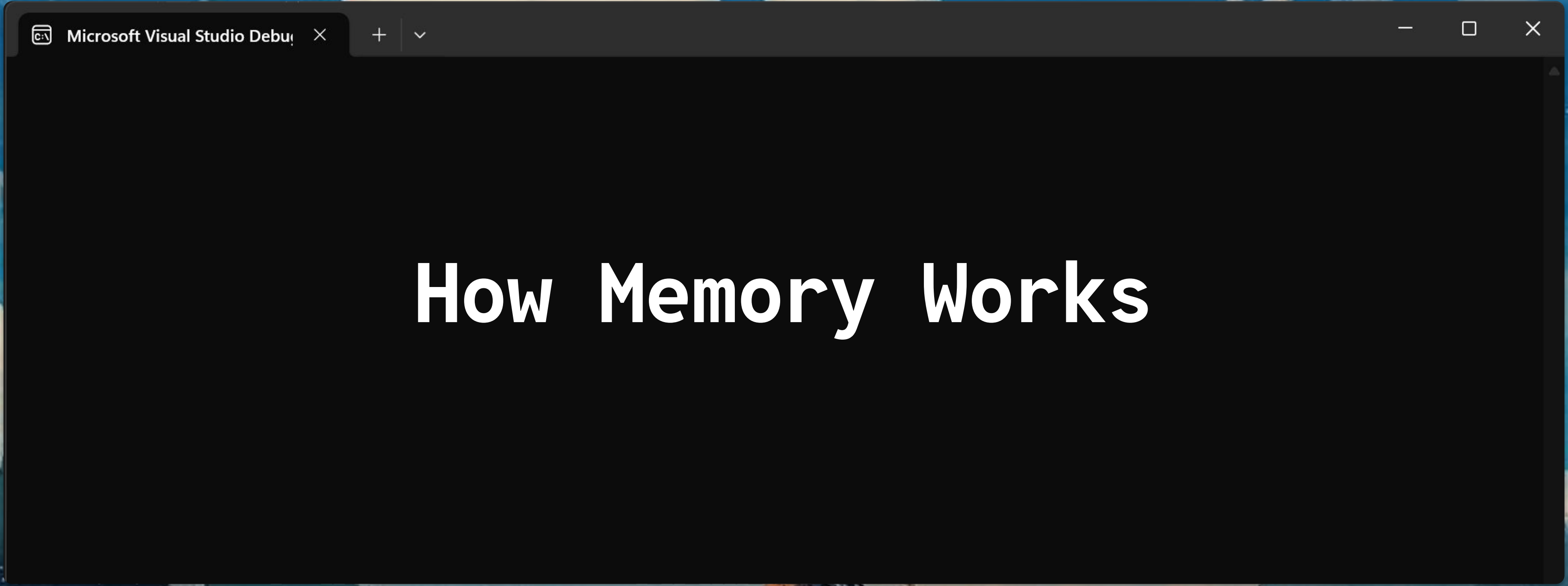
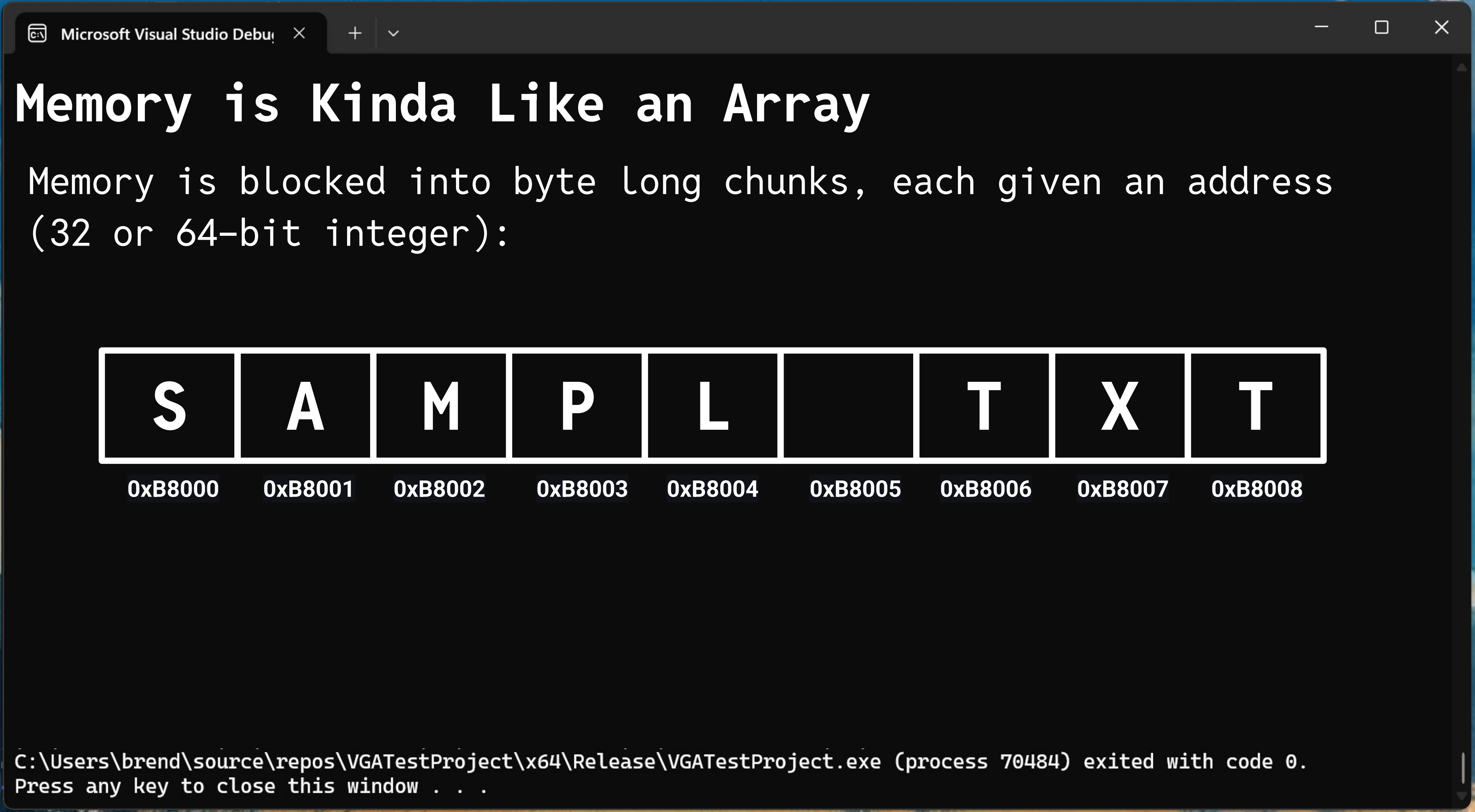
Pointing Fingers

Week 3

```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```



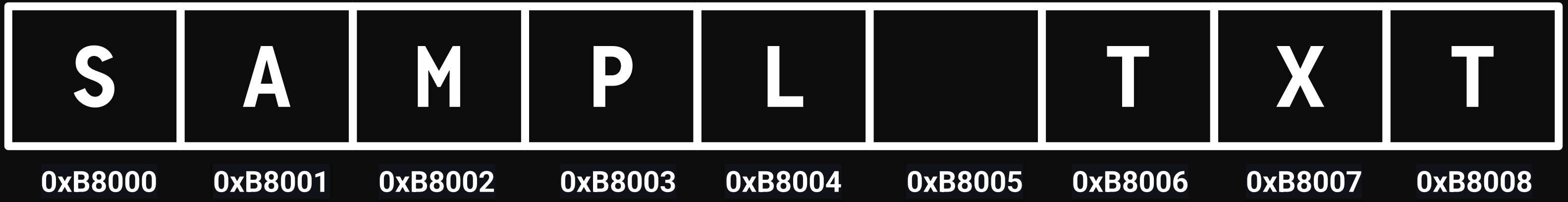
How Memory Works



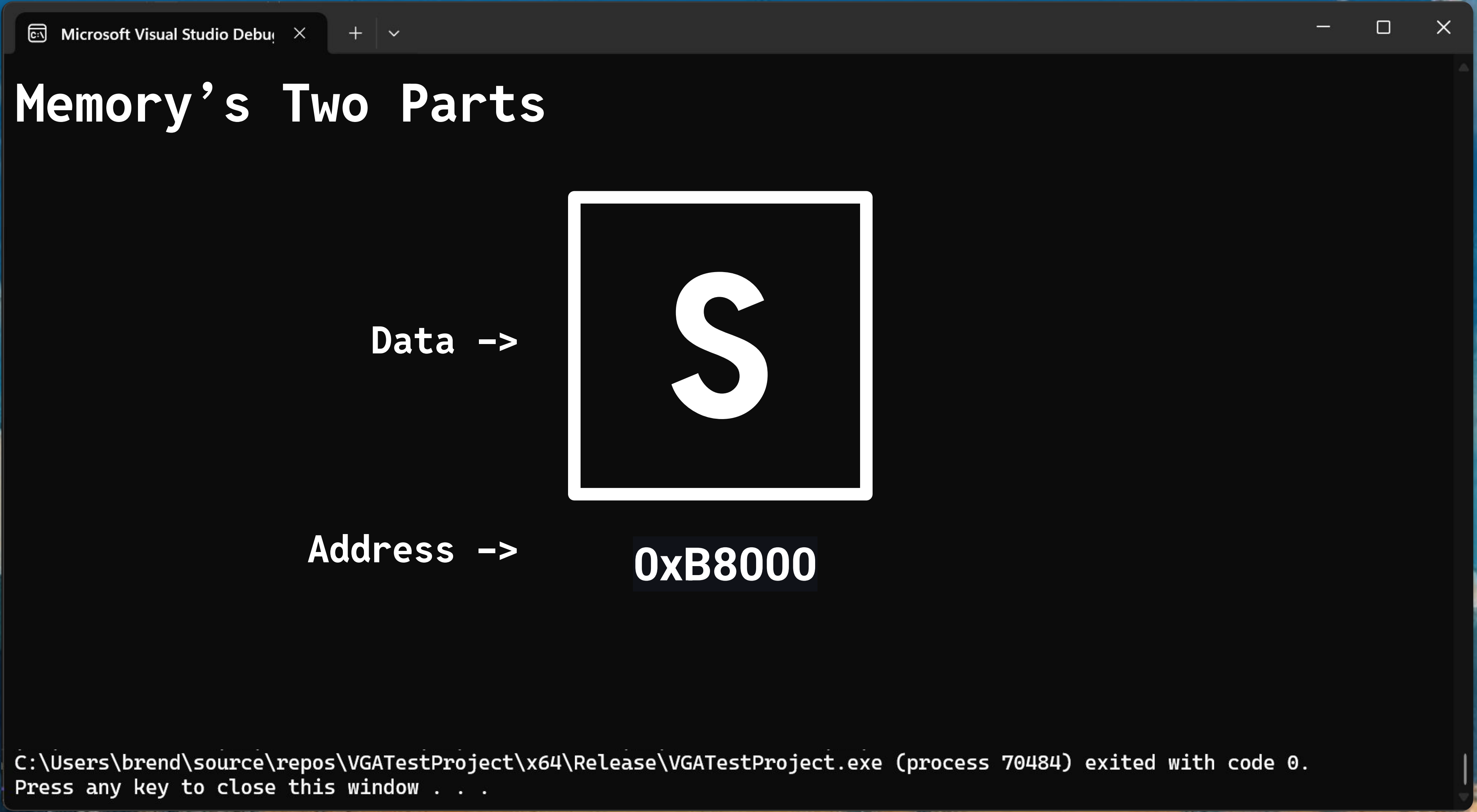
Microsoft Visual Studio Debu

Memory is Kinda Like an Array

Memory is blocked into byte long chunks, each given an address (32 or 64-bit integer):



```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```


Microsoft Visual Studio Debu

Memory’s Two Parts

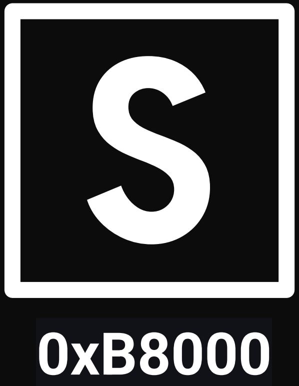
```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```

Pointers

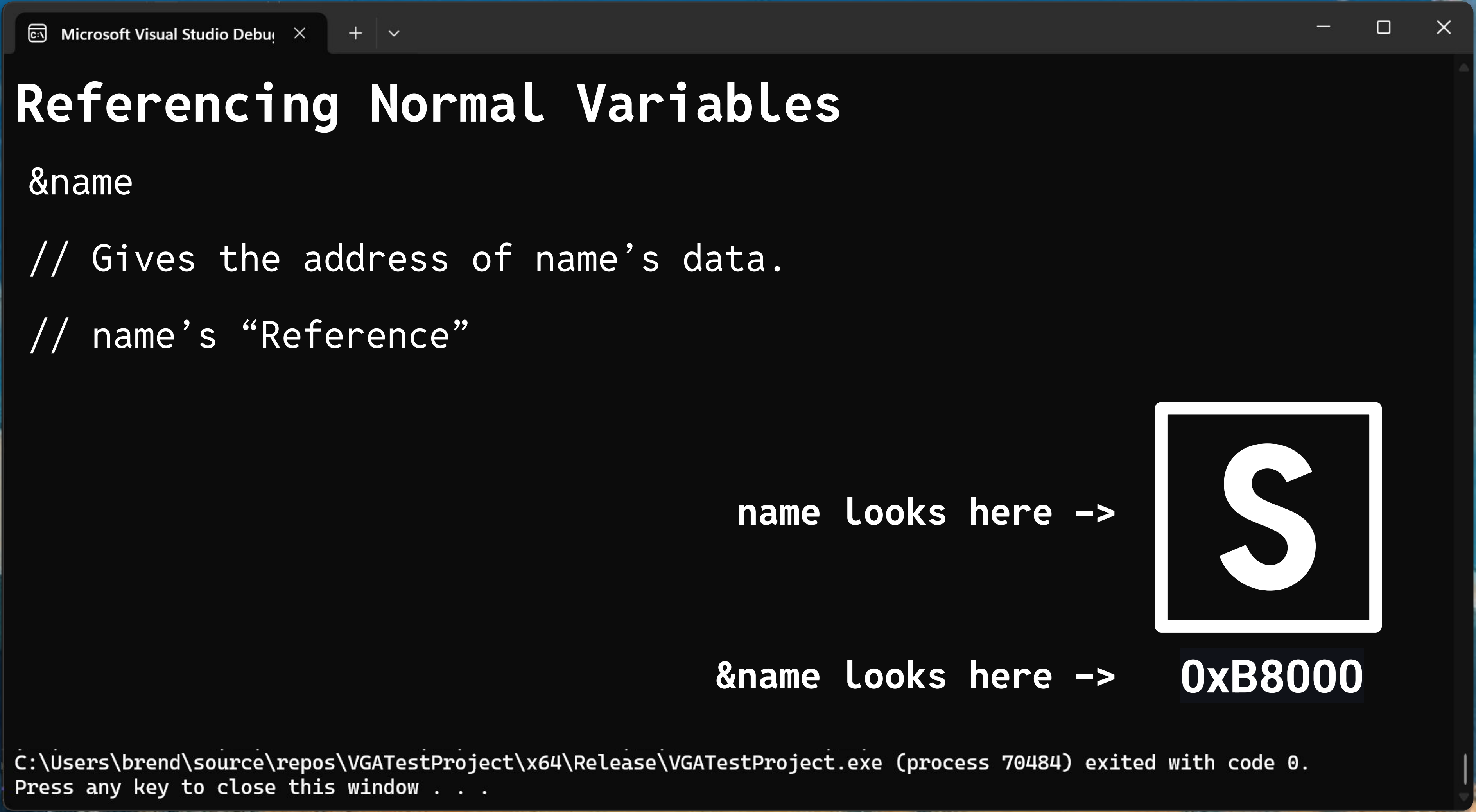
Normal Variables

```
Type name = value;

// Creates a block in memory and represents the actual data
```

name looks here ->



```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```



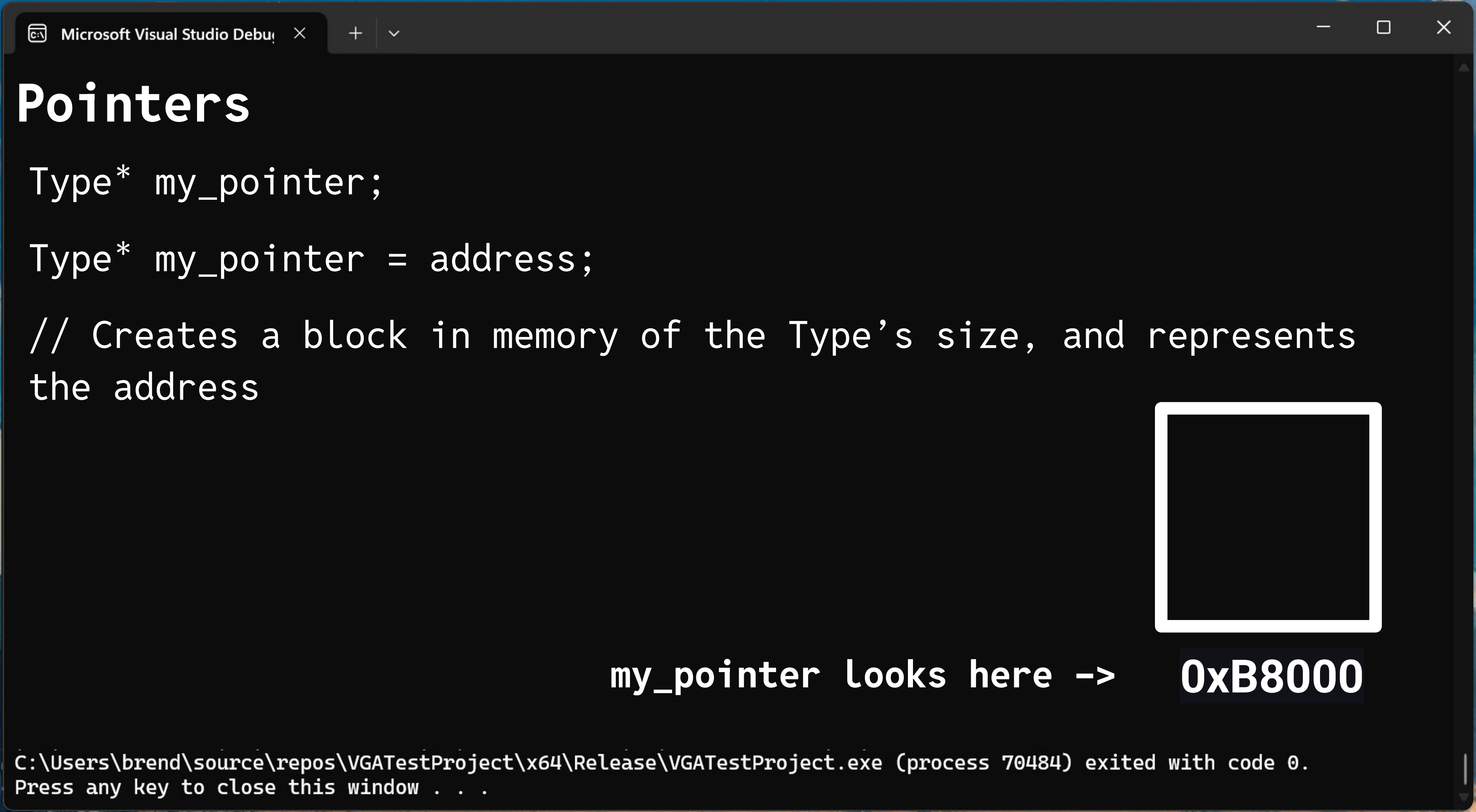
Referencing Normal Variables

```
&name

// Gives the address of name’s data.

// name’s “Reference”
```

name looks here -> S

&name looks here -> 0xB8000

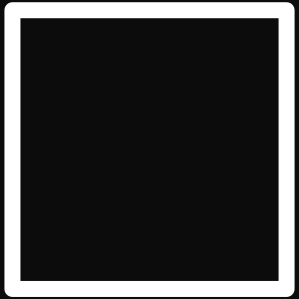
```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```



Microsoft Visual Studio Debu

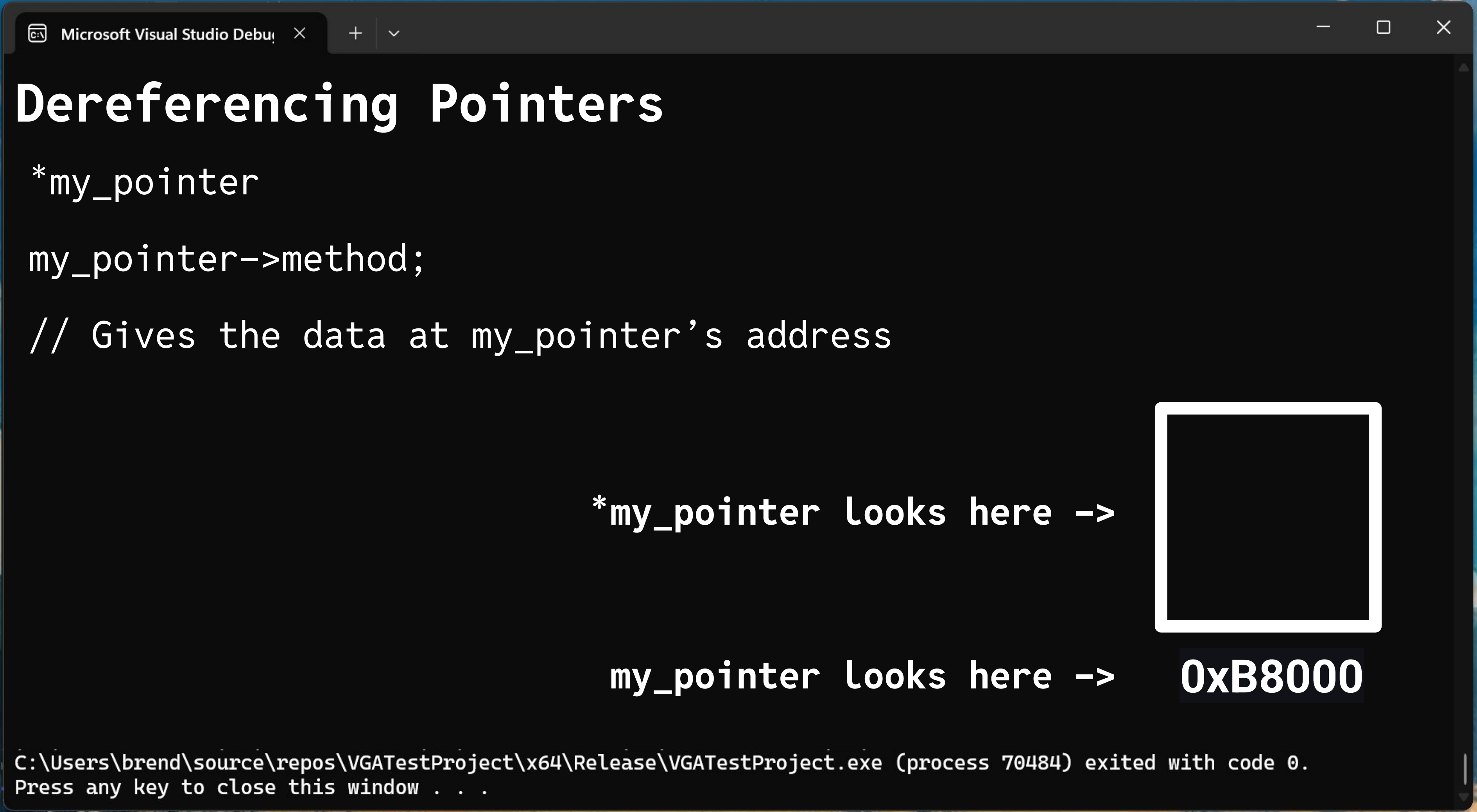
Pointers

```
Type* my_pointer;

Type* my_pointer = address;

// Creates a block in memory of the Type's size, and represents
the address
```



my_pointer looks here -> 0xB8000

```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```

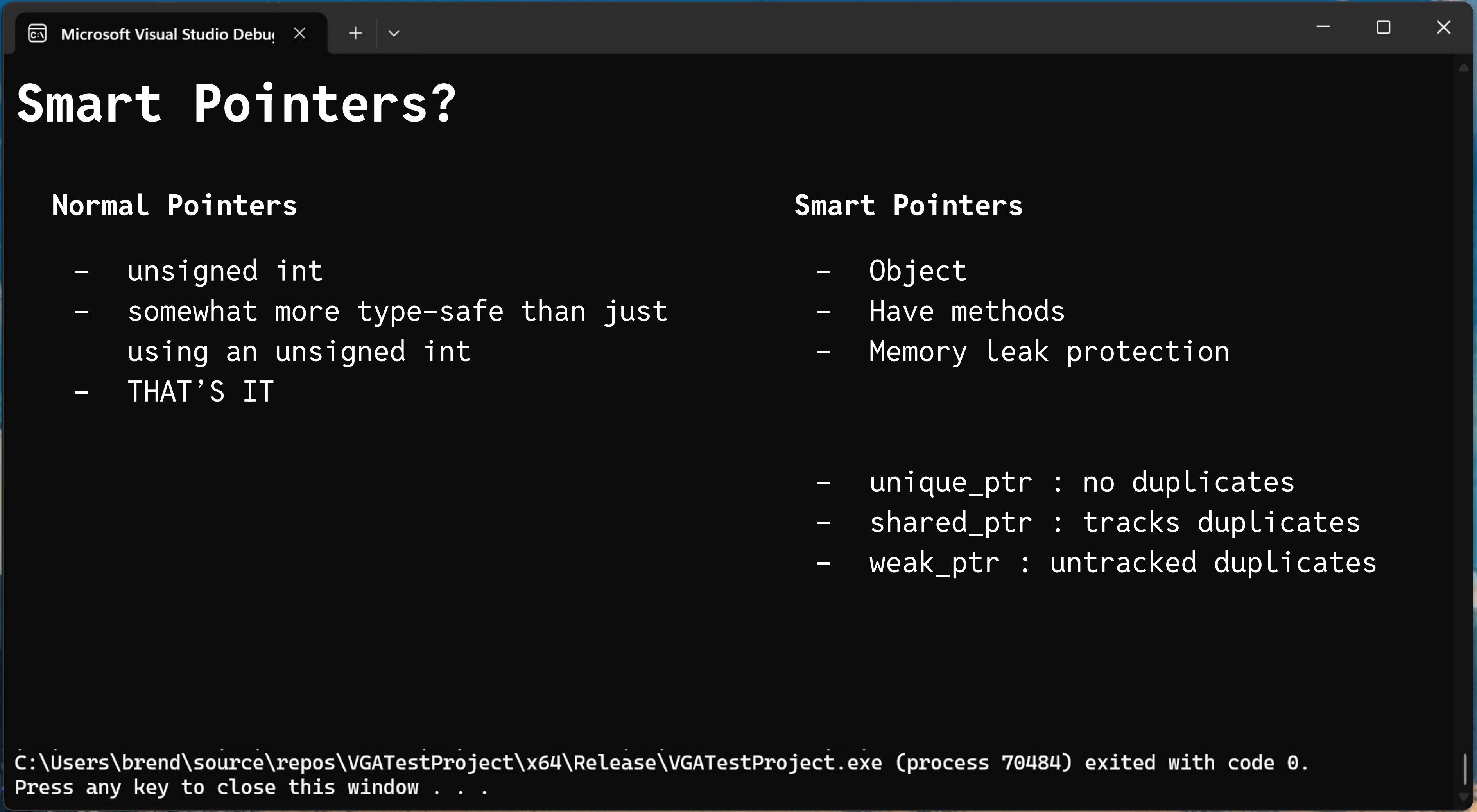


Dereferencing Pointers

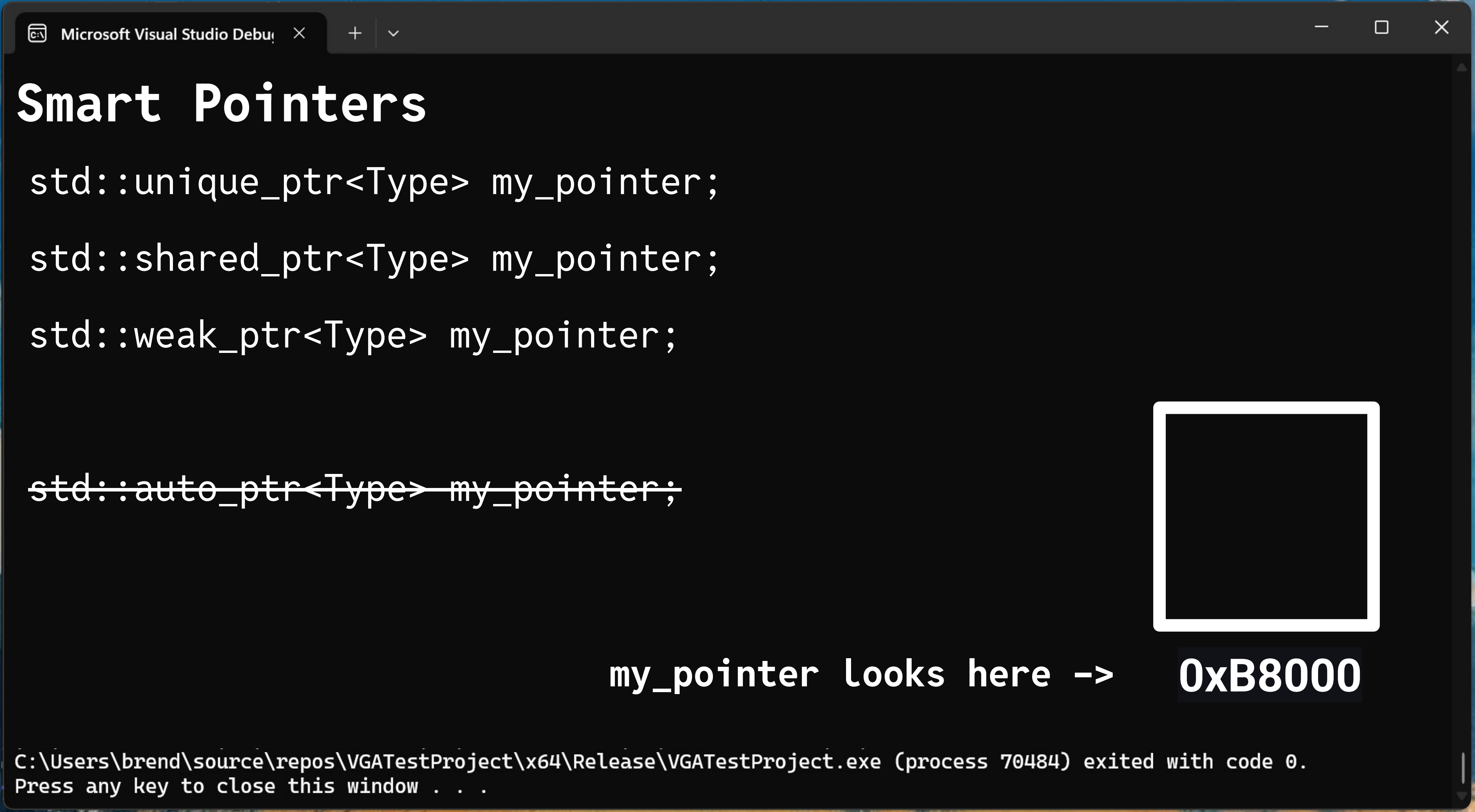
```
*my_pointer

my_pointer->method;

// Gives the data at my_pointer's address
```

```
*my_pointer looks here ->

my_pointer looks here ->   0xB8000
```

```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```



```
Smart Pointers?

  Normal Pointers                              Smart Pointers

   -  unsigned int                              -  Object
   -  somewhat more type-safe than just         -  Have methods
      using an unsigned int                     -  Memory leak protection
   -  THAT'S IT



                                                -  unique_ptr : no duplicates
                                                -  shared_ptr : tracks duplicates
                                                -  weak_ptr : untracked duplicates




C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```



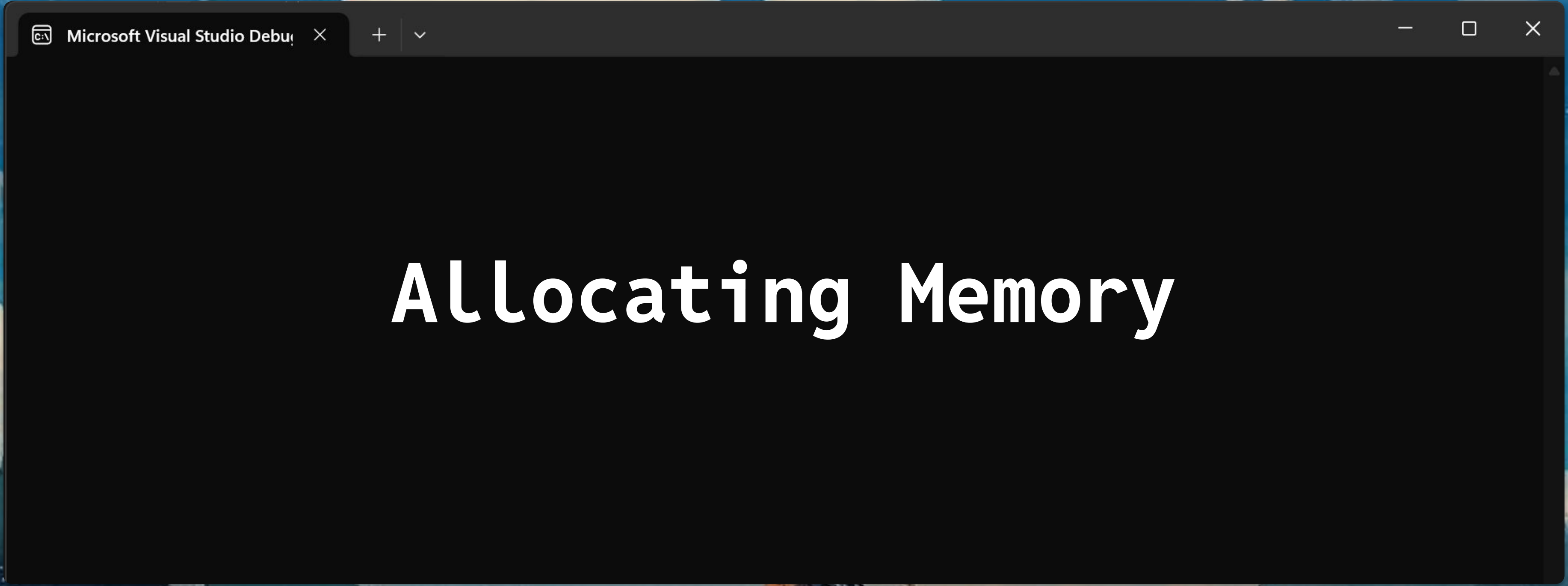
Smart Pointers

```
std::unique_ptr<Type> my_pointer;

std::shared_ptr<Type> my_pointer;

std::weak_ptr<Type> my_pointer;
```

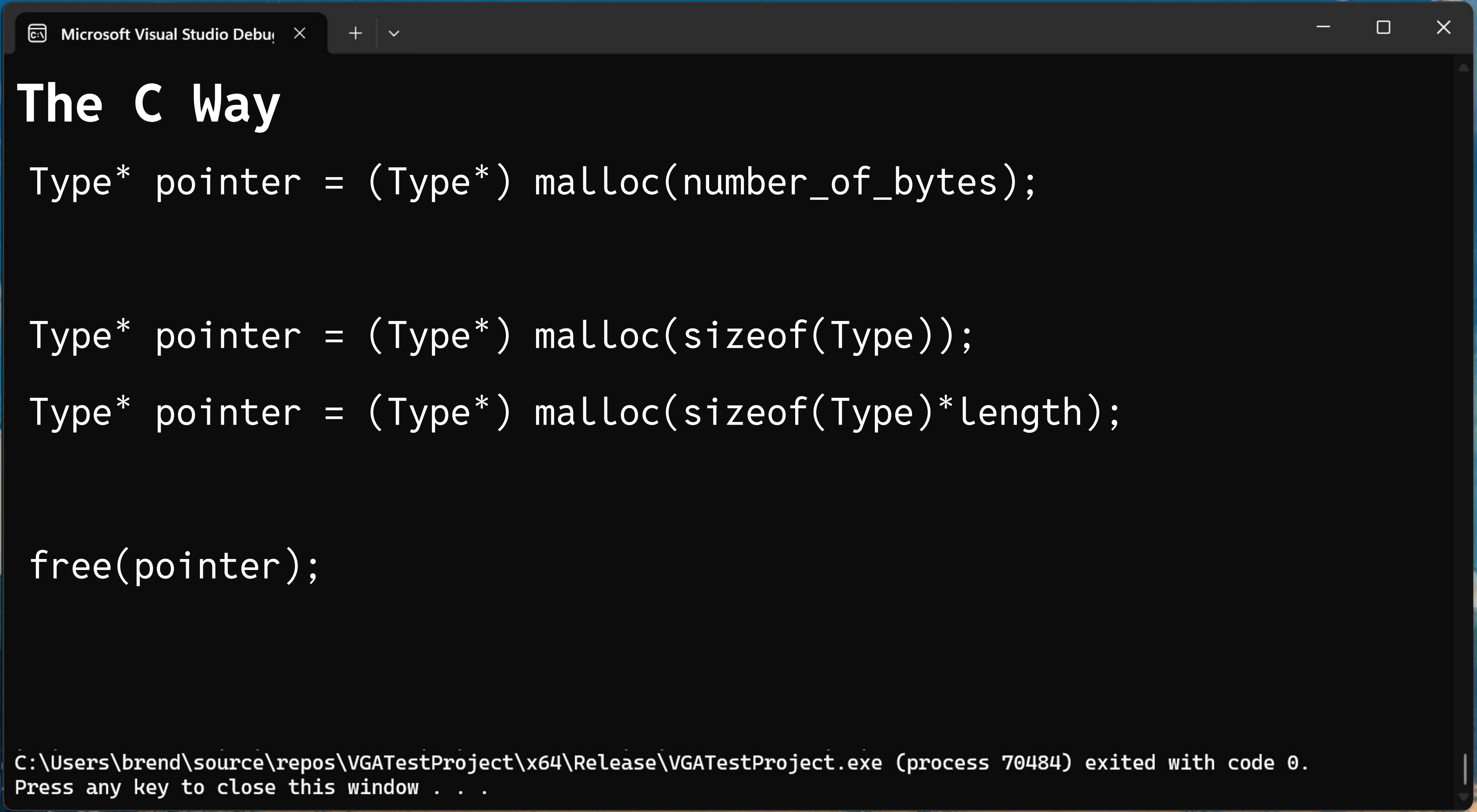
~~std::auto_ptr<Type> my_pointer;~~

my_pointer looks here -> 0xB8000

```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```



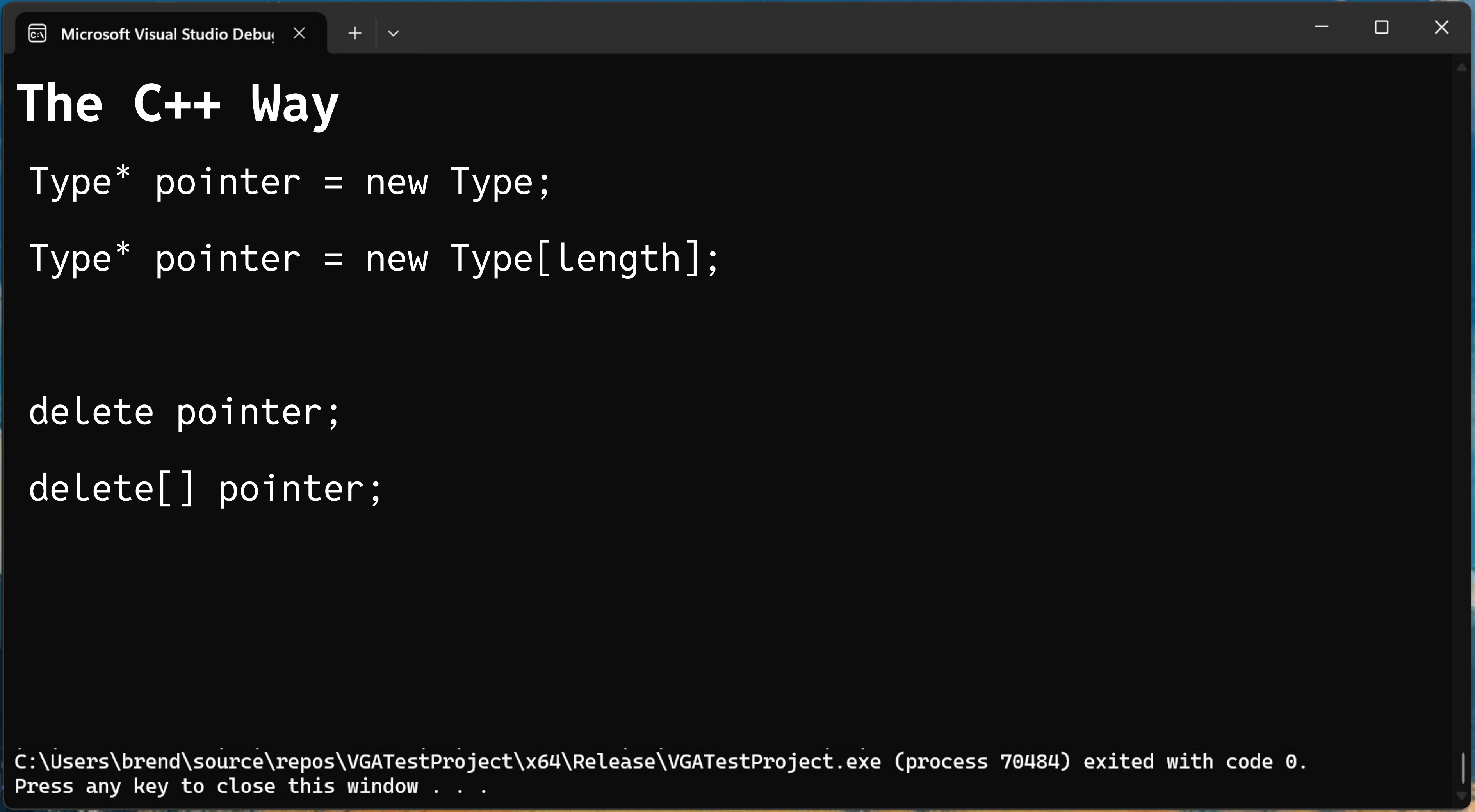
Allocating Memory



The C Way

```
Type* pointer = (Type*) malloc(number_of_bytes);


Type* pointer = (Type*) malloc(sizeof(Type));

Type* pointer = (Type*) malloc(sizeof(Type)*length);


free(pointer);
```

```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```

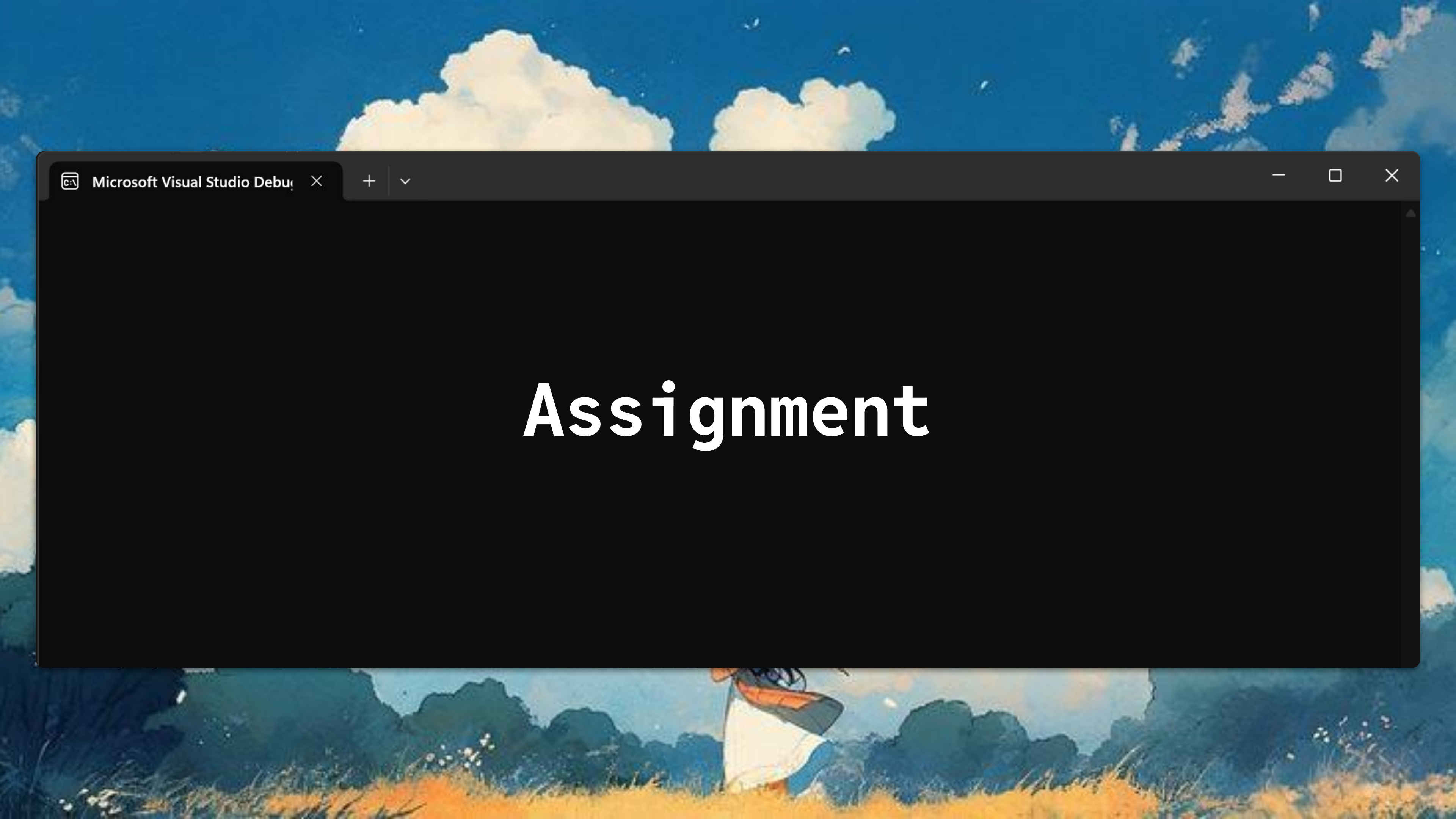


Microsoft Visual Studio Debu

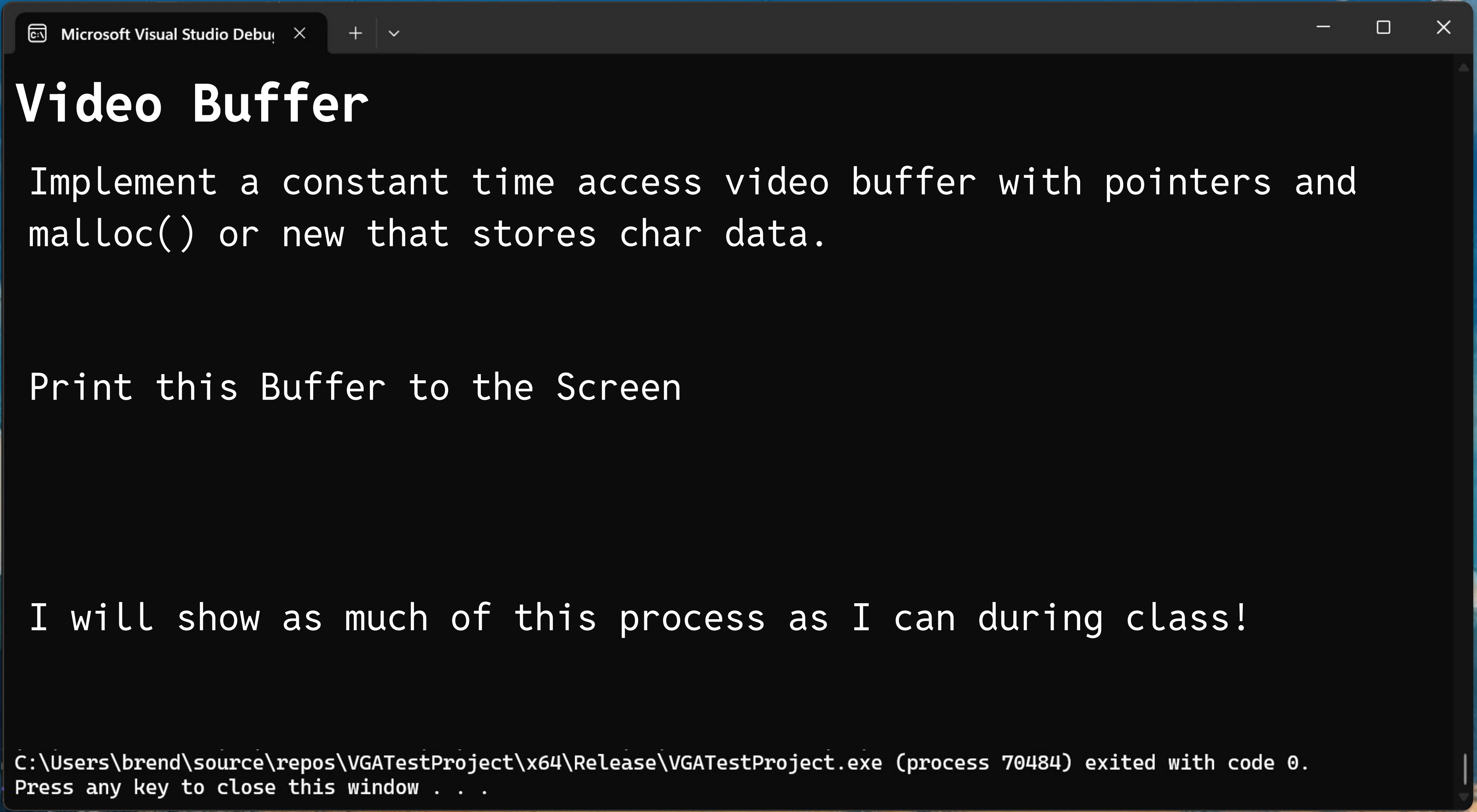
The C++ Way

```
Type* pointer = new Type;

Type* pointer = new Type[length];



delete pointer;

delete[] pointer;
```

```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```



Assignment



Video Buffer

Implement a constant time access video buffer with pointers and malloc() or new that stores char data.

Print this Buffer to the Screen

I will show as much of this process as I can during class!

```
C:\Users\brend\source\repos\VGATestProject\x64\Release\VGATestProject.exe (process 70484) exited with code 0.
Press any key to close this window . . .
```