

Sample Screenshots:

The screenshot shows a VS Code editor with a C file named `Jimenez_Lab2.c`. The code defines a function `add_fractions` that takes two fractions and returns their sum. The `main` function prompts the user to enter the number of test cases and then for each case, the numerators and denominators of two fractions. It then prints the results of the addition.

```
48 int main()
49 {
50     int test_cases;
51     scanf("%d", &test_cases);
52     int denominators[test_cases][2];
53     for (int i = 0; i < test_cases; i++)
54     {
55         printf("Enter the numerators and denominators of the two fractions for case #%d (separated by spaces): ", i + 1);
56         scanf("%d %d %d %d", &numerators[i][0], &denominators[i][0], &numerators[i][1], &denominators[i][1]);
57     }
58     printf("\nResults:\n");
59     for (int i = 0; i < test_cases; i++)
60     {
61         printf("Case #%d: ", i + 1);
62         add_fractions(numerators[i][0], denominators[i][0], numerators[i][1], denominators[i][1]);
63         printf("\n");
64     }
65     return 0;
66 }
```

The terminal output shows the program being compiled and executed. The user enters 3 test cases. For each case, two fractions are entered. The results are printed as Case #1: 3/4, Case #2: 7/8, and Case #3: 3, 3/4.

```
PS C:\Users\chean\Downloads> cd "c:\Users\chean\Downloads\" ; if ($?) { gcc Jimenez_Lab2.c -o Jimenez_Lab2 } ; if ($?) { .\Jimenez_Lab2 }
Enter the number of test cases: 3
Enter the numerators and denominators of the two fractions for case #1 (separated by spaces): 1 2 1 4
Enter the numerators and denominators of the two fractions for case #2 (separated by spaces): 3 8 1 2
Enter the numerators and denominators of the two fractions for case #3 (separated by spaces): 7 2 1 4

Results:
Case #1: 3/4
Case #2: 7/8
Case #3: 3, 3/4
PS C:\Users\chean\Downloads>
```

The screenshot shows a VS Code editor with a C file named `Jimenez_Lab2.c`. The code defines a function `add_fractions` that takes two fractions and returns their sum. The `main` function prompts the user to enter the number of test cases and then for each case, the numerators and denominators of two fractions. It then prints the results of the addition.

```
48 int main()
49 {
50     int test_cases;
51     scanf("%d", &test_cases);
52     int denominators[test_cases][2];
53     for (int i = 0; i < test_cases; i++)
54     {
55         printf("Enter the numerators and denominators of the two fractions for case #%d (separated by spaces): ", i + 1);
56         scanf("%d %d %d %d", &numerators[i][0], &denominators[i][0], &numerators[i][1], &denominators[i][1]);
57     }
58     printf("\nResults:\n");
59     for (int i = 0; i < test_cases; i++)
60     {
61         printf("Case #%d: ", i + 1);
62         add_fractions(numerators[i][0], denominators[i][0], numerators[i][1], denominators[i][1]);
63         printf("\n");
64     }
65     return 0;
66 }
```

The terminal output shows the program being compiled and executed. The user enters 1 test case. For the first case, two fractions are entered. The result is printed as Case #1: 11/15.

```
PS C:\Users\chean\Downloads> cd "c:\Users\chean\Downloads\" ; if ($?) { gcc Jimenez_Lab2.c -o Jimenez_Lab2 } ; if ($?) { .\Jimenez_Lab2 }
Enter the number of test cases: 1
Enter the numerators and denominators of the two fractions for case #1 (separated by spaces): 1 3 2 5

Results:
Case #1: 11/15
PS C:\Users\chean\Downloads>
```

Source Code:

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

```
// Function to calculate the greatest common divisor (GCD) of two numbers
```

```
int gcd(int a, int b)
```

```
{
```

```
    if (b == 0)
```

```
        return a;
```

```
    return gcd(b, a % b);
```

```
}
```

```
// Function to simplify a fraction
```

```
void simplify_fraction(int *numerator, int *denominator)
```

```
{
```

```
    int common_divisor = gcd(*numerator, *denominator);
```

```
    *numerator /= common_divisor;
```

```
    *denominator /= common_divisor;
```

```
}
```

```
// Function to add two fractions and display the result
```

```
void add_fractions(int num1, int denom1, int num2, int denom2)
```

```
{
```

```
    int result_num = num1 * denom2 + num2 * denom1;
```

```
    int result_denom = denom1 * denom2;
```

```
    simplify_fraction(&result_num, &result_denom);
```

```
// Display the result in proper format
```

```
if (result_num >= result_denom)
```

```
{
```

```

int whole_part = result_num / result_denom;
int fractional_part = result_num % result_denom;

if (fractional_part == 0)
{
    printf("%d", whole_part);
}
else
{
    printf("%d, %d/%d", whole_part, fractional_part, result_denom);
}
}
else
{
    printf("%d/%d", result_num, result_denom);
}
}

int main()
{
    int test_cases;

    printf("Enter the number of test cases: ");
    scanf("%d", &test_cases);

    int numerators[test_cases][2];
    int denominators[test_cases][2];

    for (int i = 0; i < test_cases; i++)
    {

```

```
    printf("Enter the numerators and denominators of the two fractions for case #%%d (separated by spaces): ", i + 1);
```

```
    scanf("%%d %%d %%d %%d", &numerators[i][0], &denominators[i][0], &numerators[i][1], &denominators[i][1]);
```

```
}
```

```
printf("\nResults:\n");
```

```
for (int i = 0; i < test_cases; i++)
```

```
{
```

```
    printf("Case #%%d: ", i + 1);
```

```
    add_fractions(numerators[i][0], denominators[i][0], numerators[i][1], denominators[i][1]);
```

```
    printf("\n");
```

```
}
```

```
return 0;
```

```
}
```