C++ Exercises

1. Write a function that takes two integers as parameters and returns the sum of those numbers.
2. Write a function that takes a string as a parameter and returns the length of that string.
3. Write a function that takes an array of integers as a parameter and returns the sum of all the elements in the array.
4. Write a function that takes an array of integers as a parameter and returns the average of all the elements in the array.
5. Write a function that takes two arrays of integers as parameters and returns a new array that contains the sum of the corresponding elements in the two arrays.
6. Write a function that takes a number as a parameter and returns true if the number is prime, and false otherwise.
7. Write a function that takes a string as a parameter and returns true if the string is a palindrome, and false otherwise.
8. Write a function that takes an array of integers as a parameter and returns a new array that contains only the even numbers from the original array.
9. Write a function that takes an array of integers as a parameter and returns a new array that contains only the odd numbers from the original array.
10. Write a function that takes an array of integers as a parameter and returns a new array that contains the same elements as the original array, but sorted in ascending order.

1. Write a function that takes two integers as parameters and returns the sum of those numbers.

```
int sum(int a, int b) {
  return a + b;
}
```

2. Write a function that takes a string as a parameter and returns the length of that string.

```
int stringLength(string str) {
  int length = 0;
  while (str[length] != '\0') {
    length++;
  }
  return length;
}
```

3. Write a function that takes an array of integers as a parameter and returns the sum of all the elements in the array.

```
int arraySum(int arr[], int size) {
  int sum = 0;
  for (int i = 0; i < size; i++) {
    sum += arr[i];
  }
  return sum;
}
```

4. Write a function that takes an array of integers as a parameter and returns the average of all the elements in the array.

```
double arrayAverage(int arr[], int size) {
  double sum = 0;
  for (int i = 0; i < size; i++) {
      sum += arr[i];
  }
  return sum / size;
}
```

5. Write a function that takes two arrays of integers as parameters and returns a new array that contains the sum of the corresponding elements in the two arrays.

```
void arraySum(int arr1[], int arr2[], int size, int result[]) {
    for (int i = 0; i < size; i++) {
        result[i] = arr1[i] + arr2[i];
    }
}
```

6. Write a function that takes a number as a parameter and returns true if the number is prime, and false otherwise.

```
bool isPrime(int num) {
    if (num < 2) {
        return false;
    }
    for (int i = 2; i <= num / 2; i++) {
        if (num % i == 0) {
            return false;
        }
    }
    return true;
}
```

7. Write a function that takes a string as a parameter and returns true if the string is a palindrome, and false otherwise.

```
bool isPalindrome(string str) {
    int length = stringLength(str);
    for (int i = 0; i < length / 2; i++) {
        if (str[i] != str[length - i - 1]) {
            return false;
        }
    }
    return true;
}
```

8. Write a function that takes an array of integers as a parameter and returns a new array that contains only the even numbers from the original array.

```
void getEvenNumbers(int arr[], int size, int result[], int& resultSize) {
    resultSize = 0;
    for (int i = 0; i < size; i++) {
        if (arr[i] % 2 == 0) {
            result[resultSize] = arr[i];
            resultSize++;
        }
    }
}
```

9. Write a function that takes an array of integers as a parameter and returns a new array that contains only the odd numbers from the original array.

```
void getOddNumbers(int arr[], int size, int result[], int& resultSize) {
    resultSize = 0;
    for (int i = 0; i < size; i++) {
        if (arr[i] % 2 != 0) {
            result[resultSize] = arr[i];
            resultSize++;
        }
    }
}
```

10. Write a function that takes an array of integers as a parameter and returns a new array that contains the same elements as the original array, but sorted in ascending order.

```
void sortArray(int arr[], int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];

                arr[j + 1] = temp;

            }

        }

    }

}
```