Program 1

Code: complex.c

```
#include <stdio.h>

#include <stdlib.h>


// The complex_t structure which will be used to define complex
numbers

struct complex_t {double real, imaginary;};


// Prototypes for multiply and divide functions

struct complex_t multiply(struct complex_t a, struct complex_t b);

struct complex_t* divide(struct complex_t *a, struct complex_t *b);


int main (void)

{

	struct complex_t a, b, product, *quotient;


	// User inputs the two complex numbers. For all print functions,
"i" is printed at the end to avoid it preceeding the negative sign.

	printf("This program will use complex numbers composed of a real
component 'r' and an imaginary component 'bi'.\n");

	printf("Please enter both complex numbers in the form 'r + bi',
separated by a comma: ");

	scanf("%lg + %lfi, %lf + %lfi", &a.real, &a.imaginary, &b.real,
&b.imaginary);


	// Two complex_t structures for the results of the product and
quotient of the two complex numbers

	product = multiply(a, b);
```

```
    quotient = divide(&a, &b);


    printf("Given the complex numbers: %lg + %lgi & %lg + %lgi,\n",
a.real, a.imaginary, b.real, b.imaginary);


    // Printing the product result.

    printf("The product of the complex numbers is: %lg + %lgi \n",
product.real, product.imaginary);


    // Printing the quotient result.

    printf("The quotient of the complex numbers is: %lg + %lgi\n",
quotient->real, quotient->imaginary);

    // Now that quotient is used, free up the memory

    free(quotient);


    // Since we deallocated the memory, quotient points to nothing,
so we will assign it to null

    quotient = NULL;


    return 0;

}


// A function which computes and returns the complex product of two
complex numbers

struct complex_t multiply(struct complex_t a, struct complex_t b)

{

    struct complex_t complexProduct;


    // Calculate the real component of the complex product:
```

```
	complexProduct.real = (a.real * b.real) - (a.imaginary * b.imaginary);


	// Calculate the imaginary component of the complex product:
	complexProduct.imaginary = (a.real * b.imaginary) + (a.imaginary * b.real);


	// Returns the complex product
	return complexProduct;
}


// A function which computes and returns a pointer to the complex quotient of two complex numbers
struct complex_t* divide(struct complex_t *a, struct complex_t *b)
{
	struct complex_t *complexQuotient = malloc(sizeof(struct complex_t));


	// Check to make sure that the malloc succeeded in allocating memory before continuing
	if (complexQuotient)
	{
		double divisor = b->real * b->real + b->imaginary * b->imaginary;


		// To avoid dividing by zero, preform a check
		if (divisor > 0)
		{
			// Calculate the real component of the complex quotient:
```

```
                complexQuotient->real = (a->real * b->real + a-
>imaginary * b->imaginary) / divisor;


                // Calculate the imaginary component of the complex
quotient:
                complexQuotient->imaginary = (b->real * a->imaginary -
a->real * b->imaginary) / divisor;
            }


            // If the second complex number was zero, division by zero
will occur, thus print out a message and exit the program
            else
            {
                printf("Error: Division by zero.\nTerminating...\n");
                exit(EXIT_FAILURE);
            }
      }
      // If the malloc failed, print an error message and exit the
program
      else
      {
            printf("Failed to allocate memory
block.\nTerminating...\n");
            exit(EXIT_FAILURE);
      }


      // Returns the complex quotient pointer
      return complexQuotient;
}
```

Code: complexTypeDef.c

```
#include <stdio.h>

#include <stdlib.h>


// The complex_t structure which will be used to define complex
numbers

typedef struct

{

      double real, imaginary;

} complex_t;


// Prototypes for multiply and divide functions

complex_t multiply(complex_t a, complex_t b);

complex_t* divide(complex_t *a, complex_t *b);


int main (void)

{

      complex_t a, b, product, *quotient;


      // User inputs the two complex numbers. For all print functions,
"i" is printed at the end to avoid it preceeding the negative sign.

      printf("This program will use complex numbers composed of a real
component 'r' and an imaginary component 'bi'.\n");

      printf("Please enter both complex numbers in the form 'r + bi',
separated by a comma: ");

      scanf("%lg + %lfi, %lf + %lfi", &a.real, &a.imaginary, &b.real,
&b.imaginary);
```

```
	// Two complex_t structures for the results of the product and
quotient of the two complex numbers
	product = multiply(a, b);
	quotient = divide(&a, &b);

	printf("Given the complex numbers: %lg + %lgi & %lg + %lgi,\n",
a.real, a.imaginary, b.real, b.imaginary);

	// Printing the product result.
	printf("The product of the complex numbers is: %lg + %lgi \n",
product.real, product.imaginary);

	// Printing the quotient result.
	printf("The quotient of the complex numbers is: %lg + %lgi\n",
quotient->real, quotient->imaginary);
	// Now that quotient is used, free up the memory
	free(quotient);

	// Since we deallocated the memory, quotient points to nothing,
so we will assign it to null
	quotient = NULL;

	return 0;
}

// A function which computes and returns the complex product of two
complex numbers
complex_t multiply(complex_t a, complex_t b)
{
```

```
	complex_t complexProduct;

	// Calculate the real component of the complex product:
	complexProduct.real = (a.real * b.real) - (a.imaginary * 
b.imaginary);

	// Calculate the imaginary component of the complex product:
	complexProduct.imaginary = (a.real * b.imaginary) + (a.imaginary 
* b.real);

	// Returns the complex product
	return complexProduct;
}

// A function which computes and returns a pointer to the complex 
quotient of two complex numbers
complex_t* divide(complex_t *a, complex_t *b)
{
	complex_t *complexQuotient = malloc(sizeof(complex_t));

	// Check to make sure that the malloc succeeded in allocating 
memory before continuing
	if (complexQuotient)
	{
		double divisor = b->real * b->real + b->imaginary * b-
>imaginary;
				// To avoid dividing by zero, preform a check
		if (divisor > 0)
		{
```

```
                // Calculate the real component of the complex
quotient:
                complexQuotient->real = (a->real * b->real + a-
>imaginary * b->imaginary) / divisor;


                // Calculate the imaginary component of the complex
quotient:
                complexQuotient->imaginary = (b->real * a->imaginary -
a->real * b->imaginary) / divisor;
           }


           // If the second complex number was zero, division by zero
will occur, thus print out a message and exit the program
           else
           {
                printf("Error: Division by zero.\nTerminating...\n");
                exit(EXIT_FAILURE);
           }
      }
      // If the malloc failed, print an error message and exit the
program
      else
      {
           printf("Failed to allocate memory
block.\nTerminating...\n");
           exit(EXIT_FAILURE);
      }
           // Returns the complex quotient pointer
      return complexQuotient;
}
```

Cases (using complex):

Two complex numbers

obelix[12]% complex

This program will use complex numbers composed of a real component 'r' and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a comma: 1 + 2.4i, 3.0 + 4.67i
Given the complex numbers: 1 + 2.4i & 3 + 4.67i,
The product of the complex numbers is: -8.208 + 11.87i
The quotient of the complex numbers is: 0.461165 + 0.0821191i

Two real numbers

obelix[15]% complex

This program will use complex numbers composed of a real component 'r' and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a comma: 4 + 0i, 12 + 0i
Given the complex numbers: 4 + 0i & 12 + 0i,
The product of the complex numbers is: 48 + 0i
The quotient of the complex numbers is: 0.333333 + 0i

Two imaginary numbers

obelix[18]% complex

This program will use complex numbers composed of a real component 'r' and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a comma: 0 + 4i, 0 + 7i
Given the complex numbers: 0 + 4i & 0 + 7i,
The product of the complex numbers is: -28 + 0i
The quotient of the complex numbers is: 0.571429 + 0i

First number is real, second number is imaginary

```
obelix[19]% complex

This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a
comma: 10 + 0i, 0 + 5i
Given the complex numbers: 10 + 0i & 0 + 5i,
The product of the complex numbers is: 0 + 50i
The quotient of the complex numbers is: 0 + -2i
```

First number is imaginary, second number is real

```
obelix[20]% complex

This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a
comma: 0 + 9i, 3 + 0i
Given the complex numbers: 0 + 9i & 3 + 0i,
The product of the complex numbers is: 0 + 27i
The quotient of the complex numbers is: 0 + 3i
```

First number is zero, second number is complex

```
obelix[21]% complex

This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a
comma: 0 + 0i, 5 + 23i
Given the complex numbers: 0 + 0i & 5 + 23i,
The product of the complex numbers is: 0 + 0i
The quotient of the complex numbers is: 0 + 0i
```

First number is complex, second number is zero

```
obelix[23]% complex

This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a
comma: 3 + 4i, 0 + 0i
Error: Division by zero.
Terminating...
```

First number is complex, second number is imaginary

```
obelix[24]% complex

This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a
comma: 5 + 6i, 0 + 3i
Given the complex numbers: 5 + 6i & 0 + 3i,
The product of the complex numbers is: -18 + 15i
The quotient of the complex numbers is: 2 + -1.66667i
```

First number is imaginary, second number is complex

```
obelix[25]% complex

This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a
comma: 0 + 3i, 5 + 6i
Given the complex numbers: 0 + 3i & 5 + 6i,
The product of the complex numbers is: -18 + 15i
The quotient of the complex numbers is: 0.295082 + 0.245902i
```

First number is zero, second number is real

obelix[26]% complex

```
This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
```
Please enter both complex numbers in the form 'r + bi', separated by a comma: 0 + 0i, 4 + 0i
```
Given the complex numbers: 0 + 0i & 4 + 0i,
The product of the complex numbers is: 0 + 0i
The quotient of the complex numbers is: 0 + 0i
```

First number is real, second number is zero

obelix[27]% complex

```
This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
```
Please enter both complex numbers in the form 'r + bi', separated by a comma: 4 + 0i, 0 + 0i
```
Error: Division by zero.
Terminating...
```

Cases (using complexTypeDef):

Two complex numbers

obelix[12]% complex

This program will use complex numbers composed of a real component 'r' and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a comma: 1 + 2.4i, 3.0 + 4.67i
Given the complex numbers: 1 + 2.4i & 3 + 4.67i,
The product of the complex numbers is: -8.208 + 11.87i
The quotient of the complex numbers is: 0.461165 + 0.0821191i

Two real numbers

obelix[15]% complex

This program will use complex numbers composed of a real component 'r' and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a comma: 4 + 0i, 12 + 0i
Given the complex numbers: 4 + 0i & 12 + 0i,
The product of the complex numbers is: 48 + 0i
The quotient of the complex numbers is: 0.333333 + 0i

Two imaginary numbers

obelix[18]% complex

This program will use complex numbers composed of a real component 'r' and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a comma: 0 + 4i, 0 + 7i
Given the complex numbers: 0 + 4i & 0 + 7i,
The product of the complex numbers is: -28 + 0i
The quotient of the complex numbers is: 0.571429 + 0i

First number is real, second number is imaginary

obelix[19]% complex

```
This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
```
Please enter both complex numbers in the form 'r + bi', separated by a comma: 10 + 0i, 0 + 5i
```
Given the complex numbers: 10 + 0i & 0 + 5i,
The product of the complex numbers is: 0 + 50i
The quotient of the complex numbers is: 0 + -2i
```

First number is imaginary, second number is real

obelix[20]% complex

```
This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
```
Please enter both complex numbers in the form 'r + bi', separated by a comma: 0 + 9i, 3 + 0i
```
Given the complex numbers: 0 + 9i & 3 + 0i,
The product of the complex numbers is: 0 + 27i
The quotient of the complex numbers is: 0 + 3i
```

First number is zero, second number is complex

obelix[21]% complex

```
This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
```
Please enter both complex numbers in the form 'r + bi', separated by a comma: 0 + 0i, 5 + 23i
```
Given the complex numbers: 0 + 0i & 5 + 23i,
The product of the complex numbers is: 0 + 0i
The quotient of the complex numbers is: 0 + 0i
```

First number is complex, second number is zero

```
obelix[23]% complex

This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a
comma: 3 + 4i, 0 + 0i
Error: Division by zero.
Terminating...
```

First number is complex, second number is imaginary

```
obelix[24]% complex

This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a
comma: 5 + 6i, 0 + 3i
Given the complex numbers: 5 + 6i & 0 + 3i,
The product of the complex numbers is: -18 + 15i
The quotient of the complex numbers is: 2 + -1.66667i
```

First number is imaginary, second number is complex

```
obelix[25]% complex

This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
Please enter both complex numbers in the form 'r + bi', separated by a
comma: 0 + 3i, 5 + 6i
Given the complex numbers: 0 + 3i & 5 + 6i,
The product of the complex numbers is: -18 + 15i
The quotient of the complex numbers is: 0.295082 + 0.245902i
```

First number is zero, second number is real

obelix[26]% complex

```
This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
```
Please enter both complex numbers in the form 'r + bi', separated by a comma: 0 + 0i, 4 + 0i

```
Given the complex numbers: 0 + 0i & 4 + 0i,
The product of the complex numbers is: 0 + 0i
The quotient of the complex numbers is: 0 + 0i
```

First number is real, second number is zero

obelix[27]% complex

```
This program will use complex numbers composed of a real component 'r'
and an imaginary component 'bi'.
```
Please enter both complex numbers in the form 'r + bi', separated by a comma: 4 + 0i, 0 + 0i

```
Error: Division by zero.
Terminating...
```