Project 6

1. \*p3 = S

\*p3 = T, p3 = 6940

\*p1 = S, p1 = 9772

1. D. \*p = 75;
2. Lines (2) and (4) did not compile because the pointers in those lines are trying to store addresses of different types from themselves. For Line 2, we are trying to store a char address in a double pointer. In Line 4, we are trying to store an int address in a char pointer.
3. **char** blocks[3] = {'A','B','C'};

**char** \*ptr = &blocks[0]; // ptr will be assigned the value: 4434

cout << ptr << endl;

**char** temp;

temp = blocks[0]; // temp will be assigned the value: A

temp = \*(blocks + 2); // temp will be assigned the value: C

temp = \*(ptr + 1); // temp will be assigned the value: B

temp = \*ptr; // temp will be assigned the value: A

ptr = blocks + 1; // ptr will be assigned the value: 4435

temp = \*ptr; // temp will be assigned the value: B

temp = \*(ptr + 1); // temp will be assigned the value: C

ptr = blocks; // ptr will be assigned the value: 4434

temp = \*++ptr; // temp will be assigned the value: B

temp = ++\*ptr; // temp will be assigned the value: C

temp = \*ptr++; // temp will be assigned the value: C

temp = \*ptr; // temp will be assigned the value: C

1. **int** num[ 6 ] = { 0, 0, 0, 0, 0, 0 };

num[0] = 100;

num[1] = 90;

num[2] = 80;

num[3] = 70;

num[4] = 60;

**for** (**int** i = 0; i <= 5; i++)

cout << num[ i ] << " ";

cout << endl;

1. **void** revString(**char**\* msg)

{

**char**\* ptr = msg;

**while** (\*msg != '\0') { //not at 0 byte at end

**if** (isdigit(\*msg) == **false**) { //not a digit

\*ptr = \*msg;

ptr++;

}

msg++; //update to go through msg

}

\*ptr = '\0'; //add 0 byte at end