

Program each of the following tasks in your C++ compiler. Keep compiling and executing even after writing a single line of code.

Task 01

Implement the following function named **squareElements**:

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

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

The function should return a pointer to a newly created array with the following modification: each element in the new array should be the square of the corresponding element in the input array. Store the size of the new array in parameter **newArraySize**. It should store 0 (zero) in **newArraySize** and return **NULL** if **ar** has no elements.

For example, if the input array is {2, 4, 6, 8}, the output array should be {4, 16, 36, 64}.

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

Task 02

Implement a function named **pairWiseSum**:

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

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

For example:

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

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

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

Task 03

Implement the following function named **reverseCase**:

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

The function should accept a C-string **str** as an argument and return a newly created string that 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 an empty string (**str** contains no characters). The function should not display anything.

For instance:

The string argument "Hello, four Score And Seven years Ago" will be converted to "hELLO, FOUR sCORE aND sEVEN YEARS aGO".

The string argument "Bsf21mXyZ, pU. LahoRE" will be converted to "bSF21MxYz, Pu. lAHORE".

Demonstrate the working of the **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.