1. Write a function that will read 2 numbers and calculate and display the sum and difference.

```
#include <stdio.h>
void calculateSumAndDifference() {
 int num1, num2;
 printf("Enter two numbers: ");
 scanf("%d %d", &num1, &num2);
 int sum = num1 + num2;
 int difference = num1 - num2;
 printf("Sum: %d\n", sum);
 printf("Difference: %d\n", difference);
}
int main() {
 calculateSumAndDifference();
 return 0;
}
```

2. Write a function that accepts 2 numbers as parameters and calculates and displays the sum and difference.

```
#include <stdio.h>
void calculateSumAndDifference(int num1, int num2) {
 int sum = num1 + num2;
 int difference = num1 - num2;
 printf("Sum: %d\n", sum);
 printf("Difference: %d\n", difference);
}
int main() {
 int num1, num2;
 printf("Enter two numbers: ");
 scanf("%d %d", &num1, &num2);
 calculateSumAndDifference(num1, num2);
 return 0;
}
```

3. Write a function that accepts 2 whole numbers as parameters and calculates and returns the product.

```
#include <stdio.h>
int calculateProduct(int num1, int num2) {
 return num1 * num2;
}
int main() {
 int num1, num2;
 printf("Enter two numbers: ");
 scanf("%d %d", &num1, &num2);
 int product = calculateProduct(num1, num2);
 printf("Product: %d\n", product);
 return 0;
}
```

4. Write a function that accepts 2 whole numbers as parameters and calculates and returns the quotient.

```
#include <stdio.h>
float calculateQuotient(int num1, int num2) {
 if (num2 == 0) {
 printf("Error: Division by zero.\n");
 return 0;
 }
 return (float)num1 / num2;
}
int main() {
 int num1, num2;
 printf("Enter two numbers: ");
 scanf("%d %d", &num1, &num2);
 float quotient = calculateQuotient(num1, num2);
 printf("Quotient: %.2f\n", quotient);
 return 0;
}
```

5. Write a function to read 2 numbers and display the sum. Call this function from the main function several times.

```
#include <stdio.h>
void displaySum() {
 int num1, num2;
 printf("Enter two numbers: ");
 scanf("%d %d", &num1, &num2);
 int sum = num1 + num2;
 printf("Sum: %d\n", sum);
}
int main() {
 int i;
 for (i = 0; i < 3; i++) {
 displaySum();
 }
 return 0;
}
```

6. Write a function which accepts 2 integers as parameters and display the sum, difference, and product using a single printf statement.

```
#include <stdio.h>
void calculateAndDisplay(int num1, int num2) {
 int sum = num1 + num2;
 int difference = num1 - num2;
 int product = num1 * num2;
 printf("Sum: %d, Difference: %d, Product: %d\n", sum, difference, product);
}
int main() {
 int num1, num2;
 printf("Enter two numbers: ");
 scanf("%d %d", &num1, &num2);
 calculateAndDisplay(num1, num2);
 return 0;
}
```

7. Write a function which accepts an integer and a float value as parameters and returns the product as a double value. Display the result from the main function.

```
#include <stdio.h>
double calculateProduct(int num1, float num2) {
 return num1 * num2;
}
int main() {
 int num1;
 float num2;
 printf("Enter an integer and a float value: ");
 scanf("%d %f", &num1, &num2);
 double product = calculateProduct(num1, num2);
 printf("Product: %.2lf\n", product);
 return 0;
}
```

8. Give the function header for each of the following functions:

a.) Function `hypotenuse` that takes two double-precision floating-point arguments, `side1` and `side2`, and returns a double-precision floating-point result.

```
double hypotenuse(double side1, double side2);
```

b.) Function `smallest` that takes three integers, `x`, `y`, `z`, and returns an integer.

```
int smallest(int x, int y, int z);
```

c.) Function `instructions` that does not receive any arguments and does not return a value.

```
void instructions(void);
```

d.) Function `intToFloat` that takes an integer argument, `number`, and returns a floating-point result.

```
float intToFloat(int number);
```