Main.cpp

```
// This program uses subscript notation with a pointer variable
and
// pointer notation with an array name.
#include <iostream>
#include "pointers.h"
using namespace std;

int main()
{
   const int NUM_COINS = 5;
   int coins[NUM_COINS] = {1, 2, 3, 4, 5};
   int *intPtr; // Pointer to a double
   int count;   // Array index

   Pointers p;

   // Display the contents of the coins array. Use subscripts
   // with the pointer!
   p.printValues(coins, intPtr, NUM_COINS);

   // Display the contents of the array again, but this time
   // use pointer notation with the array name!
   p.printAgain(coins, intPtr, NUM_COINS);

   return 0;
}
```

Pointers.h

```
class Pointers
{
private:
public:
    void printValues(int coins[], int *intPtr, int NUM_COINS);
    void printAgain(int coins[], int *intPtr, int NUM_COINS);
};
```

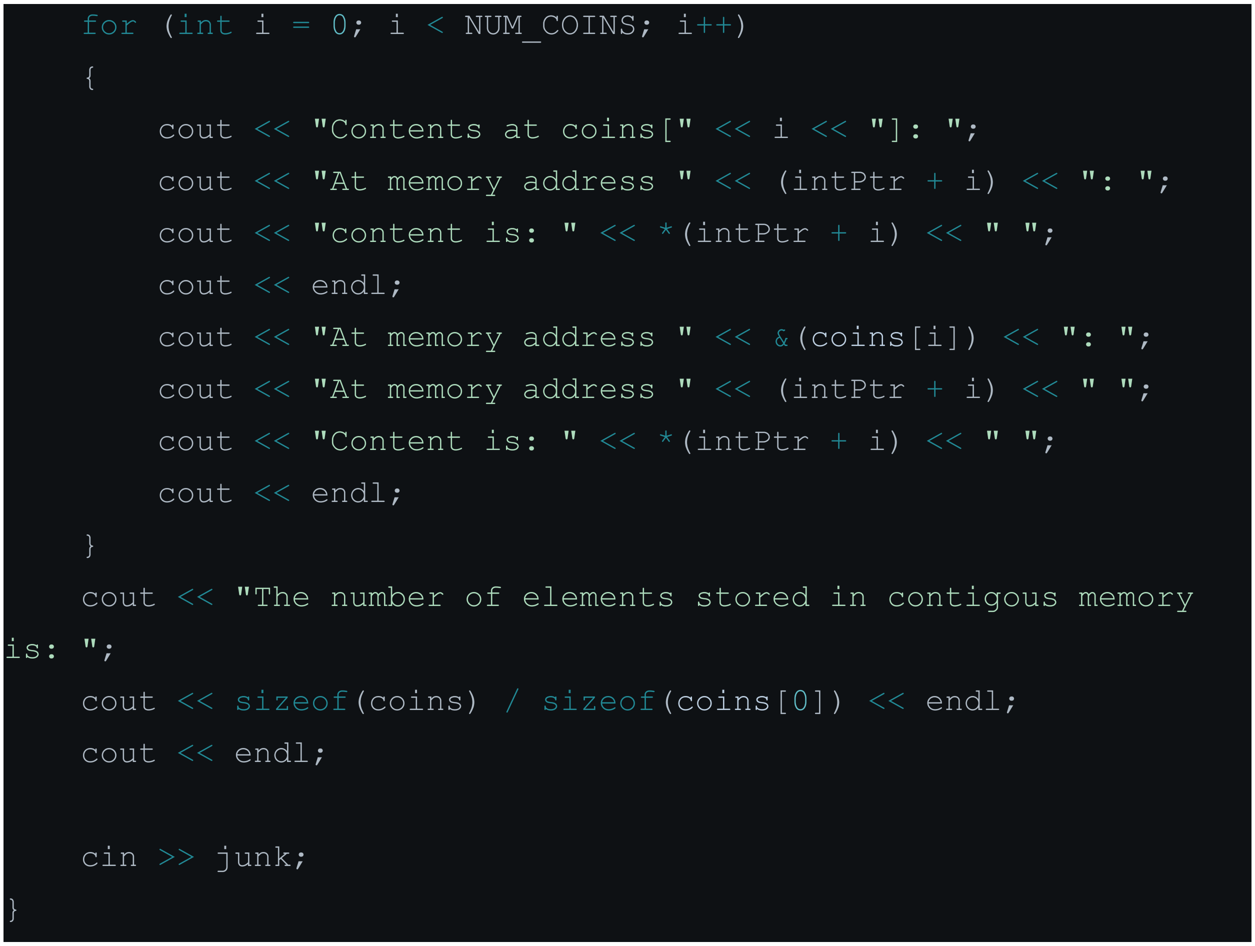
Pointers.cpp

```
#include <iostream>
#include "pointers.h"

using namespace std;

void Pointers::printValues(int coins[], int *intPtr, int
NUM_COINS)
{
    // Assign the address of the coins array to intPtr.
    intPtr = coins;
    cout << "Here are the values in the coins array:\n";
    for (int i = 0; i < NUM_COINS; i++)
    {
        cout << intPtr[i] << " ";
    }
    cout << endl;
    for (int i = 0; i < NUM_COINS; i++)
    {
        cout << coins[i] << " ";
    }
}

void Pointers::printAgain(int coins[], int *intPtr, int
NUM_COINS)
{
    intPtr = coins;
    cout << "\nAnd here they are again:\n";
    char junk;
```



```
    for (int i = 0; i < NUM_COINS; i++)
    {
        cout << "Contents at coins[" << i << "]: ";
        cout << "At memory address " << (intPtr + i) << ": ";
        cout << "content is: " << *(intPtr + i) << " ";
        cout << endl;
        cout << "At memory address " << &(coins[i]) << ": ";
        cout << "At memory address " << (intPtr + i) << " ";
        cout << "Content is: " << *(intPtr + i) << " ";
        cout << endl;
    }
    cout << "The number of elements stored in contigous memory 
is: ";
    cout << sizeof(coins) / sizeof(coins[0]) << endl;
    cout << endl;

    cin >> junk;
}
```

Output: