Given the following static array declaration, sketch the layout and subsequent contents as it would look in memory:

Value:									
Address:	0x4f83821c	0x4f838220	0x4f838224	0x4f838228	0x4f83822c	0x4f838230	0x4f838234	0x4f838238	0x4f83823c
Identifier:									

```
const unsigned int NUM_SECTIONS = 5;
cout << &NUM_SECTIONS << endl;               // prints 0x4f83823c

int sectionSizes[NUM_SECTIONS] = {0};
cout << sectionSizes << endl;                // prints 0x4f838224

const int SECTION_A = 0, SECTION_B = 1, SECTION_C = 2, SECTION_D = 3, SECTION_E = 4;
sectionSizes[ SECTION_C ] = 60;
sectionSizes[ SECTION_D ] = 61;

cout << sectionSizes[ SECTION_C ] << endl;   // what does this print?
// equation and math to determine location of each element within array?




cout << &sectionSizes[ SECTION_C ] << endl;  // what does this print?

sectionSizes[ SECTION_A ] = sectionSizes[ SECTION_C ];

sectionSizes[ SECTION_B ] = sectionSizes[ SECTION_D ] - 2;
sectionSizes[ SECTION_E ] = sectionSizes[ SECTION_A ]; // runtime to access each element? O(      )

// runtime to print entire array contents? O(      )
for(unsigned int i = 0; i < NUM_SECTIONS; i++) {
  cout << "Section " << (char)('A' + i) << ": " << sectionSizes[i] << endl;
}
```