Functions

1. Calculate Simple Interest

```
#include <stdio.h>

// Function to calculate simple interest
float calculate_simple_interest(float principal, float rate, float time) {
    if (principal < 0 || rate < 0 || time < 0) {
        printf("ERROR: Negative values are not allowed.\n");
        return -1;
    }
    return (principal * rate * time) / 100;
}
int main() {
    float principal, rate, time, interest, total_amount;
    printf("Enter principal amount: ");
    scanf("%f", &principal);
    printf("Enter interest rate (per annum): ");
    scanf("%f", &rate);
    printf("Enter time period (in years): ");
    scanf("%f", &time);
    interest = calculate_simple_interest(principal, rate, time);
    if (interest >= 0) {
        total_amount = principal + interest;
        printf("Simple Interest: %.2f\n", interest);
        printf("Total Amount after %.1f years: %.2f\n", time, total_amount);
    }
    return 0;
```

```
}
```

2. Calculate Power of a Number

```
#include <stdio.h>
// Function to calculate a^n
int power(int a, int n) {
    if (n < 0) {
        printf("ERROR: Negative exponent not supported.\n");
        return -1;
    }
    int result = 1;
    for (int i = 0; i < n; i++) {
        result *= a;
    }
    return result;
}
int main() {
    int base, exponent, result;
    printf("Enter base (a): ");
    scanf("%d", &base);
    printf("Enter exponent (n): ");
    scanf("%d", &exponent);
    result = power(base, exponent);
    if (result != -1) {
        printf("%d raised to the power %d is: %d\n", base, exponent, result);
    }
    return 0;
```

```
}
```

3. Multiply Two Numbers Without Using * Operator

```
#include <stdio.h>
// Function to multiply using addition
int multiply(int a, int b) {
    int result = 0;
    int sign = 1;

    // Handle negative numbers
    if (a < 0) {
        a = -a;
        sign = -sign;
    }
    if (b < 0) {
        b = -b;
        sign = -sign;
    }

    // Add 'a' to itself 'b' times
    for (int i = 0; i < b; i++) {
        result += a;
    }
    return sign * result;
}
int main() {
    int num1, num2, product;
```

```
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    product = multiply(num1, num2);
    printf("Product of %d and %d is: %d\n", num1, num2, product);
    return 0;
}
```

4. Quotient of Division

```
#include <stdio.h>
// Function to calculate quotient
int quotient(int a, int b) {
    if (b == 0) {
        printf("ERROR: Division by zero is not allowed.\n");
        return -1;
    }
    return a / b;
}
int main() {
    int num1, num2, result;
    printf("Enter dividend (a): ");
    scanf("%d", &num1);
    printf("Enter divisor (b): ");
    scanf("%d", &num2);
    result = quotient(num1, num2);
    if (result != -1) {
        printf("Quotient of %d / %d is: %d\n", num1, num2, result);
    }
```

```
    return 0;
}
```

5. Remainder of Division

```
#include <stdio.h>
// Function to calculate remainder
int remainder(int a, int b) {
    if (b == 0) {
        printf("ERROR: Division by zero is not allowed.\n");
        return -1;
    }
    return a % b;
}
int main() {
    int num1, num2, result;
    printf("Enter dividend (a): ");
    scanf("%d", &num1);
    printf("Enter divisor (b): ");
    scanf("%d", &num2);
    result = remainder(num1, num2);
    if (result != -1) {
        printf("Remainder of %d / %d is: %d\n", num1, num2, result);
    }
    return 0;
}
```

6. Multiplication Table

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

```
// Function to print multiplication table
void print_multiplication_table(int n) {
   printf("Multiplication table for %d:\n", n);
   for (int i = 1; i <= 10; i++) {
      printf("%d × %d = %d\n", n, i, n * i);
   }
}
int main() {
   int num;
   printf("Enter a number: ");
   scanf("%d", &num);
   print_multiplication_table(num);
   return 0;
}
```

7. Check if a Number is Prime

```
#include <stdio.h>
// Function to check if a number is prime
int isPrime(int n) {
   int i;
   // Check for negative numbers and special cases
   if (n <= 1) {
      return 0;  // Not prime
   }
   if (n <= 3) {
      return 1;  // Prime
   }
```

```
    if (n % 2 == 0 || n % 3 == 0) {
        return 0;  // Not prime
    }
    // Check divisibility by numbers of form 6k±1
    for (i = 5; i * i <= n; i += 6) {
        if (n % i == 0 || n % (i + 2) == 0) {
            return 0;  // Not prime
        }
    }
    return 1;  // Prime
}
int main() {
    int num, result;
    printf("Enter a number: ");
    scanf("%d", &num);
    result = isPrime(num);
    if (result == 1) {
        printf("%d is a prime number.\n", num);
    } else {
        printf("%d is not a prime number.\n", num);
    }
    return 0;
}
```

8. Check if a Character is Alphanumeric

```
#include <stdio.h>
// Function to check if character is alphanumeric
```

```
int fun_alpha_num(char c) {
   if ((c >= 'A' && c <= 'Z') ||
      (c >= 'a' && c <= 'z') ||
      (c >= '0' && c <= '9')) {
      return 1;  // Alphanumeric
   } else {
      return 0;  // Not alphanumeric
   }
}
int main() {
   char ch;
   int result;
   printf("Enter a character: ");
   scanf(" %c", &ch);
   result = fun_alpha_num(ch);
   if (result == 1) {
      printf("'%c' is alphanumeric.\n", ch);
   } else {
      printf("'%c' is not alphanumeric.\n", ch);
   }
   return 0;
}
```

9. Calculator Program with Multiple Functions

```
#include <stdio.h>
// Function for addition
int add(int a, int b) {
```

```
    return a + b;
}
// Function for subtraction
int sub(int a, int b) {
    return a - b;
}
// Function for multiplication
int mul(int a, int b) {
    return a * b;
}
// Function for quotient
int quotient(int a, int b) {
    if (b == 0) {
        printf("ERROR: Division by zero is not allowed.\n");
        return -1;
    }
    return a / b;
}
// Function for remainder
int remainder(int a, int b) {
    if (b == 0) {
        printf("ERROR: Division by zero is not allowed.\n");
        return -1;
    }
    return a % b;
}
```

```
int main() {
    int num1, num2, result;
    char operator;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    printf("Enter an operator (+, -, *, /, %%): ");
    scanf(" %c", &operator);
    switch(operator) {
        case '+':
            result = add(num1, num2);
            printf("%d + %d = %d\n", num1, num2, result);
            break;
        case '-':
            result = sub(num1, num2);
            printf("%d - %d = %d\n", num1, num2, result);
            break;
        case '*':
            result = mul(num1, num2);
            printf("%d * %d = %d\n", num1, num2, result);
            break;
        case '/':
            result = quotient(num1, num2);
            if (result != -1)
                printf("%d / %d = %d\n", num1, num2, result);
            break;
        case '%':
```

```
            result = remainder(num1, num2);
            if (result != -1)
                printf("%d %% %d = %d\n", num1, num2, result);
            break;
        default:
            printf("Invalid operator.\n");
    }
    return 0;
}
```

10. Check if a Year is a Leap Year

```
#include <stdio.h>
// Function to check if a year is leap year
int isLeapYear(int year) {
    if (year < 1) {
        printf("ERROR: Invalid year.\n");
        return -1;
    }
    // Leap year conditions
    if ((year % 400 == 0) || ((year % 4 == 0) && (year % 100 != 0))) {
        return 1;  // Leap year
    } else {
        return 0;  // Not a leap year
    }
}
int main() {
    int year, result;
```

```
    printf("Enter a year: ");
    scanf("%d", &year);
    result = isLeapYear(year);
    if (result == 1) {
        printf("%d is a leap year.\n", year);
    } else if (result == 0) {
        printf("%d is not a leap year.\n", year);
    }
    return 0;
}
```