

CSC-363: Problem Set 5: C++ II

Part I: Pointers (Submit in Lab)

In this problem set, your solutions must be typed. All solution code must be valid (compiles and executes); make sure to include libraries where applicable. If source code is provided, it must be in Courier New (or Consolas) font or a readable screenshot of a reasonable editor. Please submit a PDF if typed. You may submit a written document that includes memory diagrams as your work.

1. Short Answer. Provide a brief answer in the space provided.

Question	Short Answer
How many variables of type “pointer to integer” are declared in each code segment below?	(a)
(a) <code>int* x, y, z;</code> (b) <code>int* a, **b, c;</code>	(b)
What is the only integer that may be assigned to a pointer?	
What is the output of the following code? Briefly explain. <pre>int a = 10; std::cout << *&*&*&*&*&a << std::endl;</pre>	
State the contents of array <code>ary</code> after the following code fragment is executed. Also, explain with a memory diagram. <pre>int *p; int ary[] = { 1, 2, 3, 4, 5, 6, 7 }; p = ary; p++; p[4] = 100;</pre>	
State sample output of the following code. Describe the output with respect to a labeled memory diagram. Explain the equivalence between the variables <code>x0</code> , <code>x1</code> , <code>x2</code> , and <code>x3</code> compared to <code>***x3</code> , <code>**x3</code> , <code>*x3</code> , and <code>x3</code> . <pre>int x0 = 0xBEEFBEEF; int* x1 = &x0; int** x2 = &x1; int*** x3 = &x2; std::cout << std::hex << x0 << std::endl << x1 << " " << *x1 << std::endl << x2 << " " << *x2 << " " << **x2 << std::endl << x3 << " " << *x3 << " " << **x3 << " " << ***x3 << std::endl;</pre>	

In problems 2-3, suppose `ary` is an array of integers initialized to contain

`[7, 5, 3, 1, 2, 4, 6, 8]`.

State the contents of array `ary` after the following code fragment is executed. Explain and show your work.

2.

```
for (int i = 0; i < 8; i++)
{
    *(ary + i) = *ary + i;
}
```

3.

```
for (int i = 7; i >= 0; i--)
{
    *(ary + i) = *ary - i;
}
```

4. State the output of the following code. Explain and show your work.

```
void mystery(int* a, int& b, int*& c)
{
    *a = b;
    a = c;
    *a = 1;
    c = new int(3);
    b = 5;
}

void testIt()
{
    int x = 3;
    int* y = new int(1);
    int* z = new int(5);
    int** w = &y;

    mystery(&x, *z, *w);

    std::cout << x << " " << *y << " " << *z << " " << **w << std::endl;
}
```

5. State the output of the following code and explain each step.

```
int i[] = {21, 4, -17, 45};
int* i_ptr = i;

std::cout << (*i_ptr)++ << std::endl;
std::cout << *i_ptr << std::endl;
std::cout << *i_ptr++ << std::endl;
std::cout << *(i_ptr - 1) << std::endl;
std::cout << *i_ptr << std::endl;

std::cout << ++*i_ptr << std::endl;
std::cout << *++i_ptr << std::endl;
```

Does `*` or (postfix) `++` have higher precedence? If neither, justify why your conclusion with respect to the example.

Part II: Bit Manipulation (Submit via Demonstration by Friday)

Share / demonstrate your solutions with your instructor (in person).

1. What can be said about bit-shifting with a negative shift value (e.g., `0x45 >> -2`). Try some examples in Visual Studio, but also perform a web search.

For the following questions, the rightmost bit is considered to be bit 0. Also Assume good input values for p and n where $p \geq n$.

2. For each expression in the table below, give some examples and explain the purpose of each expression.

Expression	Examples	Explanation
<code>~(~0 << n)</code>		
<code>~(~0 << n) << (p + 1 - n)</code>		
<code>(x >> (p + 1 - n)) & ~(~0 << n)</code> where x is an unsigned value.		

3. Write a function `invert(x, p, n)` that returns x with the n bits that begin at position p inverted (i.e., 1 changed to 0 and vice versa), leaving the other bits unchanged.

4. Write a function `setbits(x, p, n, y)` that returns x with the n bits that begin at position p set to the rightmost n bits of y , leaving the other bits unchanged.