main.c
[wizard@archlinux workshop]$ ./a.out
3
a 150 1
b 250 2
c 350 3
Highest salary: 350.00
[wizard@archlinux workshop]$
```