

## Tutorial 10 – Pointers

Q1. Declare an integer variables  $x = 1$  and an array  $y[5] = \{10, 20, 30, 40, 50\}$ , and two integer pointers  $xPtr$  and  $yPtr$  that point to  $x$  and  $y$  respectively. Write your code to determine the following:

- What are the address of  $x$  and  $y$ ?
- What are the values stored in  $xPtr$  and  $yPtr$ ?
- What is the meaning of  $*xPtr$  and  $*yPtr$ ?
- What is the difference between  $(*yPtr + 2)$  and  $*(yPtr + 2)$ ?
- What is the following code segment doing?

```
xPtr = yPtr + 2;
(*xPtr)++;
(*yPtr)++;
```

Q2. Write a function **swapInteger()** which has two integer pointers  $a$  and  $b$  as parameters. The function interchanges the integers that are pointed to by  $a$  and  $b$ . The function prototype of **swapInteger()** is as follow:

```
void swapInteger(int *, int *);
```

Use the main function below to demonstrate the use of **swapInteger()**.

```
int main(){
    int x = 2, y = 5;

    cout << "Before swapping:" << endl;
    cout << "x is: " << x << endl;
    cout << "y is: " << y << endl;

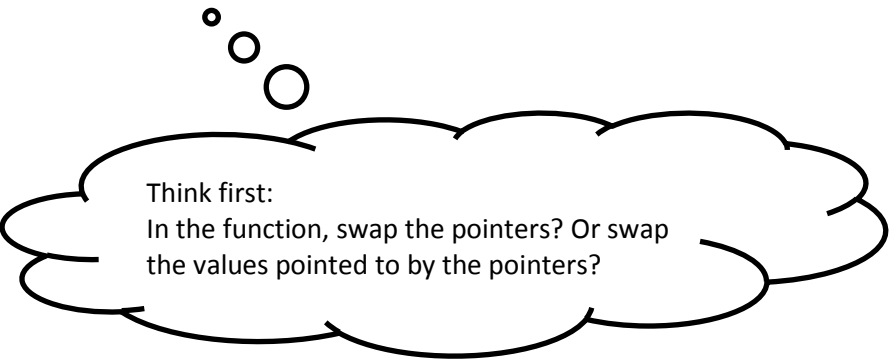
    // call swapInteger here...

    cout << "After swapping:" << endl;
    cout << "x is: " << x << endl;
    cout << "y is: " << y << endl;

    return 0;
}
```

Sample result:

```
Before swapping:
x is: 2
y is: 5
After swapping:
x is: 5
y is: 2
```



Think first:  
In the function, swap the pointers? Or swap  
the values pointed to by the pointers?

Q3. Write a function **multiplyArray(int \* const a, int size, int n)**, which multiplies each elements in the array *a* by *n*. Use **pointer arithmetic ONLY** (instead of array subscript) to access the array elements in the function.

Insert your codes into the program below to show the use of **multiplyArray()**.

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

void printArray(const int [], int);
void multiplyArray(int * const, int, int);

int main(){
    const int arraySize = 10;

    int c[arraySize] = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20};

    cout << "Before multiplyArray, array is: ";
    printArray(c, arraySize);

    // Function call on multiplyArray
    // Insert your codes here

    cout << "After multiplyArray, array is: ";
    printArray(c, arraySize);

    return 0;
}

void printArray(const int a[], int size){
    for (int i = 0; i < size; i++){
        cout << a[i] << " ";
    }
    cout << endl;
}

// Function definition of multiplyArray
// Insert your codes here
```

Below is the sample result if elements in array *c* are multiplied by 3.

**Sample result:**

Before multiplyArray, array is: 2 4 6 8 10 12 14 16 18 20

After multiplyArray, array is: 6 12 18 24 30 36 42 48 54 60

Q4. In the following program, fill in your code to allow user input of a sentence and assign it to the character pointer *buffer*. You may assume that the input sentence does not exceed 10 words and 100 characters. After the input, write your code to break the sentence into multiple words and use the array of character pointers in *word* to point to the words one by one. Note that each word should be null-terminated. You should use pointer arithmetic ONLY in this question.

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

int main() {
    char *buffer = new char [100];          // reserve 100 characters
    char *word[10] = {};                    // initialize all pointers to 0 (NULL)

    // Your code should be inserted here

    for (int i = 0; i < 10; i++)
        if (word[i] != 0)                  // check if it is a NULL pointer
            cout << i << ": " << word[i] << endl;

    return 0;
}
```

**Sample result:**

Enter a sentence with at most 10 words and 100 characters:

***C++ programming with pointer is very interesting***

0: C++

1: programming

2: with

3: pointer

4: is

5: very

6: interesting