**Objective to gain more confidence in using and manipulating C++ pointers**

1. Assume the address of variable a is 1234 and the address of variable b is 3456, what is the output of the following code fragments

    int a = 12;

    int b = 15;

    int \*pa = &a;

    int \*pb = &b;

    int \*pc;

    cout << a << endl;

    cout << pb << endl;

    pc  = &a;

    \*pc = 32;

    cout << pc << endl;

    cout << a << endl;

    pc = pb;

    cout << \*pc << endl;

1. Explain and fix the error

    char a = "A";

    double \*pa = &a;

1. What is the left-hand side in the following assignment statements (assume the address of the array is 4567)

short array[4] = {20,30,40, 50};

short \*ptr = &array[1];

short t;

cout << (int) array << endl;

t = array[0]; ;cout << t << endl; // t = ??

t = \*(array + 2); ;cout << t << endl; // t = ??

t = \*(ptr + 1); ;cout << t << endl; // t = ??

t = \*ptr; ;cout << t << endl; // t = ??

ptr = array + 1; ;cout << (int) ptr << endl; // ptr = ??

t = \*ptr; ;cout << t << endl; // t = ???

t = \*(ptr + 1); ;cout << t << endl; // t = ???

ptr = array; ;cout << (int) ptr << endl; // ptr = ???

t = \*++ptr; ;cout << t << endl; // t = ???

t = ++\*ptr; ;cout << t << endl; // t = ???

t = \*ptr++; ;cout << t << endl; // t = ???

t = \*ptr; ;cout << t << endl;

1. What is the left-hand side in the following assignment statements (assume the address of the array is 4567). Note mapping two dimensional indices (i, j) to one dimensional index (k ) is:

k = i \* COLUMNS + j

short array[2][4] = {10,20,30,40,50,60,70,80};

short \*ptr = &array[1][2];

short t;

cout << (int) array << endl;

t = array[0][0]; ;cout << t << endl; // t = ??

t = \*(array + 2); ;cout << t << endl; // t = ??

t = \*(ptr + 1); ;cout << t << endl; // t = ??

t = \*ptr; ;cout << t << endl; // t = ??

ptr = array + 1; ;cout << (int) ptr << endl; // ptr = ??

t = \*ptr; ;cout << t << endl; // t = ???

t = \*(ptr + 1); ;cout << t << endl; // t = ???

ptr = array; ;cout << (int) ptr << endl; // ptr = ???

t = \*++ptr; ;cout << t << endl; // t = ???

t = ++\*ptr; ;cout << t << endl; // t = ???

t = \*ptr++; ;cout << t << endl; // t = ???

t = \*ptr; ;cout << t << endl;