

# Object Oriented Programming Lab

## Lab 03

**Marks 10**

### Instructions

Work on this lab individually. You can use your books, notes, handouts etc. but you are not allowed to borrow anything from your peer student.

### Marking Criteria

Show your work to the instructor before leaving the lab to get some or full credit.

### What you must do

Program the following tasks in your C++ compiler and then compile and execute them.

### Task 1

Implement following function named *pairWiseSum*

```
int* pairWiseSum(const int ar[], const int size, int& newArraySize);
```

The parameters *ar* and *size* holds an array and its size respectively.

The function should return a pointer to newly created array which contains sum of the pairs of elements together, starting with elements at index 0 with 1, 2 with 3, 4 with 5 and so on. Store the size of new array in parameter *newArraySize*. keep the last element as it is If the size of *ar* is odd. It should store 0 (zero) in *newArraySize* and return NULL if *ar* has no elements. The function should not display anything.

For example, the input array with values

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10} results in the output array as {3, 7, 11, 15, 19}.

{1, 2, 3, 4, 5, 6, 7, 8, 9} results in the output array as {3, 7, 11, 15, 9}.

In main function declare arrays of different sizes. Fill the arrays with arbitrary values and then pass them to *pairWiseSum* function along with their sizes and all the required parameters. Display contents of the arrays returned by function *pairWiseSum* if any, otherwise display appropriate message. Don't forget to free the memory resource allocated by the program, if any.

### Task 2

Implement following function named *letterPerWord* that accepts C-string *str* as its argument. The function should count the number of characters of each word appearing in the string and display a table containing the word followed by its length on the screen. Display an appropriate message, if *str* is empty.

```
void letterPerWord(const char* str);
```

For instance, if the string argument is "All is well." the function should display.

Word	Number of Characters
All	3
is	2
well	4

In main function, ask the user to input a string (maximum 50 characters) and then pass it to *letterPerWord* function.

### Task 3

Implement following function named *reverseCase* that accepts a C-string *str* as an argument and return a newly created string which contains the reverse case of *str*. It should test each character to determine whether it is upper or lowercase. If a character is uppercase, it should be converted to lowercase. Likewise, if a character is lowercase, it should be converted to uppercase. No harm should be done to any other characters. It should return NULL in case of empty string (*str* contains no characters). The function should not display anything.

```
char* reverseCase(const char* str);
```

For instance, the string argument

"Hello, four Score And Seven years Ago" will be converted to "HELLO, FOUR sCORE aND sEVEN YEARS aGO"

"Bsf21mXyZ, pU. LahoRE" will be converted to "bSF21MxYz, Pu. IAHOrE"

Demonstrate the working of *reverseCase* function in main and display the modified string on the screen. Display an appropriate message if the returned string is empty. Don't forget to free the memory resource allocated by the program, if any.

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺