File
Edit
View
Help

Arrays
Part 03

File Edit View Help

Arrays

- Arrays are a collection of variables of the same type
- Foundational Data Structure
- Contiguous Block of Memory
  - The size of the Array must be specified initially
  - Arrays cannot be resized
- In Java, Arrays are considered a special kind of Object
  - Container Object
  - Identifiers contain only the reference to its contents
  - The reference *points* to contents
  - "==" Does not check the contents of the array

Creating an Array Syntax

```
//Declaring an Array
<<type>>[] <<id>>;
//Initializing an Array]
<<id>> = new <<type>>[<<size>>];
//or
<<type>>[] <<id>> = new <<type>>[size];
```

Example

```
//Creates an array of 5 integers
int[] array = new int[5];
```

File Edit View Help

Multidimensional Arrays

- Arrays may have multiple dimensions
  - More square brackets ("[]") means more dimensions
- Java creates an "Array of Arrays" for multidimensional arrays
  - Arrays are considered container objects
  - The identifier contains a reference to the first array
  - Then Arrays contain memory addresses to other arrays

Creating an 2D Array Syntax

```
//Declaring a 2D Array
<<type>>[][] <<id>> = new <<type>>[<<size01>>][<<size02>>];
```

Example

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
```

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
```

Memory

Identifier	Contents	Byte Address
…	…	…
…	…	…

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
```

Memory

Identifier	Contents	Byte Address
…	…	…
…	…	…

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
```

Memory

Identifier	Contents	Byte Address
…	…	…
array	NULL	28
…	…	…
…	…	…

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
```

Memory

Identifier	Contents	Byte Address
…	…	…
array	NULL	28
…	…	…
array[0]	NULL	60
array[1]	NULL	64
…	…	…
…	…	…

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
```

Memory

Identifier	Contents	Byte Address
…	…	…
array	NULL	28
…	…	…
array[0]	NULL	60
array[1]	NULL	64
…	…	…
…	…	…

More Memory

Identifier	Contents	Byte Address
…	…	…
array[0][0]	0	100
array[0][1]	0	104
array[0][2]	0	108
…		
array[1][0]	0	150
array[1][1]	0	154
array[1][2]	0	158
…	…	…

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
```

Memory

Identifier	Contents	Byte Address
…	…	…
array	NULL	28
…	…	…
array[0]	100	60
array[1]	150	64
…	…	…
…	…	…

More Memory

Identifier	Contents	Byte Address
…	…	…
array[0][0]	0	100
array[0][1]	0	104
array[0][2]	0	108
…		
array[1][0]	0	150
array[1][1]	0	154
array[1][2]	0	158
…	…	…

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
```

Memory

Identifier	Contents	Byte Address
…	…	…
array	NULL	28
…	…	…
array[0]	100	60
array[1]	150	64
…	…	…
…	…	…

More Memory

Identifier	Contents	Byte Address
…	…	…
array[0][0]	0	100
array[0][1]	0	104
array[0][2]	0	108
…		
array[1][0]	0	150
array[1][1]	0	154
array[1][2]	0	158
…	…	…

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
```

Memory

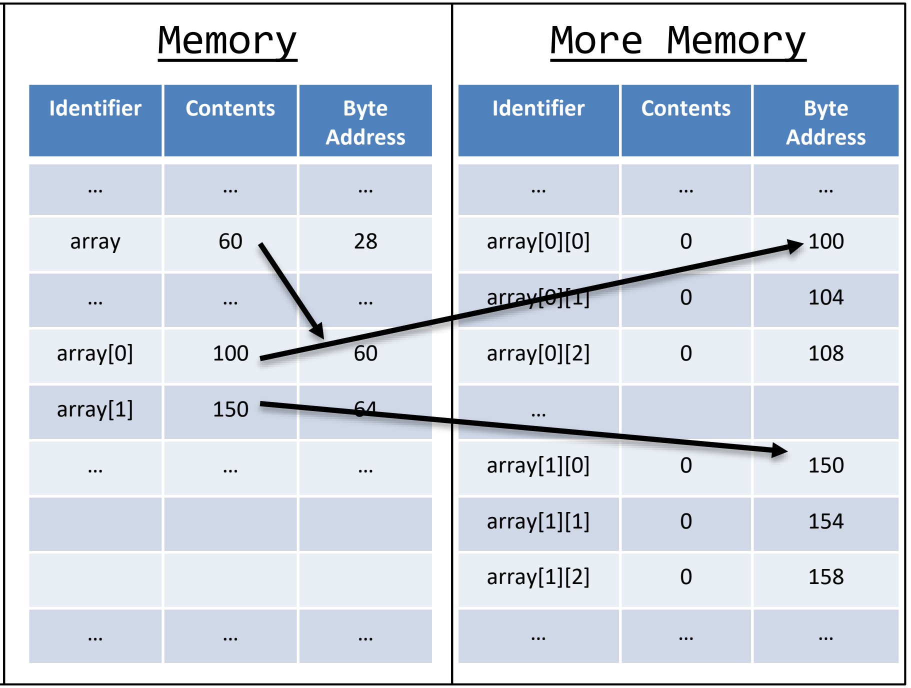
Identifier	Contents	Byte Address
…	…	…
array	60	28
…	…	…
array[0]	100	60
array[1]	150	64
…	…	…
…	…	…

More Memory

Identifier	Contents	Byte Address
…	…	…
array[0][0]	0	100
array[0][1]	0	104
array[0][2]	0	108
…		
array[1][0]	0	150
array[1][1]	0	154
array[1][2]	0	158
…	…	…

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
```



Memory

Identifier	Contents	Byte Address
…	…	…
array	60	28
…	…	…
array[0]	100	60
array[1]	150	64
…	…	…
…	…	…

More Memory

Identifier	Contents	Byte Address
…	…	…
array[0][0]	0	100
array[0][1]	0	104
array[0][2]	0	108
…		
array[1][0]	0	150
array[1][1]	0	154
array[1][2]	0	158
…	…	…

File Edit View Help

Multidimensional Arrays

- Indices still work the same way
  - Indices start at 0
  - Indices End at Size-1 (or Length-1)
  - Need an index for each dimension
- The size of each dimension can be access through the property ".length"
- Nested For-Loops are the multidimensional arrays "best friend"
  - Counting variables can be used for indexing
  - Using the property ".length" can be used in the Boolean expression

Indexing Syntax

```
//Accessing Data
<<id>>[<<index01>>][<<index02>>];
//Modifying Data
<<id>>[<<index01>>][<<index02>>] = <<value>>;
//Using .length property
<<id>>.length;//Outside dimension
<<id>>[<<index>>].length;//Inside dimension
```

Example

```
//Assigning some values
array[0][0] = 1;
array[1][1] = 5;
//Accesses and adds the assigned values
int added = array[0][0] + array[1][1];
```

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
      for(int j=0;j<array[i].length;j++)
      {
            array[i][j] = i+j;
      }
}
```

Memory

Identifier	Contents	Byte Address
…	…	…
array	60	28
…	…	…
array[0]	100	60
array[1]	150	64
…	…	…
…	…	…

More Memory

Identifier	Contents	Byte Address
…	…	…
array[0][0]	0	100
array[0][1]	0	104
array[0][2]	0	108
…		
array[1][0]	0	150
array[1][1]	0	154
array[1][2]	0	158
…	…	…

Multidimensional Arrays

```java
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
	for(int j=0;j<array[i].length;j++)
	{
		array[i][j] = i+j;
	}
}
```

Another Perspective

i \ j	0	1	2
0			
1			

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
     for(int j=0;j<array[i].length;j++)
     {
          array[i][j] = i+j;
     }
}
```

Another Perspective

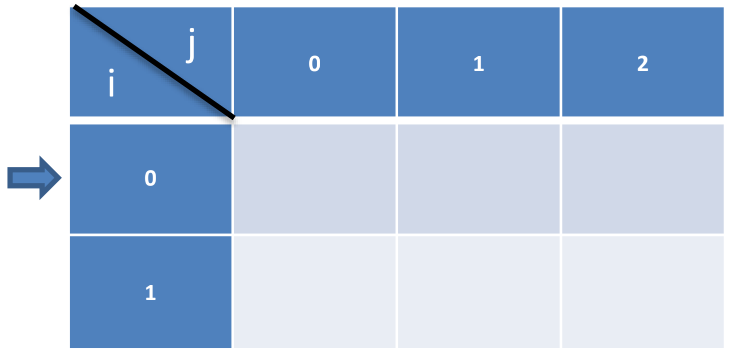
i \ j	0	1	2
0			
1			

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
     for(int j=0;j<array[i].length;j++)
     {
          array[i][j] = i+j;
     }
}
```

Another Perspective



Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
     for(int j=0;j<array[i].length;j++)
     {
          array[i][j] = i+j;
     }
}
```

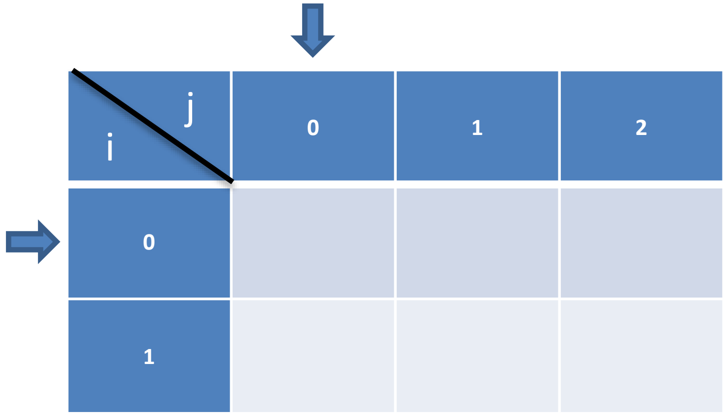
Another Perspective

i \ j	0	1	2
0			
1			

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
        array[i][j] = i+j;
    }
}
```

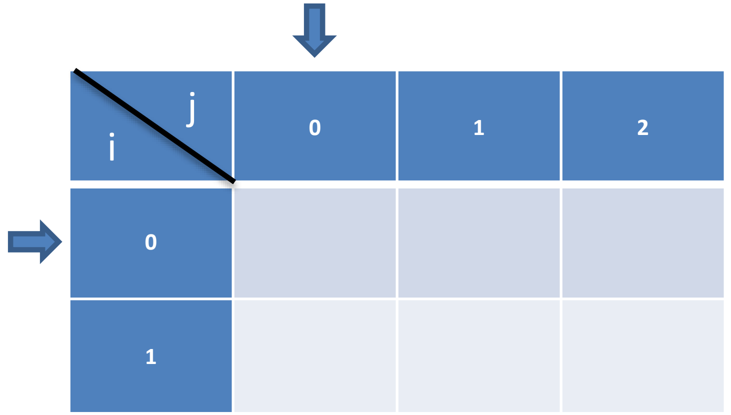
Another Perspective



Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
            array[i][j] = i+j;
    }
}
```

Another Perspective

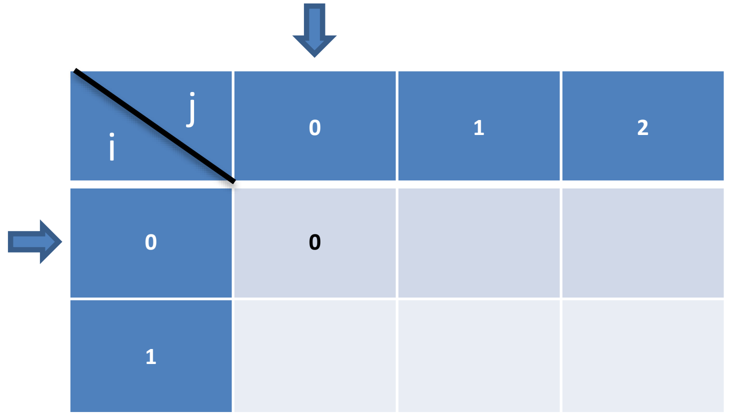


File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
        array[i][j] = i+j;
    }
}
```

Another Perspective

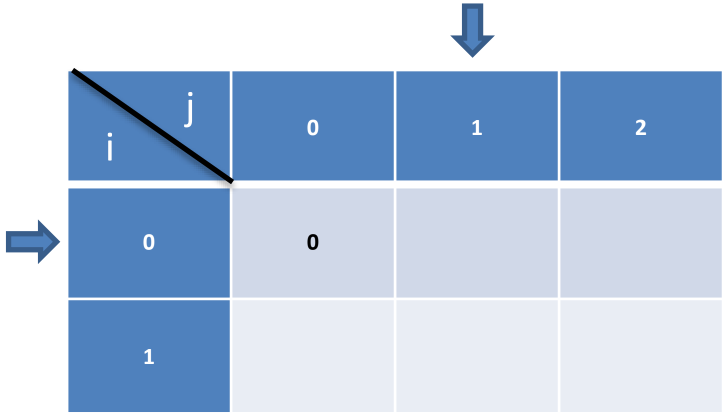


i \ j	0	1	2
0	0		
1			

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
        array[i][j] = i+j;
    }
}
```

Another Perspective

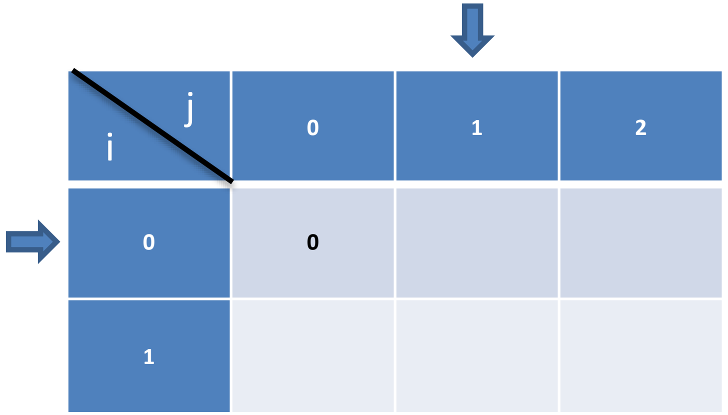


i \ j	0	1	2
0	0		
1			

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
        array[i][j] = i+j;
    }
}
```

Another Perspective

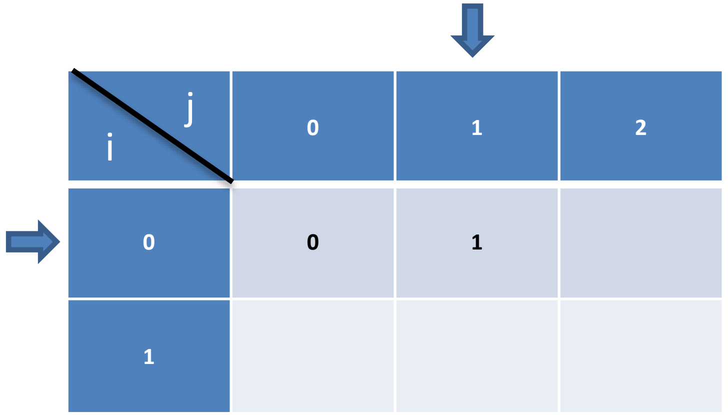


i \ j	0	1	2
0	0		
1			

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
        array[i][j] = i+j;
    }
}
```

Another Perspective



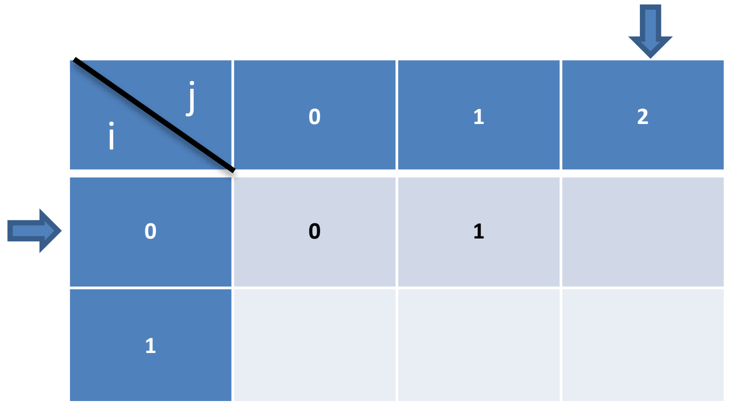
i \ j	0	1	2
0	0	1	
1			

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
            array[i][j] = i+j;
    }
}
```

Another Perspective

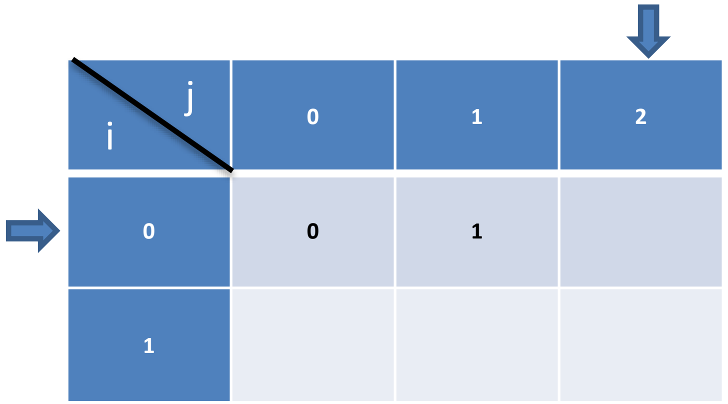


i \ j	0	1	2
0	0	1	
1			

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
        array[i][j] = i+j;
    }
}
```

Another Perspective



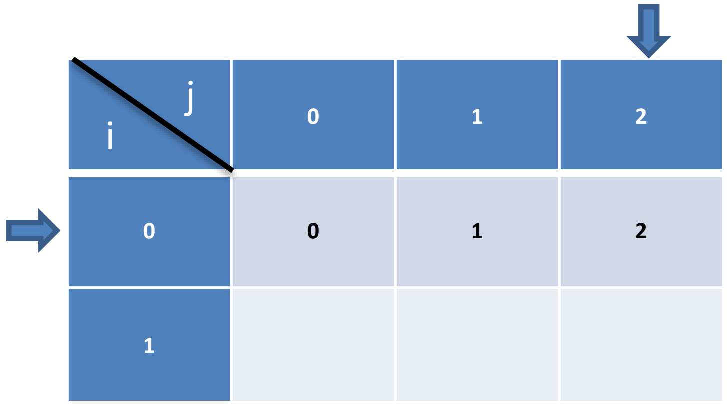
$$\begin{array}{c|ccc} i \backslash j & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & \\ 1 & & & \end{array}$$

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
            array[i][j] = i+j;
    }
}
```

Another Perspective

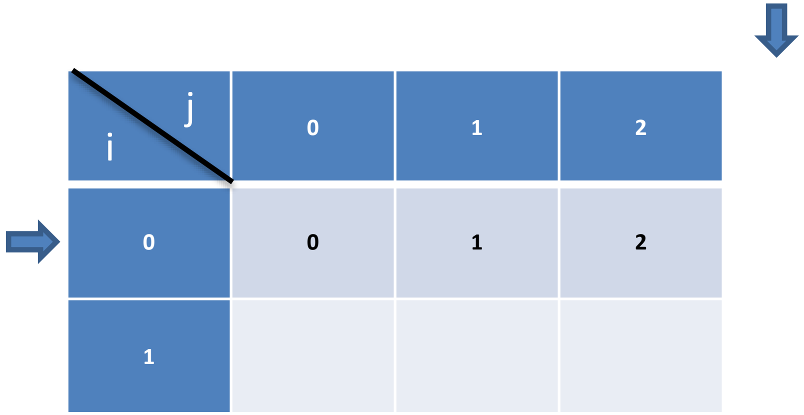


i \ j	0	1	2
0	0	1	2
1			

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
            array[i][j] = i+j;
    }
}
```

Another Perspective



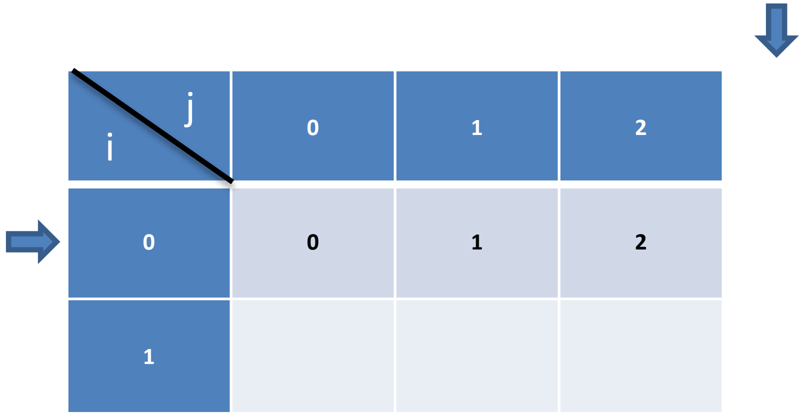
i \ j	0	1	2
0	0	1	2
1			

File Edit View Help

Multidimensional Arrays

```java
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
        array[i][j] = i+j;
    }
}
```

Another Perspective



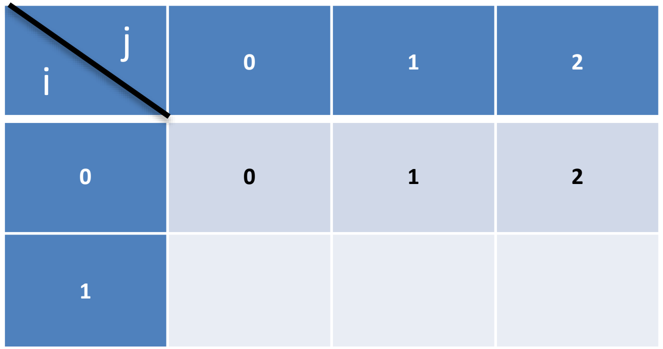
i \ j	0	1	2
0	0	1	2
1			

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
    for(int j=0;j<array[i].length;j++)
    {
        array[i][j] = i+j;
    }
}
```

Another Perspective



i \ j	0	1	2
0	0	1	2
1			

File Edit View Help

A Few Steps Later

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
      for(int j=0;j<array[i].length;j++)
      {
            array[i][j] = i+j;
      }
}
```

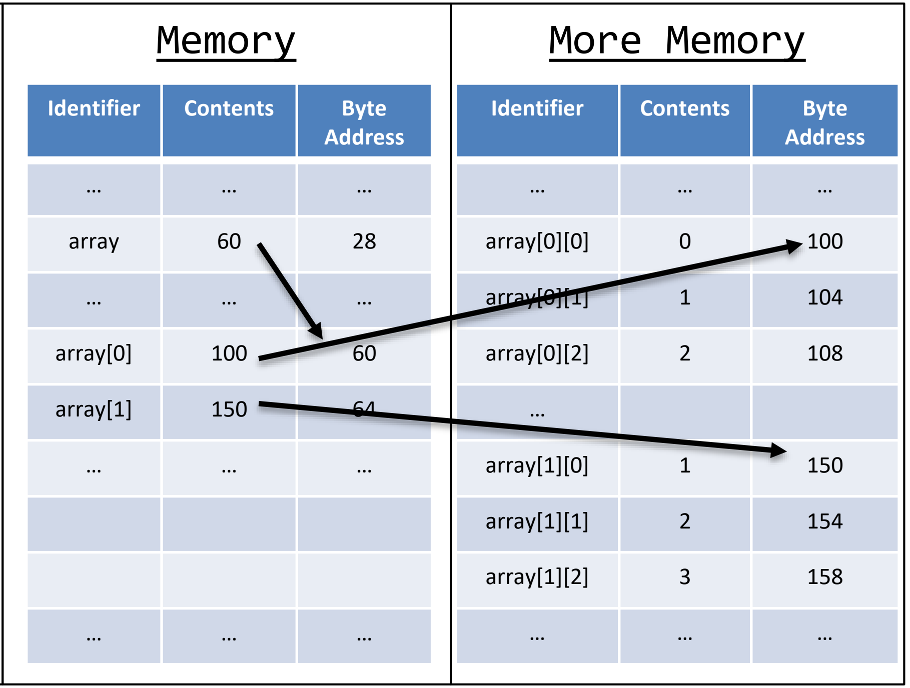
Another Perspective

i \ j	0	1	2
0	0	1	2
1	1	2	3

File Edit View Help

Multidimensional Arrays

```
//Creates a 2 x 3 2D Array of integers
int[][] array = new int[2][3];
for(int i=0;i<array.length;i++)
{
      for(int j=0;j<array[i].length;j++)
      {
            array[i][j] = i+j;
      }
}
```

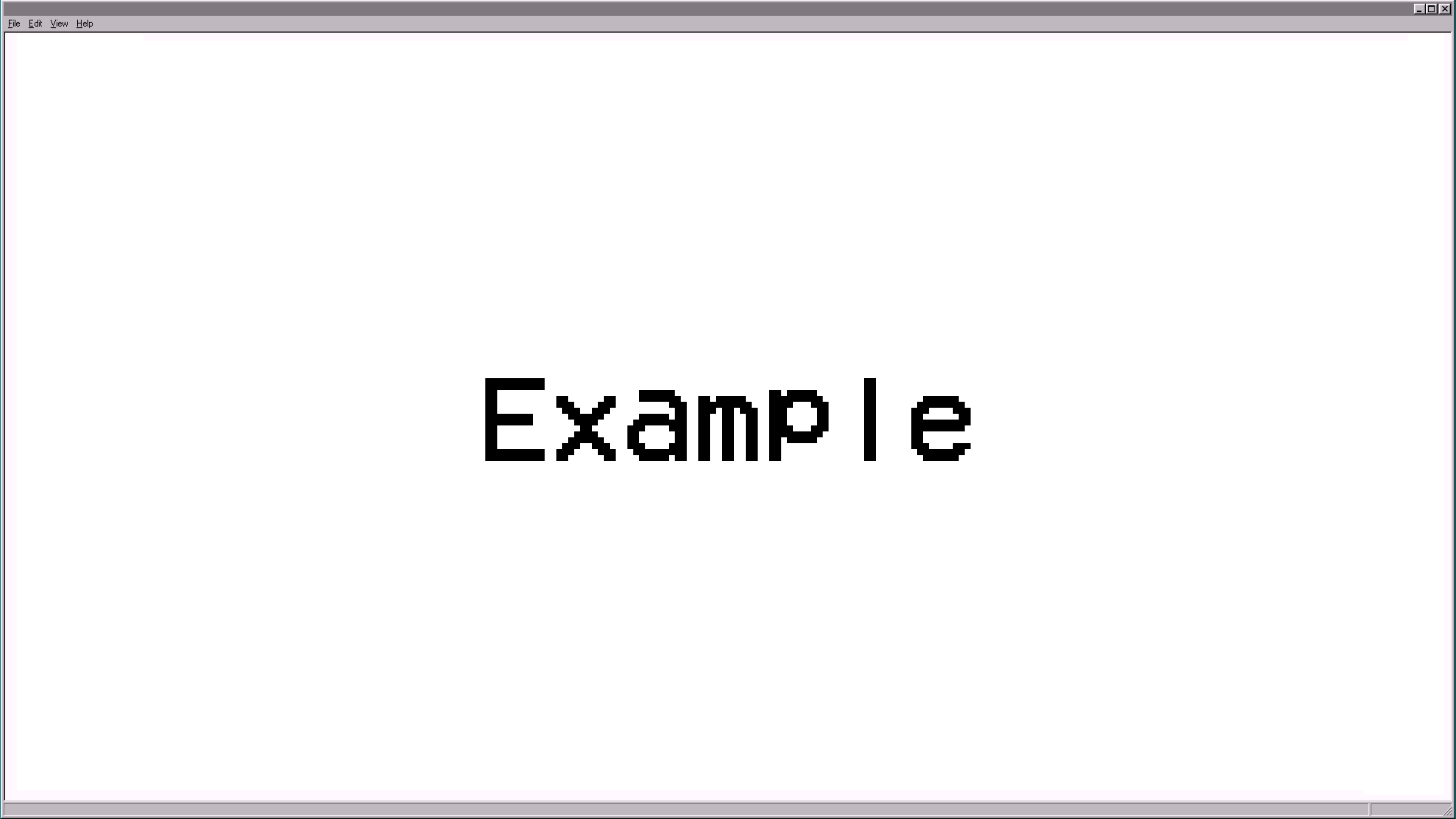


Memory

Identifier	Contents	Byte Address
…	…	…
array	60	28
…	…	…
array[0]	100	60
array[1]	150	64
…	…	…
…	…	…

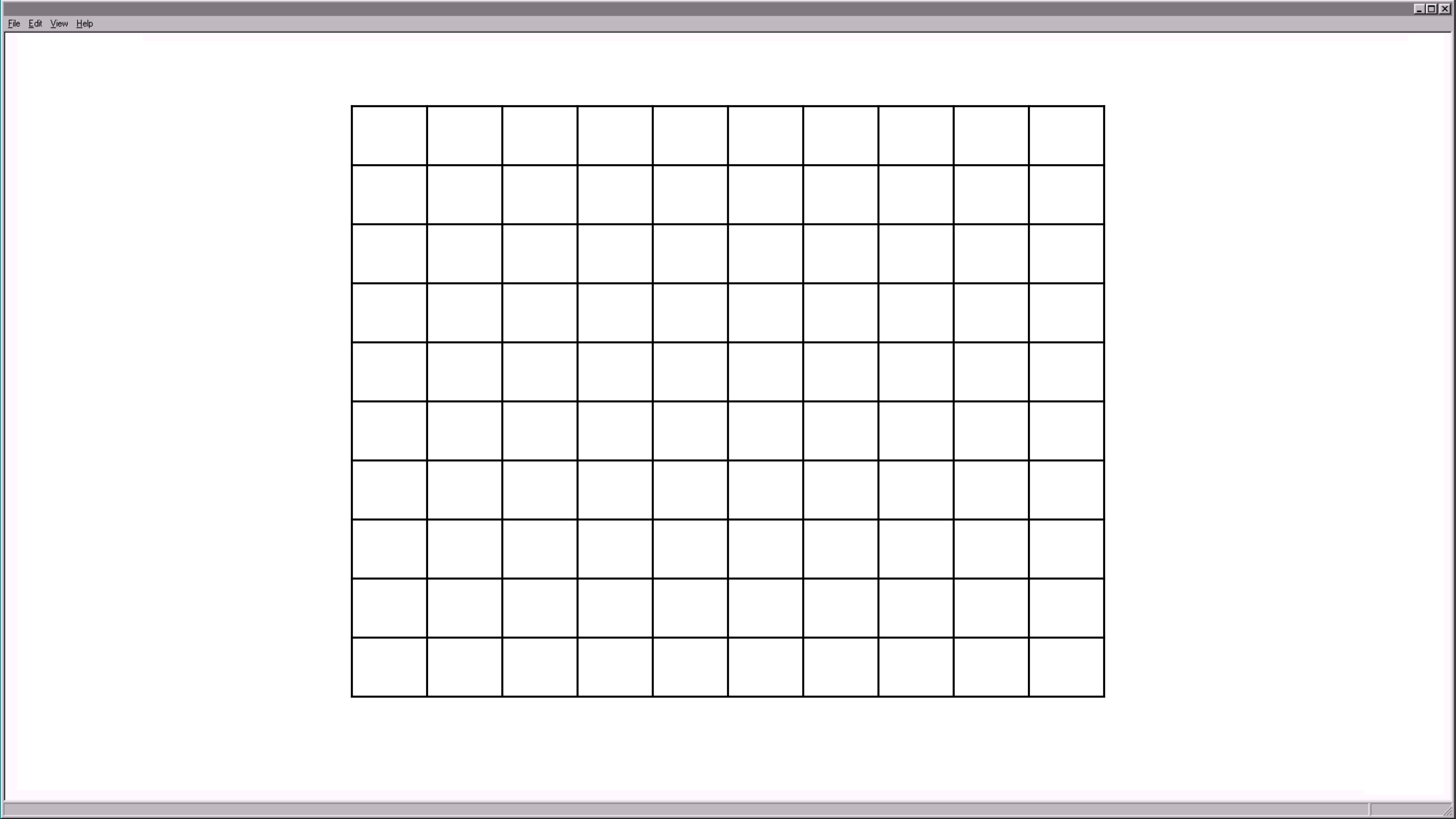
More Memory

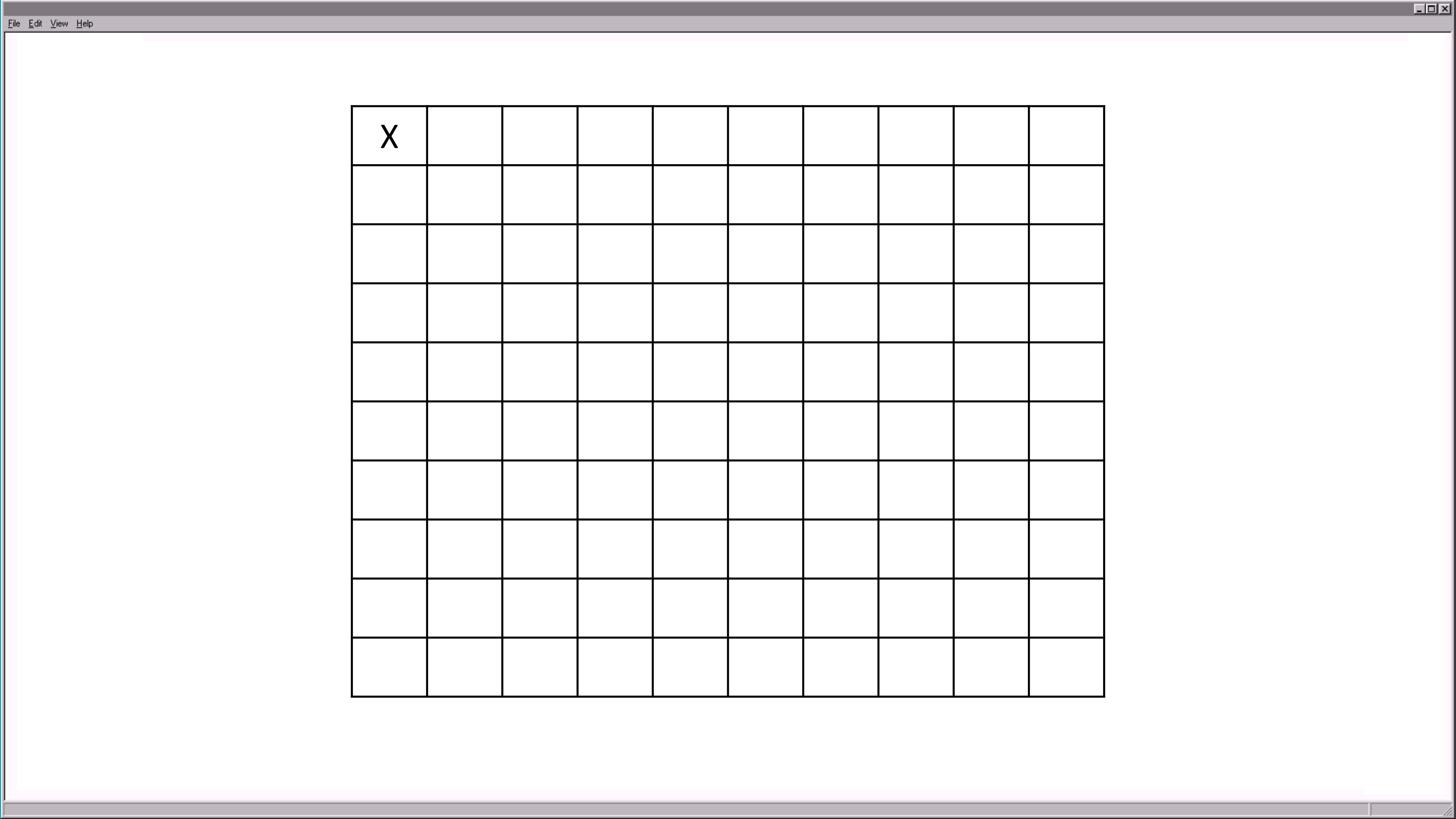
Identifier	Contents	Byte Address
…	…	…
array[0][0]	0	100
array[0][1]	1	104
array[0][2]	2	108
…		
array[1][0]	1	150
array[1][1]	2	154
array[1][2]	3	158
…	…	…



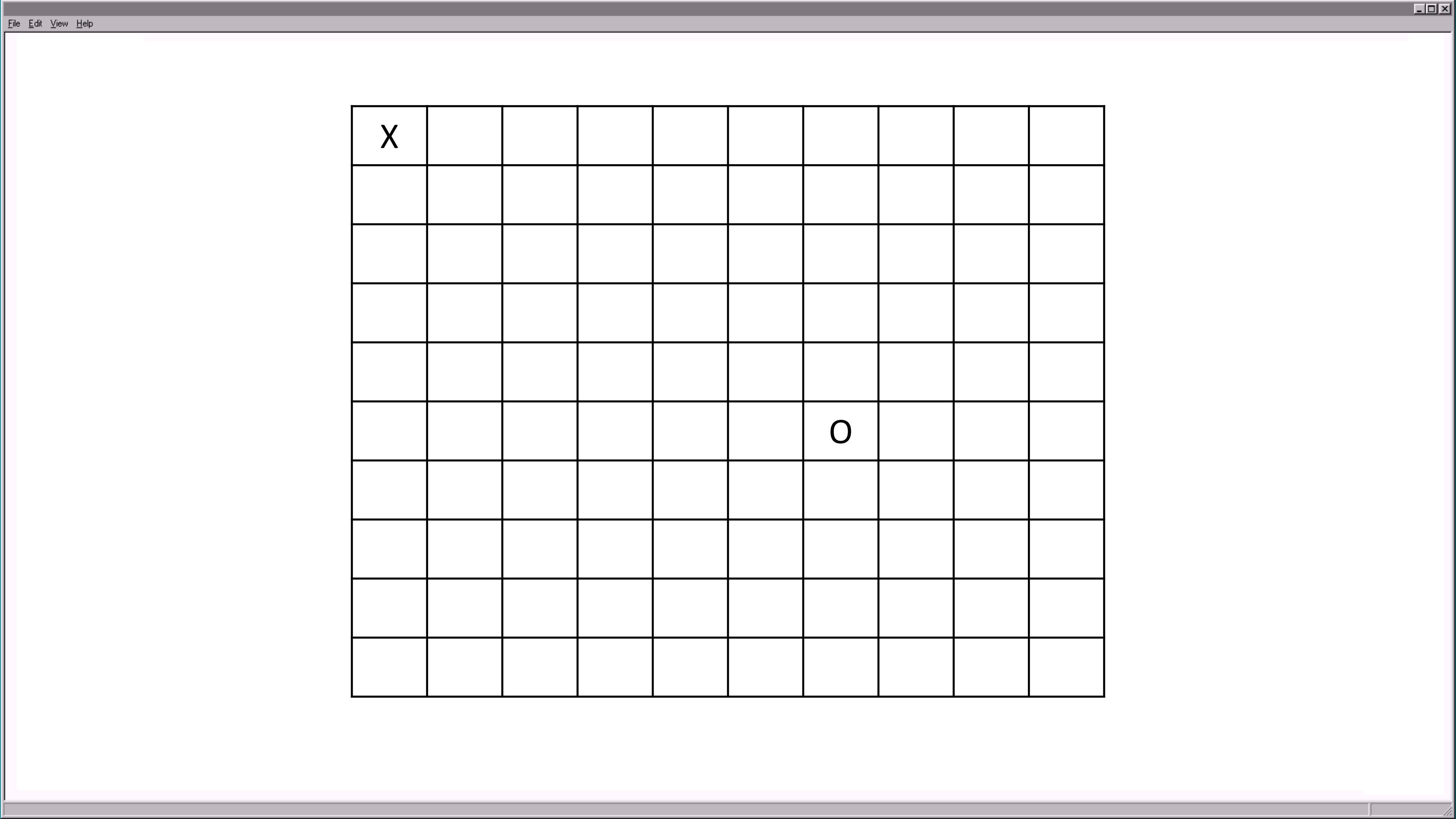
Example

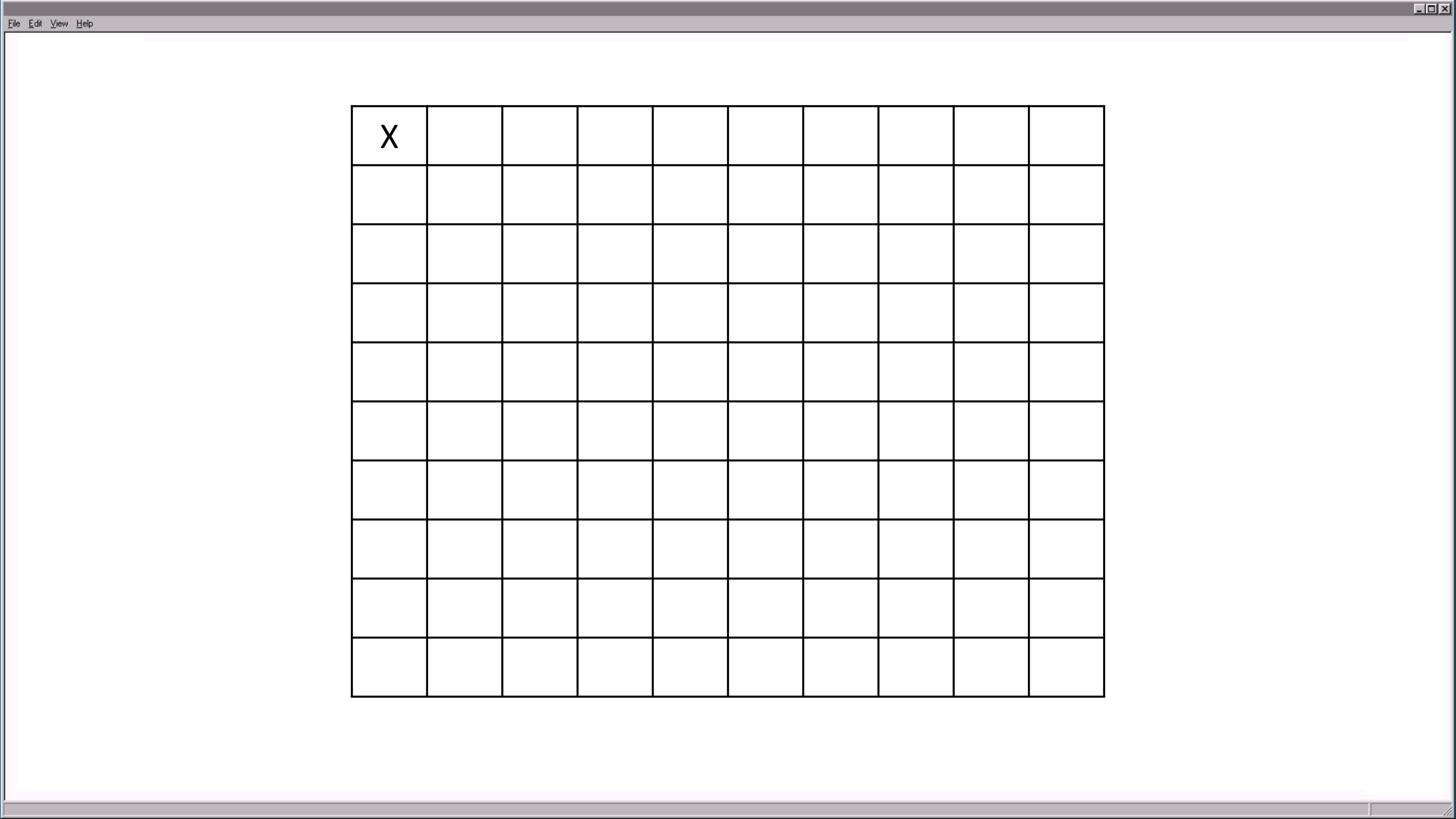
File
Edit
View
Help

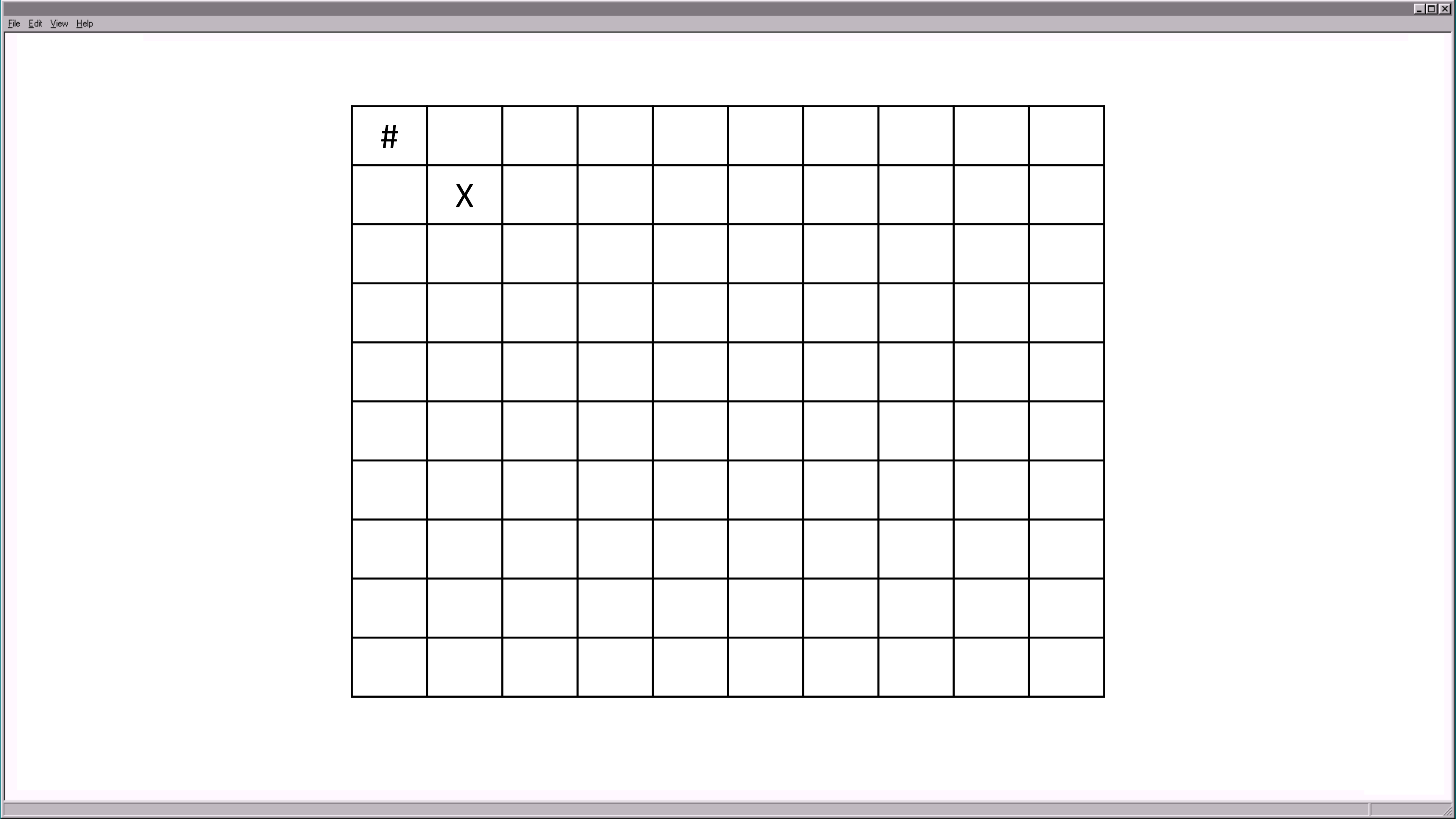

File
Edit
View
Help



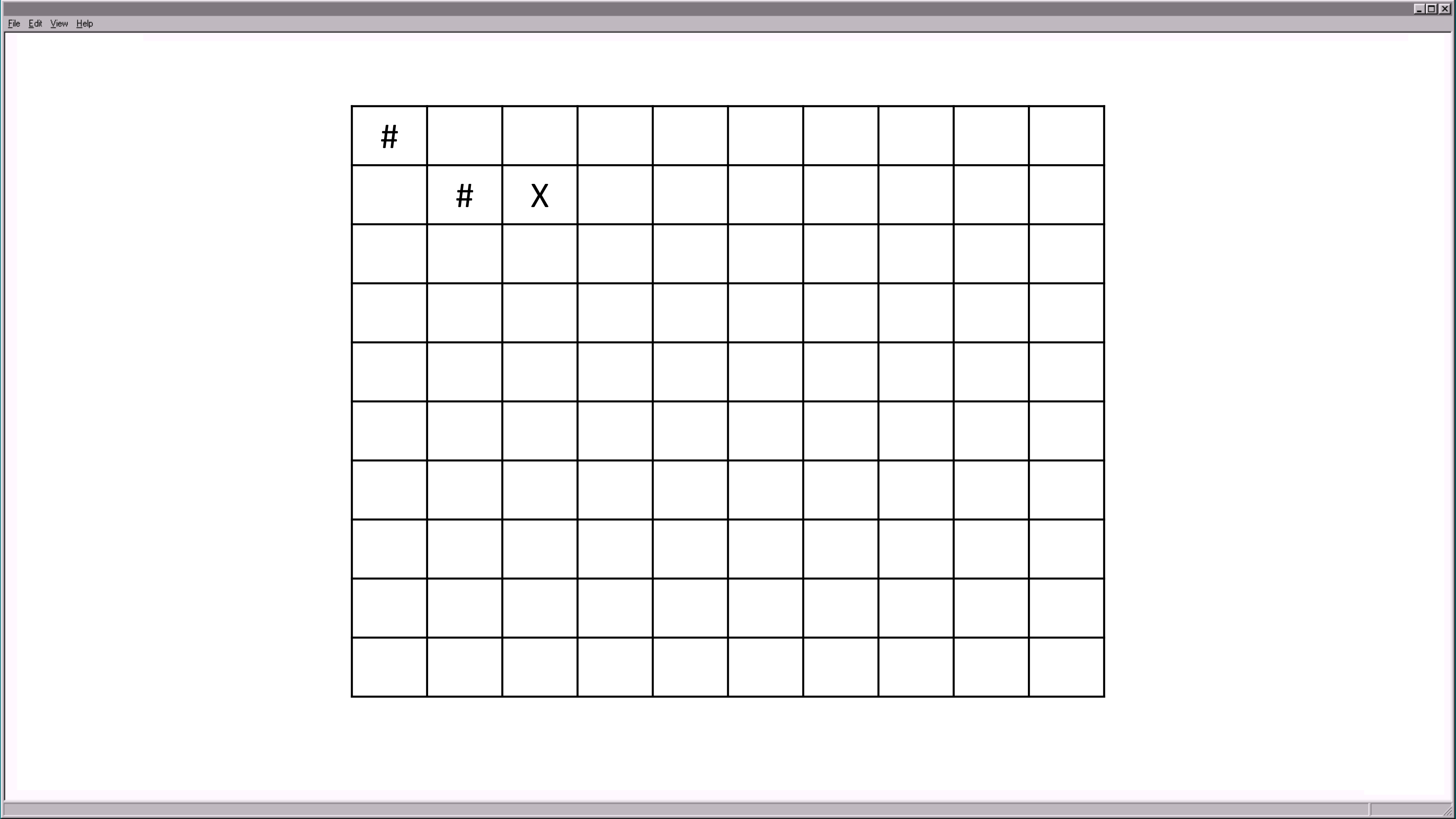
X									


File
Edit
View
Help
X
O

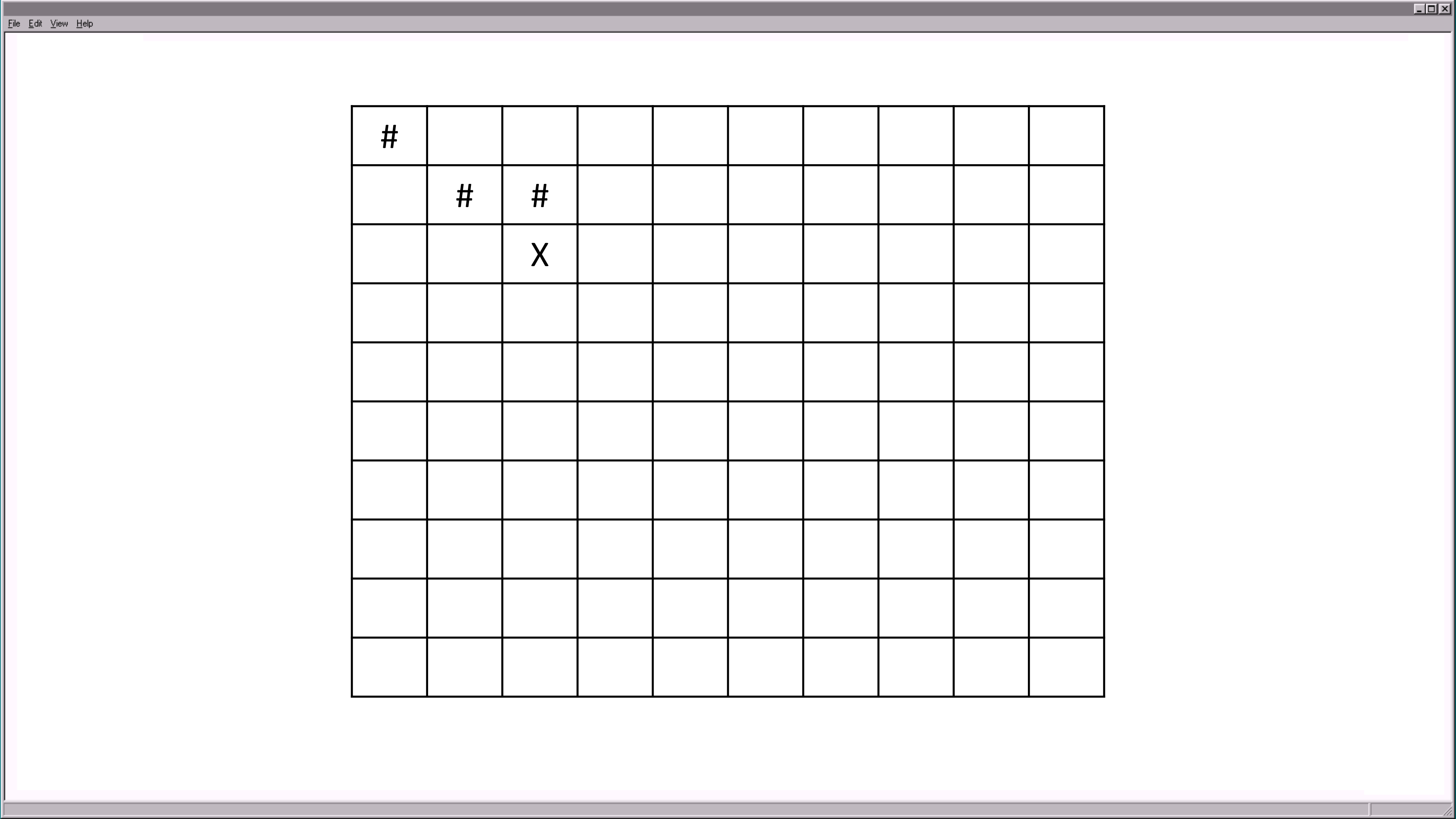

File
Edit
View
Help
X



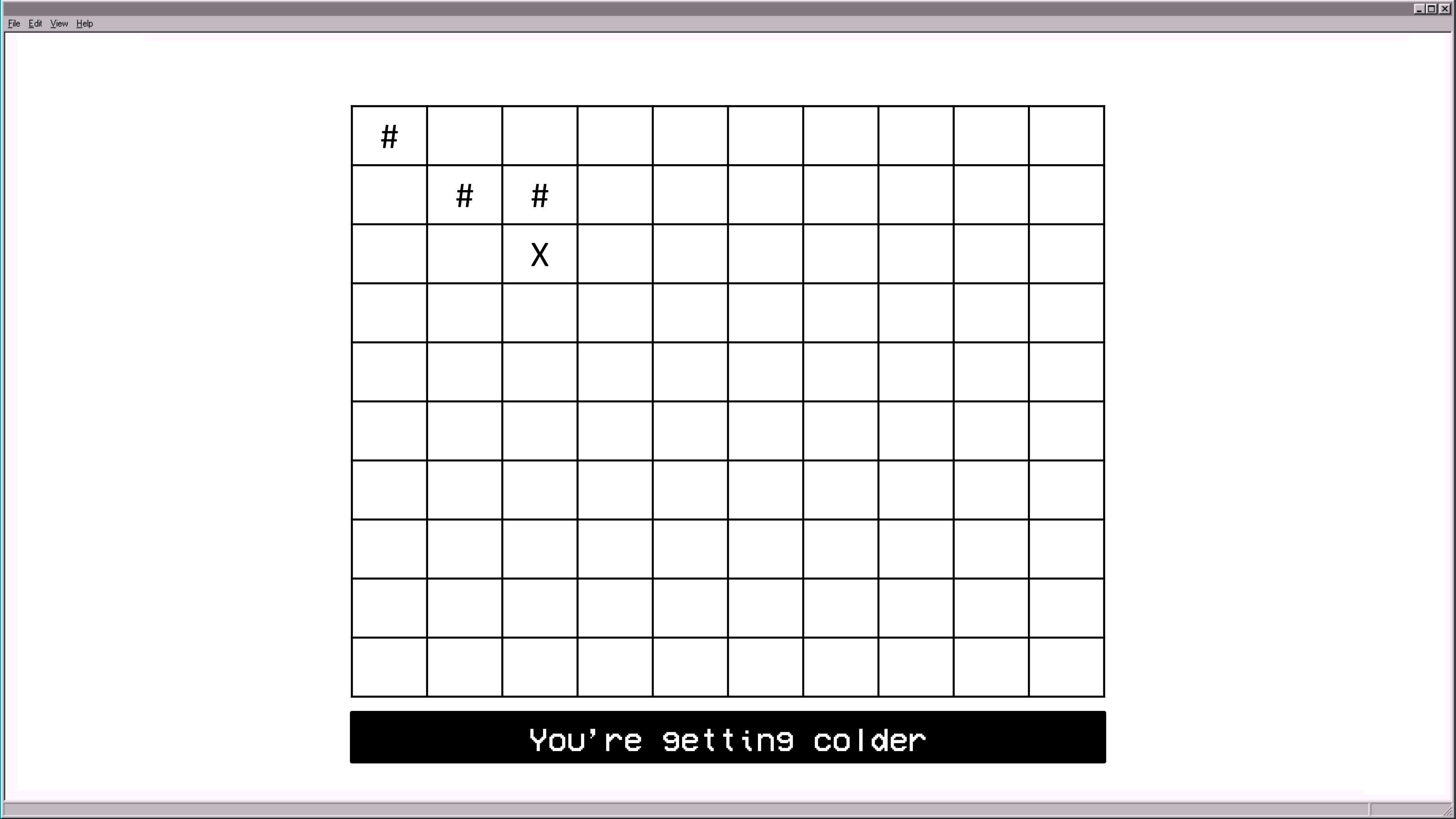
#									
	X								



#									
	#	X							

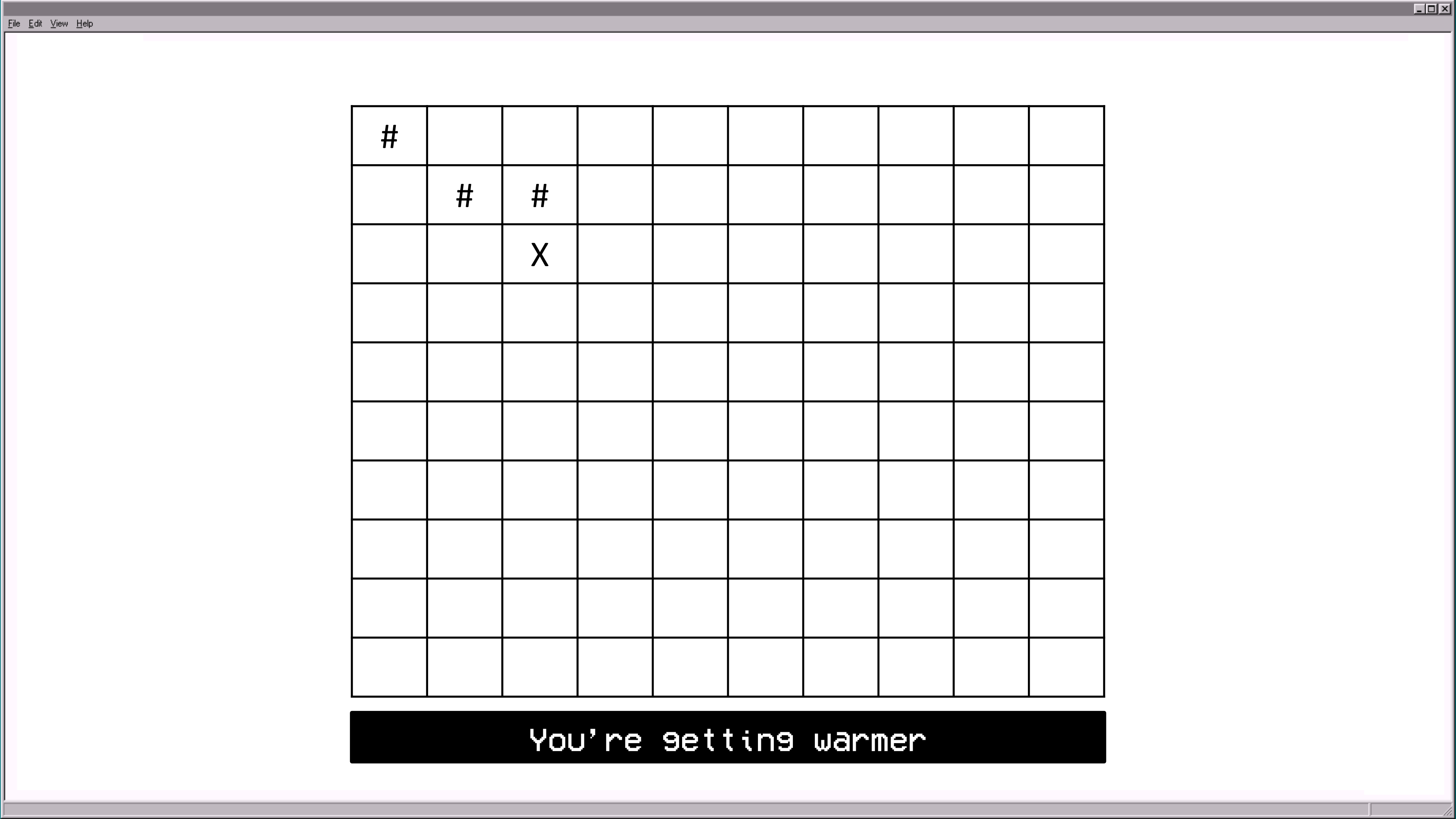


#									
	#	#							
		X							



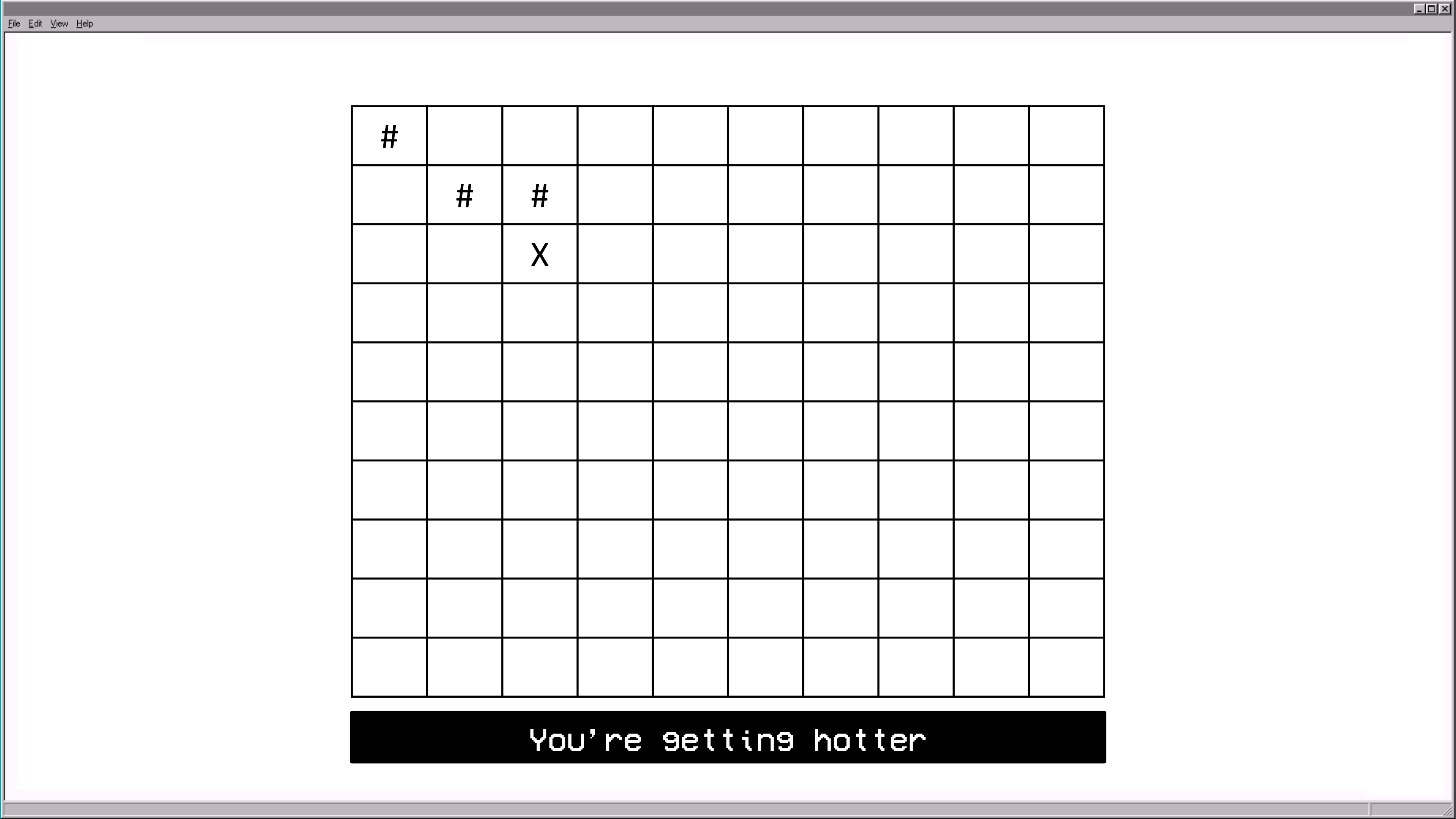
#									
	#	#							
		X							

You’re getting colder



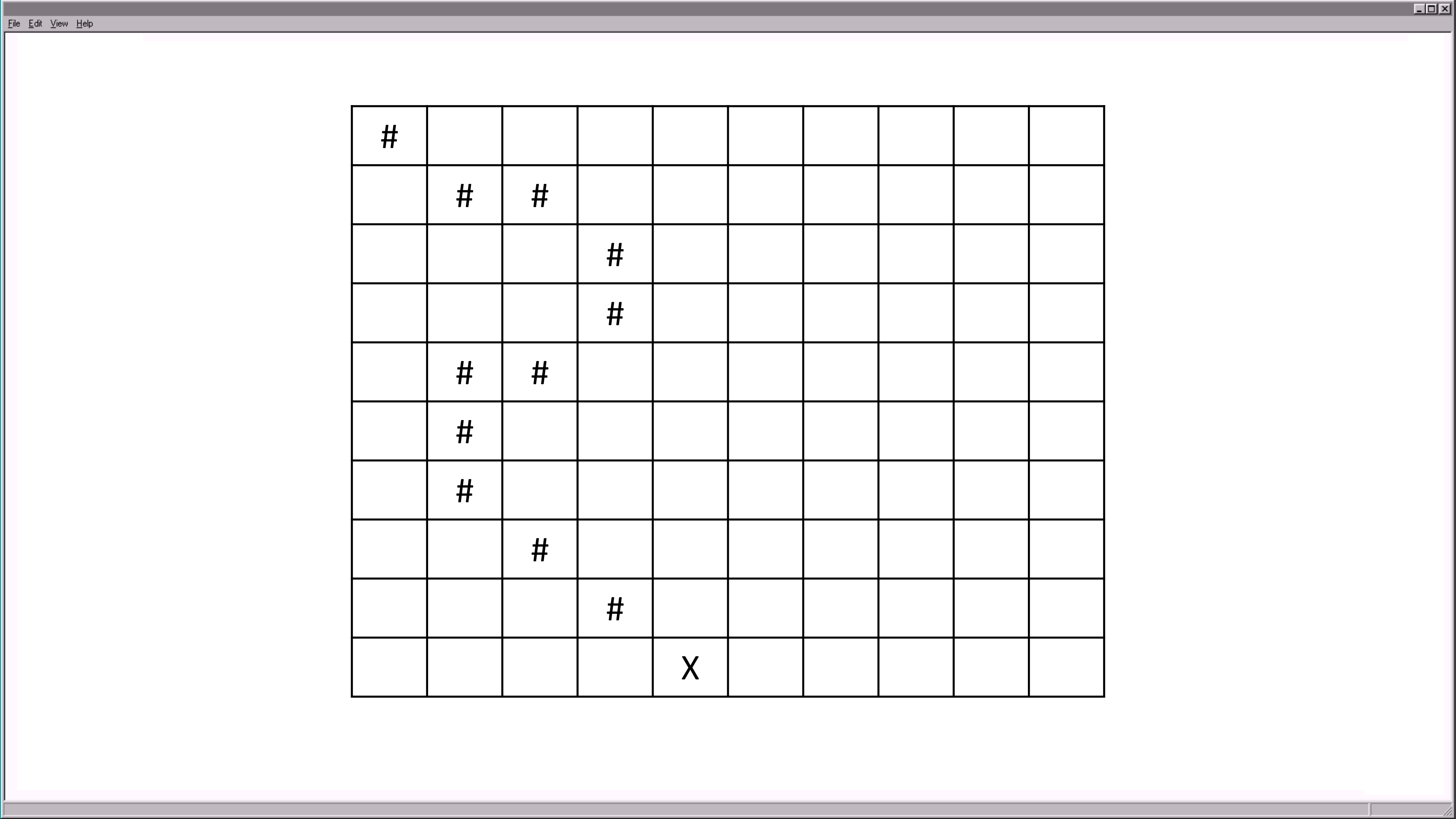
#									
	#	#							
		X							

You're getting warmer

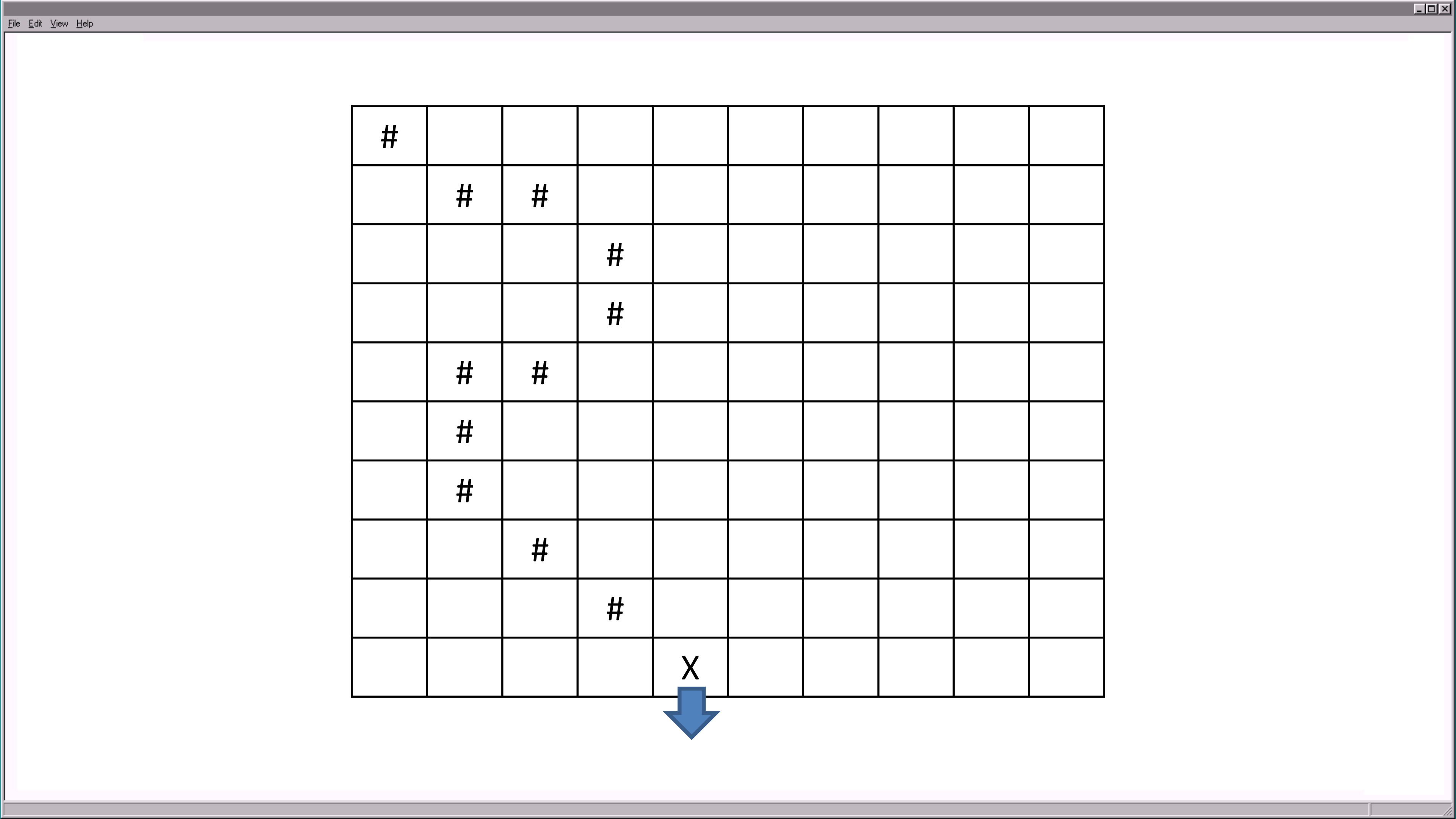


#									
	#	#							
		X							

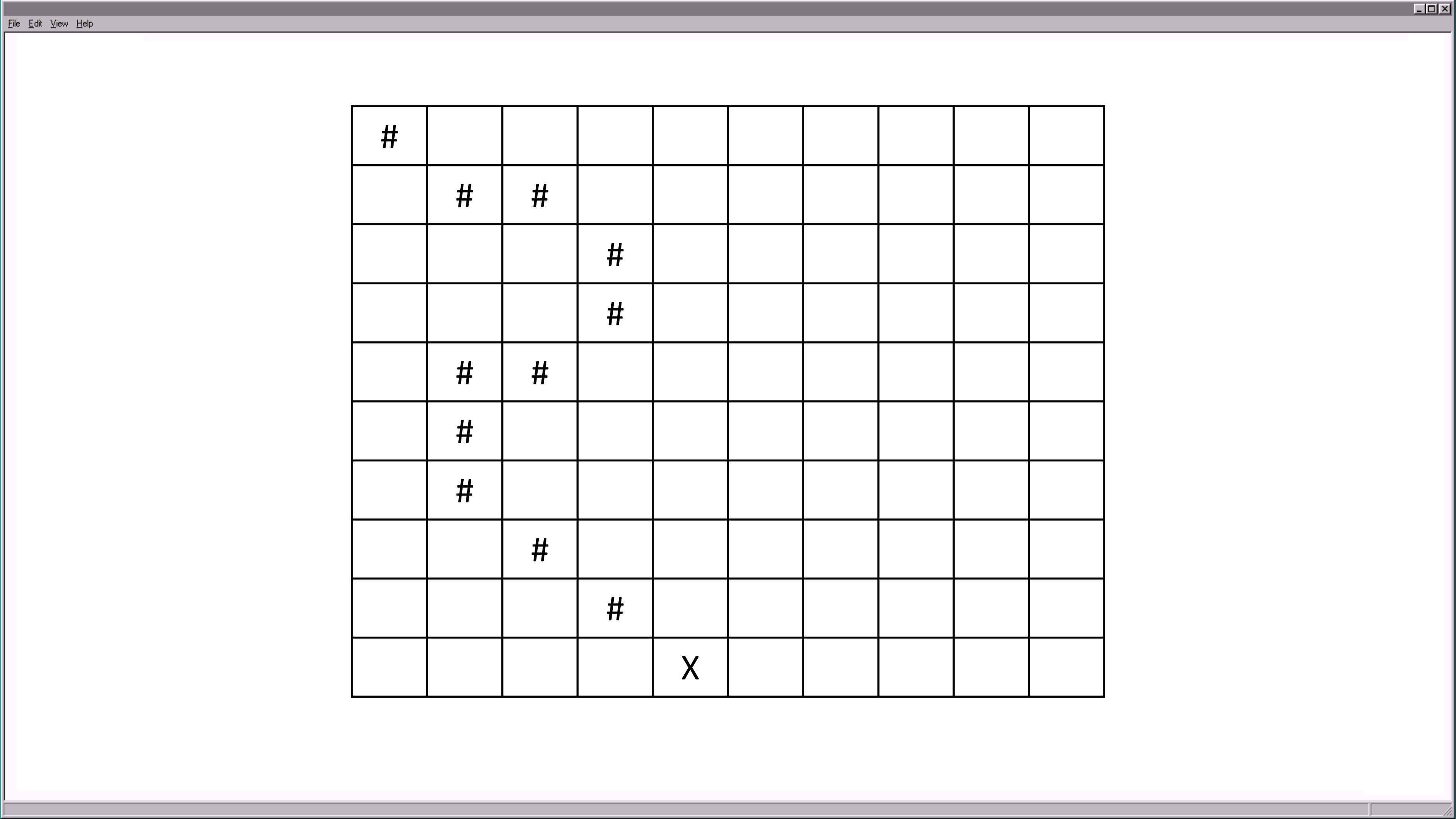
You're getting hotter



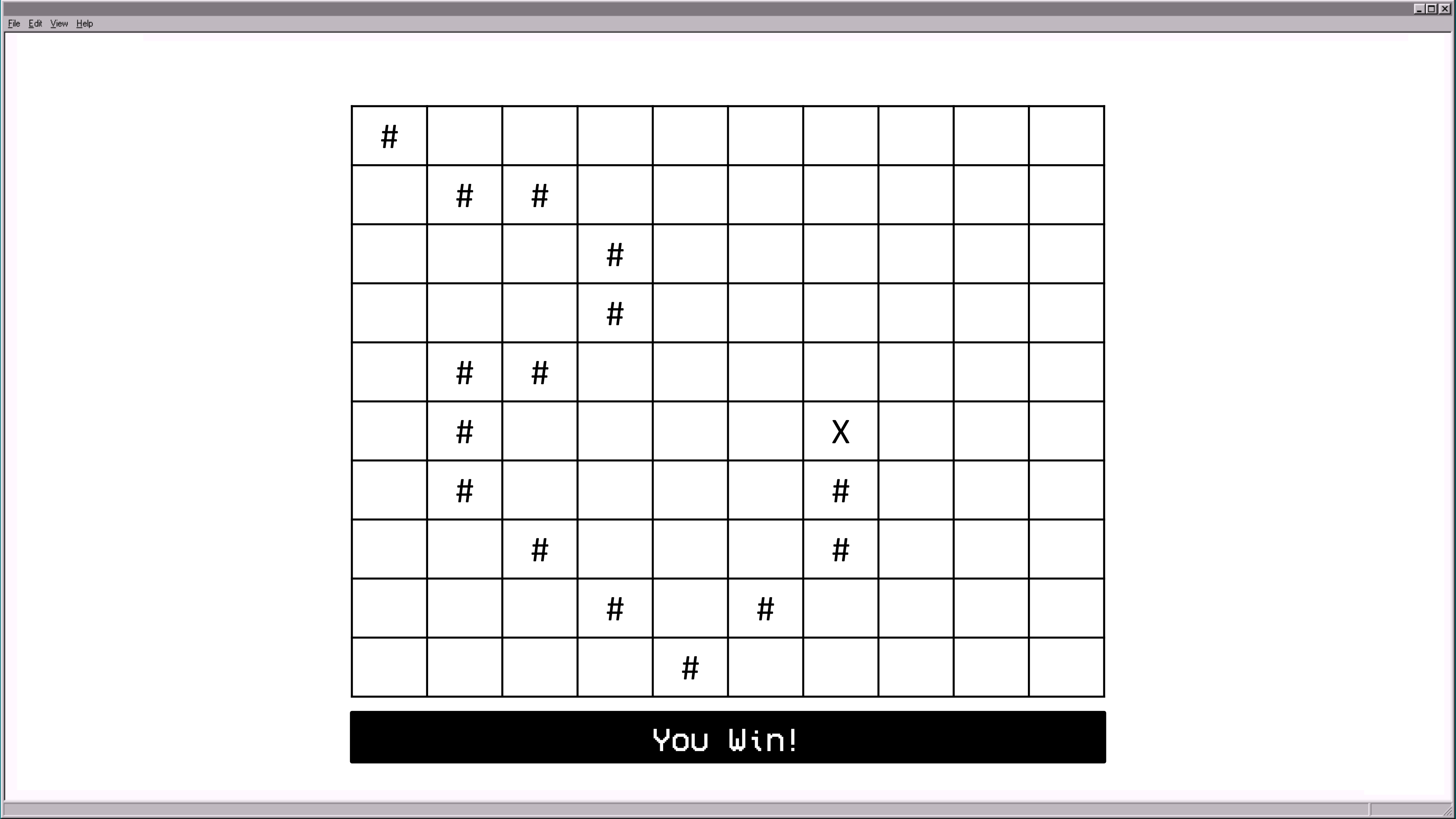
#									
	#	#							
			#						
			#						
	#	#							
	#								
	#								
		#							
			#						
				X					



#									
	#	#							
			#						
			#						
	#	#							
	#								
	#								
		#							
			#						
				X					

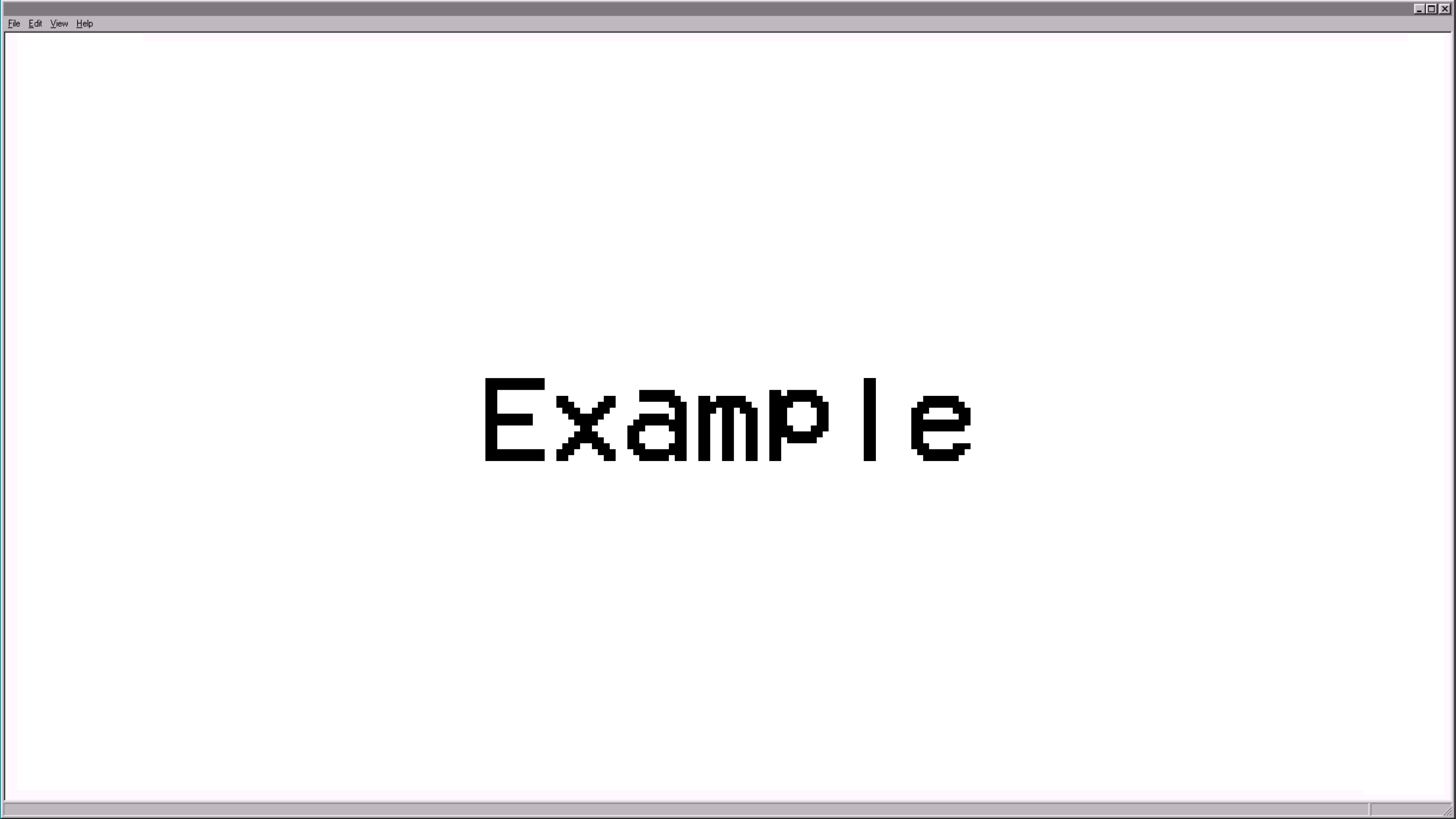


#									
	#	#							
			#						
			#						
	#	#							
	#								
	#								
		#							
			#						
				X					



#									
	#	#							
			#						
			#						
	#	#							
	#					X			
	#					#			
		#				#			
			#		#				
				#					

You Win!



Example

Multidimensional Arrays

- If the values are known, it is possible to both construct the array and initialize the values at the same time.
- Values are put inside of curly braces ("{}")
- Each value is separated by a comma (",")
- For each dimension include additional curly braces ("{}")

Creating an 2D Array Syntax

```
//Declaring an Array and Initializing its Values
<<type>>[][] <<id>> = {{<<00>>,<<01>>…},{<<10>>,<<11>>,…}};
```

Example

```
//Creates a 2 x 3 2D Array of integers
int[][] array = {{0,1,2},{1,2,3}};
```

Ragged Arrays

- Java allows multidimensional arrays to have different sizes for each dimension
  - Referred to as “Ragged Arrays”
  - Important to use <<id>>[<<index>>].length to ensure the correct size
- Not all programming languages allow this

Creating a Ragged 2D Array Syntax

```
//Declaring a Ragged Array
<<type>>[][] <<id>>;
<<id>> = new <<type>>[<<size for outside array>>];
<<id>>[<<index0>>] = new <<type>>[<<size at index 0>>];
<<id>>[<<index1>>] = new <<type>>[<<size at index 1>>];
…
```

Example

```
//Declare the array
int[][] a;
//Construct outside array
a = new int[3];
//Construct internal arrays
a[0] = new int[5];//First row has 5 elements
a[1] = new int[8];//Second row has 8 elements
a[2] = new int[2];//Third row has 2 elements
```