Tutorial / Exercise – Memory, Pointers and the Heap

Exercises:

1. Look at the following definitions and initializations:

```
char c = 'T', d = 'S';
char *p1 = &c;
char *p2 = &d;
char *p3;
```

If the address of c is 6940, the address of d is 9772, and the address of e is 2224. What will be printed when the following statements are executed sequentially?

```
p3 = &d;

cout << "*p3 = " << *p3 << endl;   // (1)

p3 = p1;

cout << "*p3 = " << *p3;   // (2)
cout << ", p3 = " << p3 << endl;   // (3)

*p1 = *p2;

cout << "*p1 = " << *p1;   // (4)
cout << ", p1 = " << p1 << endl;   // (5)
```

2. Consider the following statements:

```
int *p;
int i;
int k;

i = 42;
k = i;
p = &i;
```

3. After these statements, which of the following statements will change the value of i to 75?
   - A. k = 75;
   - B. *k = 75;
   - C. p = 75;
   - D. *p = 75;
   - E. Two or more of the answers will change i to 75.

4. Explain the error.

```
char c = 'A';
double *p = &c;
```

5. Look at the following code. Give the value of the left-hand side variable in each assignment statement. Assume the lines are executed sequentially. Assume the address of the blocks array is 4434.

```
int main()
{
    char blocks[3] = {'A','B','C'};
    char *ptr = &blocks[0];
    char temp;

    temp = blocks[0];
    temp = *(blocks + 2);
    temp = *(ptr + 1);
    temp = *ptr;

    ptr = blocks + 1;
    temp = *ptr;
    temp = *(ptr + 1);

    ptr = blocks;
    temp = *++ptr;
    temp = ++*ptr;
    temp = *ptr++;
    temp = *ptr;

    return 0;
}
```

6. Write a function for each of the following descriptions. For each function, use the pointer notation ONLY. Do NOT use the array index [] notation.

   A. Write a function `RevString(char* array)` which reverses array. The function returns nothing.
   B. Write a function `CountEven(int* array, int array_len)` which receives an integer array and its size, and returns the number of even numbers in the array.
   C. Write a function `Maximum(double* array, int array_size)` that returns a pointer to the maximum value of an array of doubles. If the array is empty, return nullptr.
   D. Write a function `Contains(char* array, char search_value)` which returns true if the 1st parameter contains the 2nd parameter, or false otherwise.

7. Predict the output of the following program. Run the program to see if your prediction is right.

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

int main()
{
    int * ptr_a, ptr_b, *ptr_c;

    ptr_a = new int;
    *ptr_a = 3;
    ptr_b = ptr_a;
    cout << *ptr_a << " " << *ptr_b << "\n";

    ptr_b = new int;
    *ptr_b = 9;
    cout << *ptr_a << " " << *ptr_b << "\n";

    *ptr_b = *ptr_a;
    cout << *ptr_a << " " << *ptr_b << "\n";

    delete ptr_a;
    ptr_a = ptr_b;
    cout << *ptr_a << " " << *&*&*&*&*ptr_b << "\n";

    ptr_c = &ptr_a;
    cout << *ptr_c << " " << **ptr_c << "\n";

    delete ptr_a;
    ptr_a = NULL;

    return 0;
}
```