21CSCI01P

Introduction to Computing

Lab (5)

This tutorial covers

- **Tutorial:**
    1. **functions**
- **Problems from 20 to 29 in the lecture slides.**
- **Chapter 6 in the textbook, 8th edition.**

Problem (1) ***Write a C++ program that does the following steps:***

1. In the **main()** function, receive from the user 10 integers, then save these 10 integers in an array **data[]**.
2. Then In the **main()** function, call a functions **positive()** and **negative()**. These two functions return nothing**.** Pass the array **data[]** in this function call.
3. The **positive()** function, receives an array of integers, and integer size, and returns nothing. The body of this function prints out all the positive integers that appear more than once in the received array.
4. The **negative()** function, receives an array of integers, and integer size, and returns nothing. The body of this function prints out all the negative integers that appear more than once in the received array.

Solution (1) ***Extracting data from a file to another file:***

```
#include<iostream>
…
…
```

Problem (2) *Write a C++ program that does the following steps:*

1. In the **main()** function, receive from the user 10 characters, then save these 10 characters in an array **data[]**.
2. Then In the **main()** function, call a function **capitals()** that returns nothing**,** pass the array **data[]** in this function call.
3. The **capitals()** function, receives an array of characters, and integer size, and returns nothing. The body of this function prints out all the capital letters that appear more than once in the received array.

Solution (2) *Extracting data from a file to another file:*

```
#include<iostream>
…
…
```

Problem (3) *Write a C++ program that does the following steps:*

1. In the **main()** function, receive from the user 10 integers, then save these 10 integers in an array **data[]**.
1. Then In the **main()** function, call a function **max3integer()** that returns nothing**,** pass the array **data[]**, and the value of **10** in this function call.
2. The **max3integer()** function, receives an array of integers, and integer size, and returns nothing. The body of this function replaces the largest 3 numbers in the received array by zero.
3. Then in the **main()** function, after calling the **max3integer()** function, prints out the array **data[]**.

Solution (3) *Extracting data from a file to another file:*

```
#include<iostream>
…
…
```

Problem (4) *Write a C++ program that does the following steps:*

2. In the **main()** function, receive from the user 10 integers, then save these 10 integers in an array **data[]**. Define a constant integer **size** of value 10.
3. Then In the **main()** function, call the functions **printOdds(), printEvens(), printPrimes()** , these functions return nothing**,** pass the array **data[]**, and integer **size** in this function call.
4. The **printOdds()** function, receives an array of integers, and integer size, and returns nothing. The body of this function prints out the odd numbers in the received array.
5. The **printEvens()** function, receives an array of integers, and integer size, and returns nothing. The body of this function prints out the even numbers in the received array.
6. The **printPrimes()** function, receives an array of integers, and integer size, and returns nothing. The body of this function prints out the prime numbers in the received array.

Solution (4) *Extracting data from a file to another file:*

```
#include<iostream>
…
…
```

Problem (5) *Write a C++ program that does the following steps:*

1. In the **main()** function, receive from the user 10 characters, then save these 10 characters in an array **data[]**. Define a constant integer **size** of value 10.
2. Then In the **main()** function, call a function **repeated()** that returns nothing**,** pass the array **data[]** and the integer **size** in this function call.
3. The **repeated()** function, receives an array of integers, and integer size, and returns nothing. The body of this function prints out all the characters that appear more than once in the received array.

Solution (5) *Extracting data from a file to another file:*

```
#include<iostream>
…
…
```

Problem (6) *Write a C++ program that does the following steps:*

1. In the **main()** function, receive from the user 10 integers, then save these 10 integers in an array **data1[]**. Then receive from the user another 10 integers, then save these 10 integers in an array **data2[]**. Define a constant integer **size** of value 10.
2. Then In the **main()** function, call a function **common()** that returns nothing**,** pass the arrays **data1[]** and **data2[]**, and the integer **size** in this function call.
3. The **common()** function, receives two arrays of integers, and integer size, and returns nothing. The body of this function prints out all the common integers that appear in both received array.

Solution (6) *Extracting data from a file to another file:*

```
#include<iostream>
…
…
```

Problem (7) *Write a C++ program that does the following steps:*

1. Consider a file “Numbers.txt” that includes 15 integer numbers, for example “22 42 10 3 4 33 …”.
2. In the **main()** function, define a constant integer **size** of value 15, and define an empty array of name **data** of size **size**.
3. Then In the **main()** function, call a function **readNumbers()** that returns nothing**,** Pass an empty array **data[]** and the integer **size** in this function call.
4. The **readNumbers()** function, receives an array of integers, and integer size, and returns nothing. The body of this function reads the 15 numbers in the file and save these numbers to the received array.
5. Back to the **main()** function, after calling the **readNumbers()** function, print the array **data** in reverse order.

Solution (7) *Extracting data from a file to another file:*

```
#include<iostream>
…
…
```