

The screenshot displays the Programiz C Online Compiler interface. The top navigation bar includes the Programiz logo, a link to 'C Online Compiler', a banner for 'Premium Coding Courses by Programiz', and a 'Programiz PRO' button. The main workspace is divided into three sections: a file explorer on the left showing 'main.c', a code editor in the center, and an output console on the right. The code in 'main.c' is a C program that collects information for two students and calculates the average of their marks. The output console shows the program's execution, including prompts for student information and the final average calculation. A 'Clear' button is located in the top right of the output console.

```
1 #include<stdio.h>
2 struct std {
3     char name[100];
4     int age;
5     float marks;
6 };
7 int main(){
8     struct std std1,std2;float average;
9     printf("enter the information for student1\n");
10    printf("name:");
11    fgets(std1.name,100,stdin);
12    printf("age:");
13    scanf("%d",&std1.age);
14    printf("marks:");
15    scanf("%f",&std1.marks);
16    getchar();
17    printf("\nenter the information for student2\n");
18    printf("name: ");
19    fgets(std2.name,100,stdin);
20    printf("age:");
21    scanf("%d",&std2.age);
22    printf("marks:");
23    scanf("%f",&std2.marks);
24    printf("\nstudent1 information\n");
25    printf("name:%s",std1.name);
26    printf("age:%d\n",std1.age);
27    printf("marks:%f",std1.marks);
28    printf("\nstudent2 information\n");
29    printf("name:%s",std2.name);
30    printf("age:%d\n",std2.age);
31    printf("marks:%f\n",std2.marks);
32    average=(std1.marks+std2.marks)/2;
33    printf("average marks of two students is %f",average);
34    return 0;
35 }
```

Output

```
enter the information for student1
name:m.madhan mohan rao
age:18
marks:80

enter the information for student2
name: p.pavan
age:19
marks:87

Student1 information
name:m.madhan mohan rao
age:18
marks:80.000000
student2 information
name:p.pavan
age:19
marks:87.000000
average marks of two students is 83.500000

--- Code Execution Successful ---
```

1).
#include<stdio.h>
struct std {
char name[100];
int age;
float marks;
};
int main(){
struct std std1,std2;float average;
printf("enter the information for student1\n");
printf("name:");
fgets(std1.name,100,stdin);
printf("age:");
scanf("%d",&std1.age);
printf("marks:");
scanf("%f",&std1.marks);
getchar();
printf("\nenter the information for student2\n");
printf("name: ");
fgets(std2.name,100,stdin);
printf("age:");
scanf("%d",&std2.age);
printf("marks:");
scanf("%f",&std2.marks);

```

printf("\nstudent1 information\n");
printf("name:%s",std1.name);
printf("age:%d\n",std1.age);
printf("marks:%f",std1.marks);
printf("\nstudent2 information\n");
printf("name:%s",std2.name);
printf("age:%d\n",std2.age);
printf("marks:%f\n",std2.marks);
average=(std1.marks+std2.marks)/2;
printf("average marks of two students is %f",average);
return 0;
}

```

The screenshot displays the Programiz Online Compiler interface. On the left, a sidebar shows icons for file management and a list of languages including C, C++, Java, JavaScript, Python, and PHP. The main editor area shows a C program named 'main.c' with the following code:

```

1 #include <stdio.h>
2 struct Car {
3     int carID;
4     char model[50];
5     float rentalRate;
6 };
7 int main() {
8     struct Car cars[3];
9     int days;
10    float totalCost;
11    for (int x = 0; x < 3; x++)
12    {
13        printf("Enter Car ID, Model, and Rental Rate per Day for Car %d: ", x + 1);
14        scanf("%d %s %f", &cars[x].carID, cars[x].model, &cars[x].rentalRate);
15    }
16    printf("Enter the number of rental days: ");
17    scanf("%d", &days);
18    printf("\nRental Details:\n");
19    for (int x = 0; x < 3; x++)
20    {
21        totalCost = cars[x].rentalRate * days;
22        printf("Car ID: %d, Model: %s, Total Cost: %.2f\n", cars[x].carID, cars[x].model,
23              totalCost);
24    }
25    return 0;
26 }

```

On the right, the 'Output' panel shows the program's execution results:

```

Enter Car ID, Model, and Rental Rate per Day for Car 1: 4563
SUV
1000
Enter Car ID, Model, and Rental Rate per Day for Car 2: 5987
VAN
2500
Enter Car ID, Model, and Rental Rate per Day for Car 3: 2365
SEDAN
999
Enter the number of rental days: 5

Rental Details:
Car ID: 4563, Model: SUV, Total Cost: 5000.00
Car ID: 5987, Model: VAN, Total Cost: 12500.00
Car ID: 2365, Model: SEDAN, Total Cost: 4995.00

--- Code Execution Successful ---

```

The interface also features a 'Premium Coding Courses by Programiz' banner at the top and a 'Programiz PRO' button in the top right corner.

2).

```

#include <stdio.h>
struct Car {
    int carID;
    char model[50];
    float rentalRate;

```

```

};
int main() {
    struct Car cars[3];
    int days;
    float totalCost;
    for (int x = 0; x < 3; x++)
    {
        printf("Enter Car ID, Model, and Rental Rate per Day for Car %d: ", x + 1);
        scanf("%d %s %f", &cars[x].carID, cars[x].model, &cars[x].rentalRate);
    }

    printf("Enter the number of rental days: ");
    scanf("%d", &days);
    printf("\nRental Details:\n");
    for (int x = 0; x < 3; x++)
    {
        totalCost = cars[x].rentalRate * days;
        printf("Car ID: %d, Model: %s, Total Cost: %.2f\n", cars[x].carID, cars[x].model, totalCost);
    }

    return 0;
}

```

The screenshot displays the Programiz Online Compiler interface. The editor on the left contains a C program named `main.c` that defines a `Complex` struct and implements functions for adding and multiplying complex numbers. The `main` function prompts the user to enter the real and imaginary parts of two complex numbers, performs the operations, and prints the results. The output window on the right shows the program's execution with the following text:

```

Enter real and imaginary parts of the first complex number: 4
5
Enter real and imaginary parts of the second complex number: 4
5
Sum: 8+10i
Product: -9+40i

```

Below the output, a message states "Code Execution Successful". The right sidebar features the Programiz logo and a link to "Premium Courses by Programiz".

3).

```
#include <stdio.h>
struct Complex {
    int real;
    int imag;
};
struct Complex add(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 multiply(struct Complex c1, struct Complex c2) {
    struct Complex result;
    result.real = c1.real * c2.real - c1.imag * c2.imag;
    result.imag = c1.real * c2.imag + c1.imag * c2.real;
    return result;
}
int main() {
    struct Complex c1, c2, sum, product;
    printf("Enter real and imaginary parts of the first complex number: ");
    scanf("%d %d", &c1.real, &c1.imag);
    printf("Enter real and imaginary parts of the second complex number: ");
    scanf("%d %d", &c2.real, &c2.imag);
    sum = add(c1, c2);
    product = multiply(c1, c2);
    printf("Sum: %d+%di\n", sum.real, sum.imag);
    printf("Product: %d+%di\n", product.real, product.imag);
    return 0;
}
```

The screenshot displays the Programiz Online Compiler interface. On the left, a sidebar contains icons for file management and a list of languages including C, C++, Java, JavaScript, Python, and PHP. The main editor area shows a C program named 'main.c' with the following code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct Employee
4 {
5     int id;
6     char name[50];
7     float salary;
8 };
9 int main()
10 {
11     int n;
12     struct Employee* employees;
13     printf("Enter the number of employees: ");
14     scanf("%d", &n);
15     employees = (struct Employee*)malloc(n * sizeof(struct Employee));
16     for (int i = 0; i < n; i++)
17     {
18         printf("Enter ID, Name, and Salary of Employee %d: ", i + 1);
19         scanf("%d %s %f", &employees[i].id, employees[i].name, &employees[i].salary);
20     }
21     printf("\nEmployee Details:\n");
22     for (int i = 0; i < n; i++)
23     {
24         printf("ID: %d, Name: %s, Salary: %.2f\n", employees[i].id, employees[i].name,
25             employees[i].salary);
26     }
27     free(employees);
28     return 0;
29 }
```

On the right, the 'Output' panel shows the program's execution results:

```
Enter the number of employees: 2
Enter ID, Name, and Salary of Employee 1: 345
sagar
30000
Enter ID, Name, and Salary of Employee 2: 346
pavan
40000

Employee Details:
ID: 345, Name: sagar, Salary: 30000.00
ID: 346, Name: pavan, Salary: 40000.00

--- Code Execution Successful ---
```

Below the output, a sidebar promotes 'Premium Courses by Programiz' with a 'Learn More' button.

4).

```
#include <stdio.h>
#include <stdlib.h>
struct Employee
{
    int id;
    char name[50];
    float salary;
};
int main()
{
    int n;
    struct Employee* employees;
    printf("Enter the number of employees: ");
    scanf("%d", &n);
    employees = (struct Employee*)malloc(n * sizeof(struct Employee));
    for (int i = 0; i < n; i++)
    {
        printf("Enter ID, Name, and Salary of Employee %d: ", i + 1);
        scanf("%d %s %f", &employees[i].id, employees[i].name, &employees[i].salary);
    }
    printf("\nEmployee Details:\n");
    for (int i = 0; i < n; i++)
```

```
{  
    printf("ID: %d, Name: %s, Salary: %.2f\n", employees[i].id, employees[i].name,  
employees[i].salary);  
}  
free(employees);  
return 0;  
}
```