Edmund Zhi

UID: 805328871

Project 6

1. cout line 1 will output “\*p3 = S” because pointer 3 points to the address of d, which is the character ‘S’.  
   cout line 2 will output “\*p3 = T, p3 = 6940” because the line “p3 = p1” makes p3 point to where p1 points, and p1 points to char c (which has address 6940), which is the character ‘T’  
   cout line 3 will print out “\*p1 = S, p1 = 6940” because the line “\*p1 = \*p2” assigns the value pointed to by p1 equal to the value pointed to by p2. p2 points to char d, which has value ‘S’. However, while the content of \*p1 changes, the address does not, which is why 6940 is still the address of p1.
2. Choice A) k = 75; does not work because we are changing the content of the value of k, which doesn’t affect the value of i. Choice B) \*k = 75 does not work because we already initialized k as an integer type and not a pointer. Choice C) p = 75 does not work because it sets the address of of the pointer p, and does not change the value. This means that Choice D) \*p = 75 does work, because it changes the value that p points to, which is i, so it changes the value of i.
3. Line 2 will not compile because you cannot initialize a pointer to a double with an address of a character. Line 4 will not compile for a similar reason, you cannot initialize a pointer to a character with the address of an int.

Text

Description automatically generated

**char** blocks[3] = {'A','B','C'}; // address: 4434

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

**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. Problem 5:

Text

Description automatically generated

// Problem 5

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

**int** value = 100;

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

num[i] = value;

value -= 10;

}

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

cout << num[j] << " ";

cout << endl;

**Text

Description automatically generated**

**void** revString(**char** \*msg) {

**char** \*first, \*last;

// makes first pointer point to the first character of the character array

first = msg;

// makes last pointer point to the last character of the character array

last = msg + strlen(msg) - 1;

// switches first and last characters, moving inwards towards the character array

**for** (**int** j = 0; j < (strlen(msg)/2); j++) {

**char** temp;

temp = \*last;

\*last = \*first;

\*first = temp;

first++;

last--;

}

}