CS 121 Worksheet - Week 11 - Pointers Introduction & Practice - Solutions

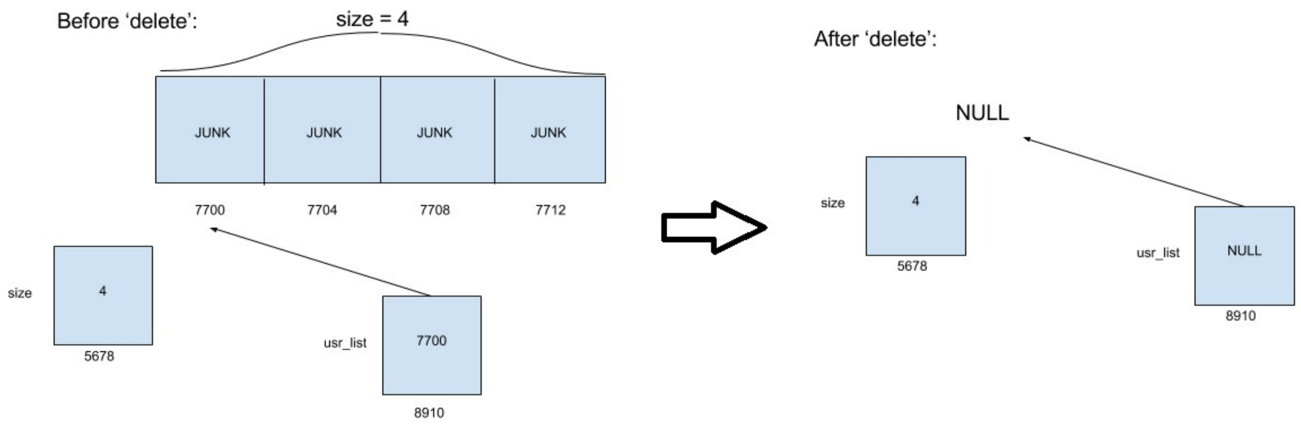
Questions:

1. If you haven't yet, observe and make comments about the diagrams on pages 2 & 3. Are there confusing parts or anything interesting you notice?

Two notes:

- Pointers and arrays have a deep relation (arrays are actually constant pointers)
- The **(dptr+1)* line works just like *dptr[1]*. (which is similar to arr[1] as they point to the same location in the example).

2. Draw out visually what you believe is occurring with the second dynamic memory allocation example. Use the reference diagram on page one to help with your illustration.



3. Translate the following to a pass by-pointer function. The function header for the new function is given to you for convenience.

```
void swap(int& a, int& b)
{
  int temp = a;
  a = b;
  b = temp;
}
```

```
void swap(int* a, int* b)
{
  int temp = *a;  // only use pointers when you need to
  *a = *b;      // set the value a points to to the value b points to
  *b = temp;// set the value located at the address b points to to temp
}
```

4. Translate the following to use pointer arithmetic instead of the bracket operator. Assume all variables below are already defined.

```
for(int i = 0; i < SIZE; ++i)
{
  *(arr + i) = 2 * i;  // example of array arithmetic also shown in the array example (page 
2)
}
```

5. Before we used partially-filled arrays to read data from a file. How could you improve memory efficiency by implementing dynamically-allocated arrays instead? HINT: Think of ways on how we may get the array size.

One way would be to note the number of elements at the beginning of the file, e.g.

Example C++ Syntax	datafile.txt
`//… etc code ...`	5
`ifstream input_file;`	abcdefg
`input_file.open(“datafile.txt”);`	this is another string
	hello everyone
`if(input_file)`	test 2 3 4
`{`	the last string is on this line
`  int num_lines, ix;`	
`  string * str_list;`	
`  input_file >> num_lines;`	
`  input_file.ignore(); // skip the ‘\n’ leftover by “ >> “`	
`  if( num_lines <= 0 )`	
`  {`	
`    cout << “Invalid array size! Exiting program.\n”;`	
`    exit(-1);`	
`  }`	
`  str_list = new string[num_lines]; // new array`	
`  for(ix = 0; ix < num_lines; ++i)`	
`  {`	
`    getline(input_file, str_list[i]);`	
`  }`	
`  input_file.close();`	
`}`	
`//… etc code…`	
`delete str_list;  // free the memory when finished with it`	