CS 003 - FUNDAMENTALS OF COMPUTER SCIENCE II (C++)

Lab5

SEP 12, 2023

INSTRUCTOR : JI LEE

Problem 1.

- Write a C++ program that gives the following output.

```
int A[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 };
int B[] = { 3, 6, 10, 11, 13, 15, 16, 17 };
int C[] = { 1, 2, 3, 5, 7, 11, 15 };
```

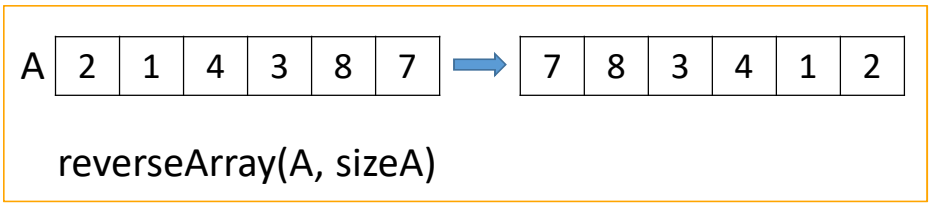
```
< Numbers in A and B, not C >
6 10

< Numbers in B and C, not A >
15

< Numbers in A and C, not B >
1 2 5 7
```

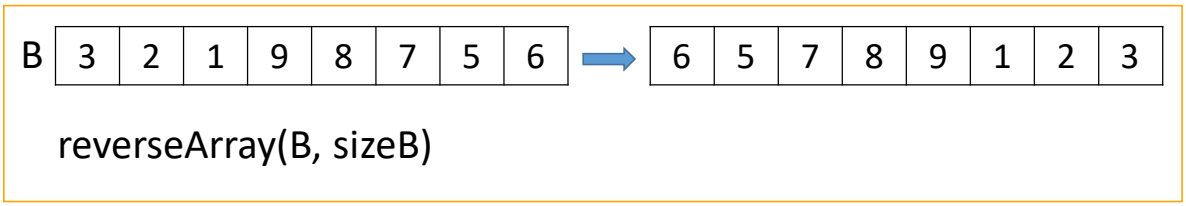
```
int A[] = { 1, 7, 10, 20, 25 };
int B[] = { 10, 25, 28 };
int C[] = { 1, 4, 7, 10, 20, 28, 30 };
```

```
< Numbers in A and B, not C >
25

< Numbers in B and C, not A >
28

< Numbers in A and C, not B >
1 7 20
```

Problem 2.

- Write a C++ program to reverse the array.





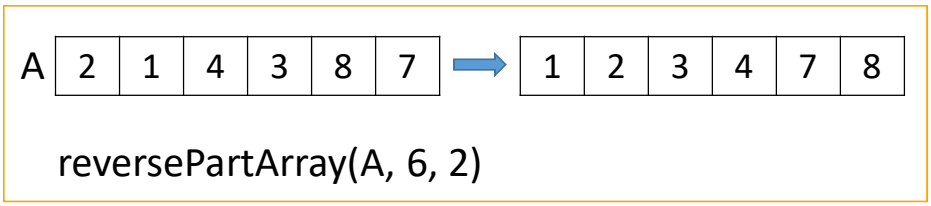

```cpp
#include <iostream>
#include <iomanip>
using namespace std;

void reverseArray(int[], int);
void printArray(int[], int);

int main()
{
    int A[] = { 2, 1, 4, 3, 8, 7 };
    int B[] = { 3, 2, 1, 9, 8, 7, 5, 6, 4 };
    int C[] = { 1, 2, 3 };

    int sizeA = sizeof(A) / sizeof(int);
    int sizeB = sizeof(B) / sizeof(int);
    int sizeC = sizeof(C) / sizeof(int);

    reverseArray(A, sizeA);
    reverseArray(B, sizeB);
    reverseArray(C, sizeC);

    printArray(A, sizeA);
    printArray(B, sizeB);
    printArray(C, sizeC);

    return 0;
}
```

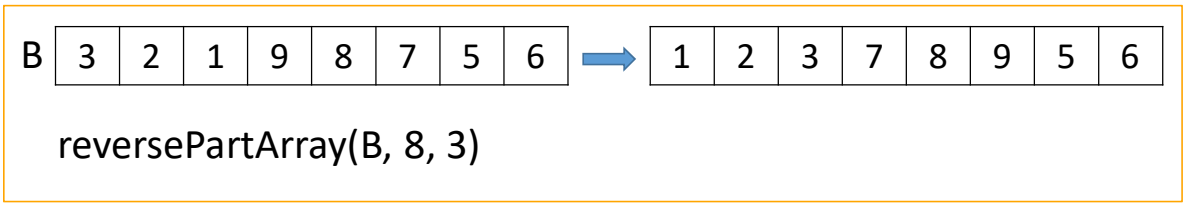
```cpp
void reverseArray(int M[], int size)
{
     // complete reverseArray function

}

void printArray(int M[], int size)
{
    for (int i = 0; i < size; i++)
        cout << setw(3) << M[i];
    cout << endl;
}
```

```
  7  8  3  4  1  2
  4  6  5  7  8  9  1  2  3
  3  2  1
```

Problem 3.

- Write a C++ program to reverse the part of the array.






```
void printArray(int[], int);
void reversePartArray(int[], int, int);

int main()
{
    int A[] = { 2, 1, 4, 3, 8, 7 };
    int B[] = { 3, 2, 1, 9, 8, 7, 5, 6 };
    int C[] = { 4, 3, 6, 8, 7, 5, 2, 1, 9, 0, 1, 2, 3 };
    int D[] = { 1, 2, 3, 4, 5 };

    // print original arrays

    // reverse arrays
    reversePartArray(A, sizeA, 2);
    reversePartArray(B, sizeB, 3);
    reversePartArray(C, sizeC, 5);
    reversePartArray(D, sizeD, 7);

    // print reversed arrays

}
```

```
< Original Array >
-------------------------------------------
A :   2  1  4  3  8  7
B :   3  2  1  9  8  7  5  6
C :   4  3  6  8  7  5  2  1  9  0  1  2  3
D :   1  2  3  4  5

< Reversed Array >
-------------------------------------------
A :   1  2  3  4  7  8
B :   1  2  3  7  8  9  5  6
C :   7  8  6  3  4  0  9  1  2  5  1  2  3
D :   1  2  3  4  5
```