Name:
Worksheet#04
CSC 211 - Spring 2019

1. Definitions:

    a. Pointer - **Variable that stores to a memory address.**

    b. Memory Address - **A hexadecimal representation of a memory address.**

    c. Dereference - **Used to directly access the variable that a pointer points to.**

2. Code:

    a. Create a basic pointer named myPtr that holds the address of variable of type *string* named pizza

    string pizza;
    string *myPtr = &pizza;

    b. Print out the memory address of a variable pizza

    cout << myPtr << endl;
    cout << &pizza<< endl;

    c. Dereference the pointer named myPtr to store the value *cheese*

    ***myPtr = "cheese";**

3. Write out a memory table for the code below:

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

void flip(int *ptr1, int *ptr2){
    int temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
}


void flop(int *ptrX, int *ptrY, int *ptrZ){
    int temp = *ptrY;
    *ptrY = *ptrX;
    *ptrX = *ptrZ;
    *ptrZ = temp;
}

int main(){
    int one = 1;
    int two = 2;
    int three = 3;

    int *a = &one;
    int *b = &two;
    int *c = &three;

    cout << "a = " << *a << " is at " << a << endl;
    cout << "b = " << *b << " is at " << b << endl;
    cout << "c = " << *c << " is at " << c << endl;
    cout << " " << endl;

    flip(a, b);

    cout << "one = " << one << " is at " << a << endl;
    cout << "two = " << two << " is at " << b << endl;
    cout << "three = " << three << " is at " << c << endl;
    cout << " " << endl;

    flop(a, b, c);

    cout << "a = " << *a << " is at " << a << endl;
```

```
    cout << "b = " << *b << " is at " << b << endl;
    cout << "c = " << *c << " is at " << c << endl;


    return 0;
}
```

Main

Memory Address	Value	Variable Name
a0	null	nullptr
a1	~~1~~ -- > ~~2~~ → 3	one
a2	~~2~~ -- > ~~1~~ -- > 2	two
a3	~~3~~ → 1	three
a4	a1	a
a5	a2	b
a6	a3	c

Flip

Memory Address	Value	Variable Name
a7	a1	ptr1
a8	a2	ptr2
a9	1	temp

Flop

Memory Address	Value	Variable Name
a10	a1	`ptrX`
a11	a2	`ptrY`
a12	a3	`ptrZ`
a13	1	temp