Homework 32
Examine the following source code and fill in the matrix below:

```cpp
#include <iostream>
using namespace std;

struct myStruct
{
  double x, y;
  int* ptr;
};

int main()
{
  myStruct* bunchOfStructs;
  myStruct singleStruct;
  int variable1, variable2;
  double variable3;
  char variable4;

  bunchOfStructs = new myStruct[3];

  singleStruct.x = 3.6;
  singleStruct.y = 12.7;
  singleStruct.ptr = new int[5];
  for(int i = 0; i < 5; i++)
    singleStruct.ptr[i] = i * 4;

  for(int i = 0; i < 3; i++)
    {
      bunchOfStructs[i].x = i / 2;
      bunchOfStructs[i].y = (i / 2) * -1;
      bunchOfStructs[i].ptr = new int[5];
      for(int j = 0; j < 5; j++)
        bunchOfStructs[i].ptr[j] = (i + j) * 3;
    }
  return 0;
}
```

1) If the value of a variable is an address, simply write ADDR for the value (but still give the type)
2) If the value of a variable is a struct, simply write STRUCT for the value (but still give the type)

**Hint:** Do this by hand, not by running in Visual Studio. You don't need to run/simulate the whole program above to be able to fill out the matrix, just look at the variable and figure out what its value would be. Once a variable is initialized its value is not modified.

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>singleStruct</td>
<td>myStruct</td>
<td></td>
</tr>
<tr>
<td>bunchOfStructs</td>
<td>myStruct[3]</td>
<td></td>
</tr>
<tr>
<td>singleStruct.x</td>
<td>double</td>
<td></td>
</tr>
<tr>
<td>singleStruct.ptr</td>
<td>int*</td>
<td></td>
</tr>
<tr>
<td>bunchOfStructs[1].y</td>
<td>double</td>
<td></td>
</tr>
<tr>
<td>bunchOfStructs[1].ptr</td>
<td>int*</td>
<td></td>
</tr>
<tr>
<td>bunchOfStructs[1]</td>
<td>myStruct</td>
<td></td>
</tr>
</tbody>
</table>