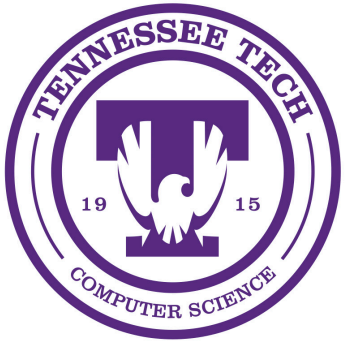
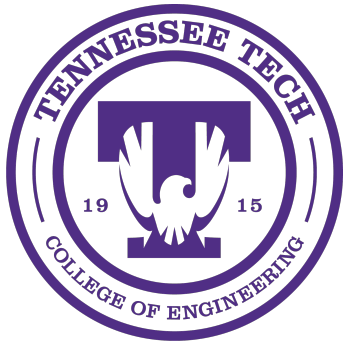


Tennessee
TECH



Strings and Pointers

Ahsan Ayub

Ph.D. Student, Department of Computer Science

Graduate Research Assistant, CEROC

CSC 1300: Introduction to Programming

Wednesday, November 3, 2021

Introducing Strings

A sequence of characters in memory. In other words, a string is simply a character array and can be manipulated as such.

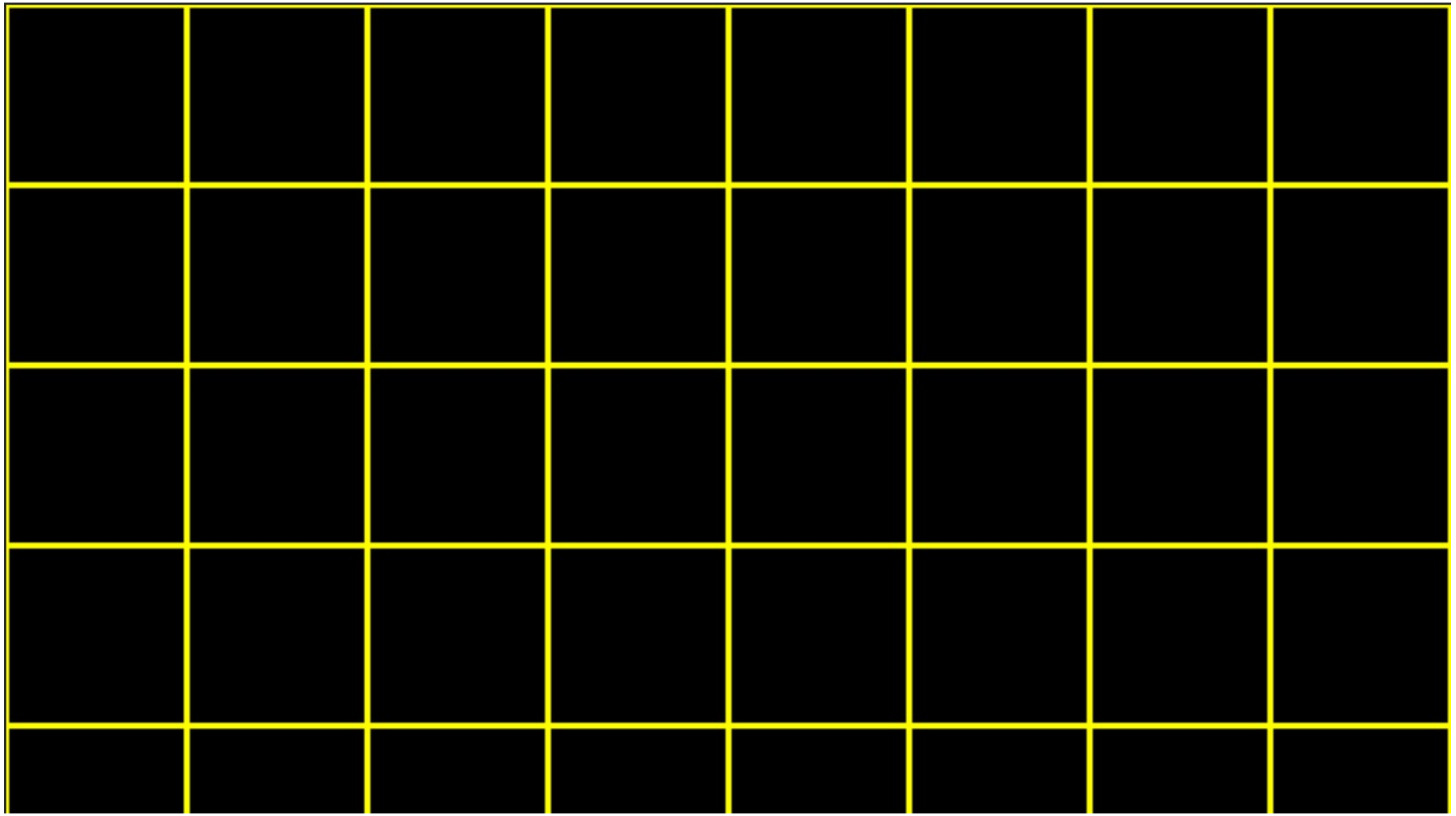
```
#include <iostream>
using namespace std;

int main()
{
        char s1[] = {'H', 'e', 'l', 'l', 'o', '\0'};
        char s2[3] = {'H', 'i', '\0'};
        char s3[] = "Hey";

        cout << s1 << endl;
        cout << s2 << endl;
        cout << s3 << endl;

        return 0;
}
```

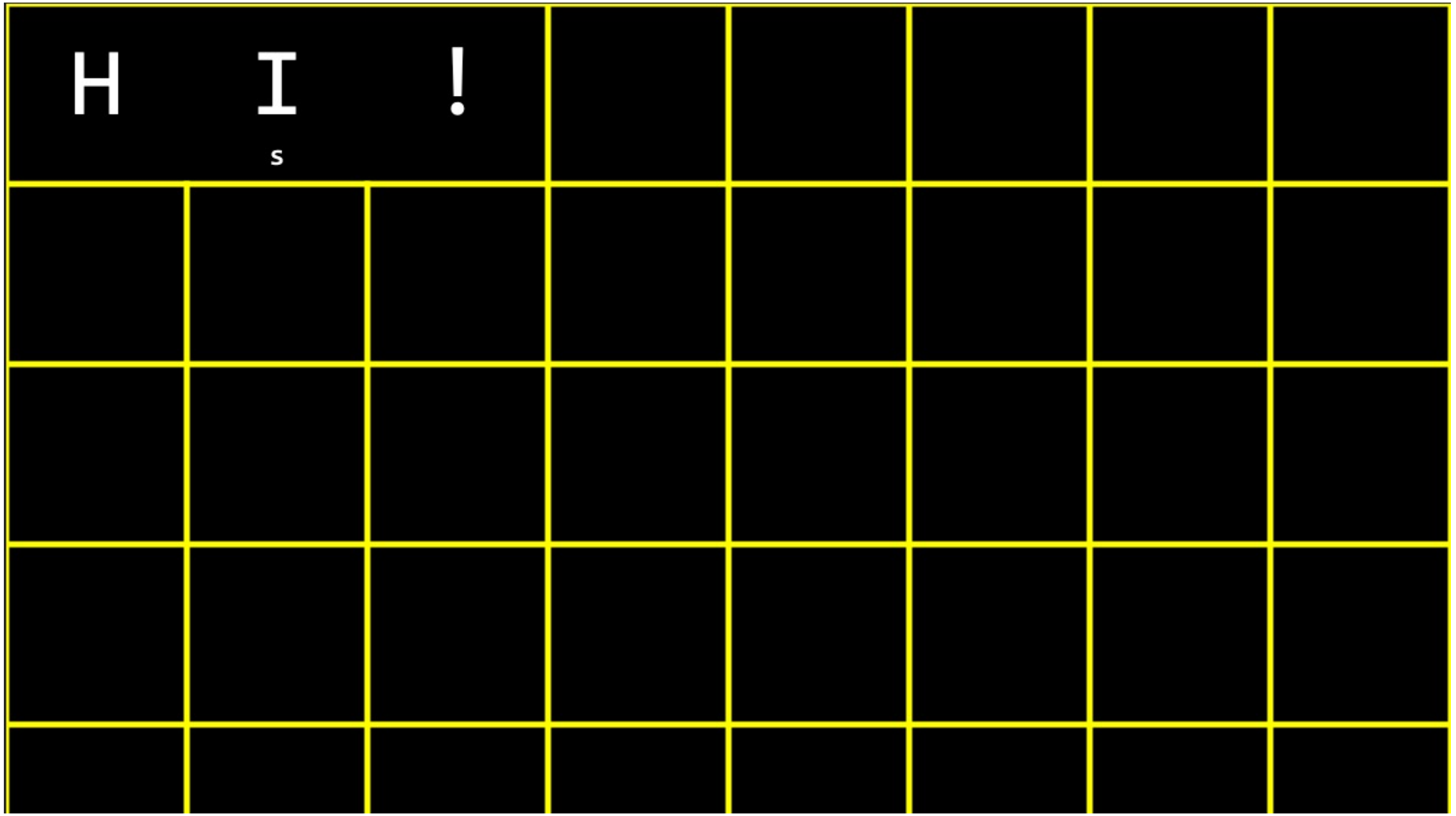
```
Hello
Hi
Hey
```

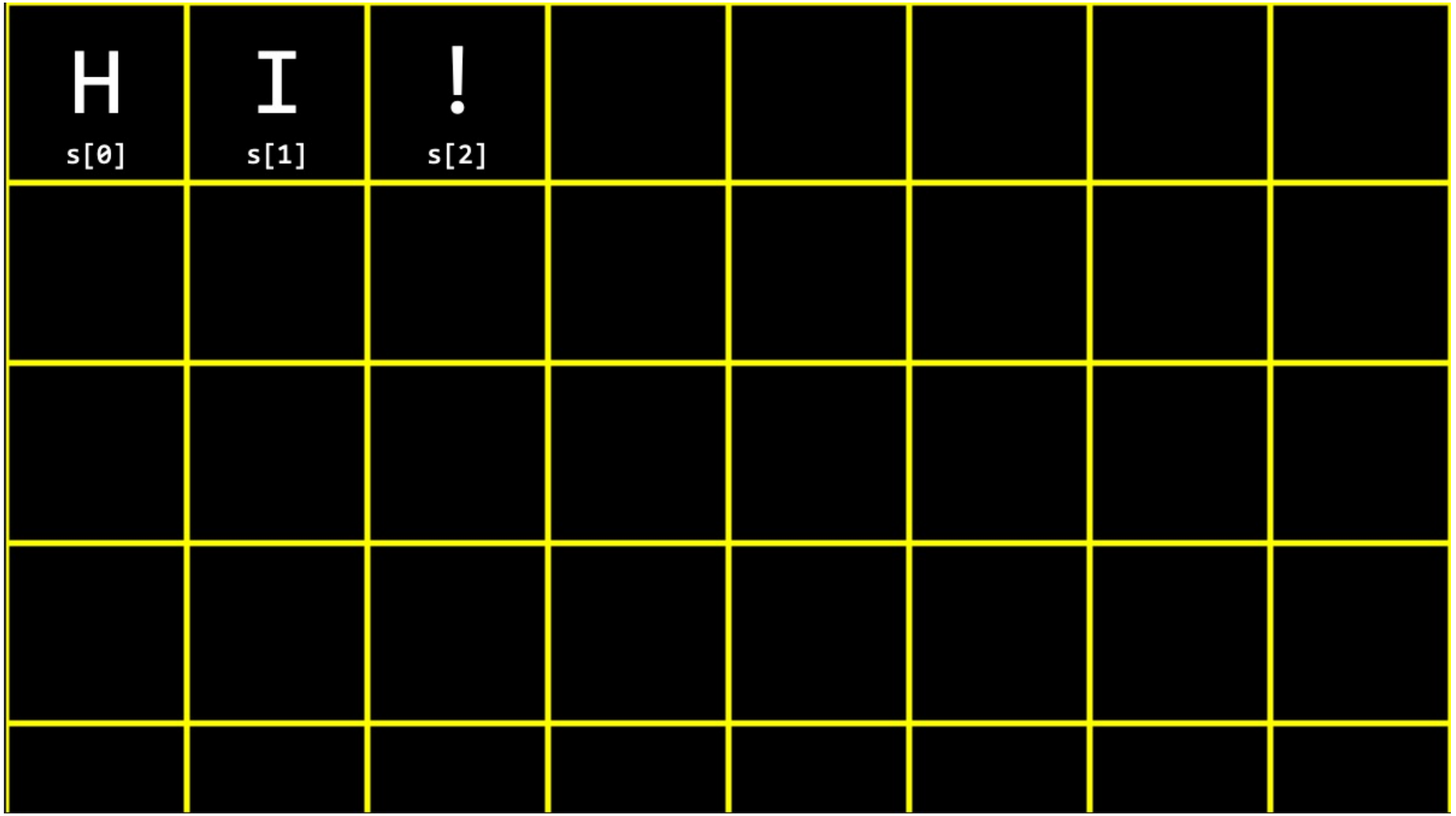
Introducing Strings

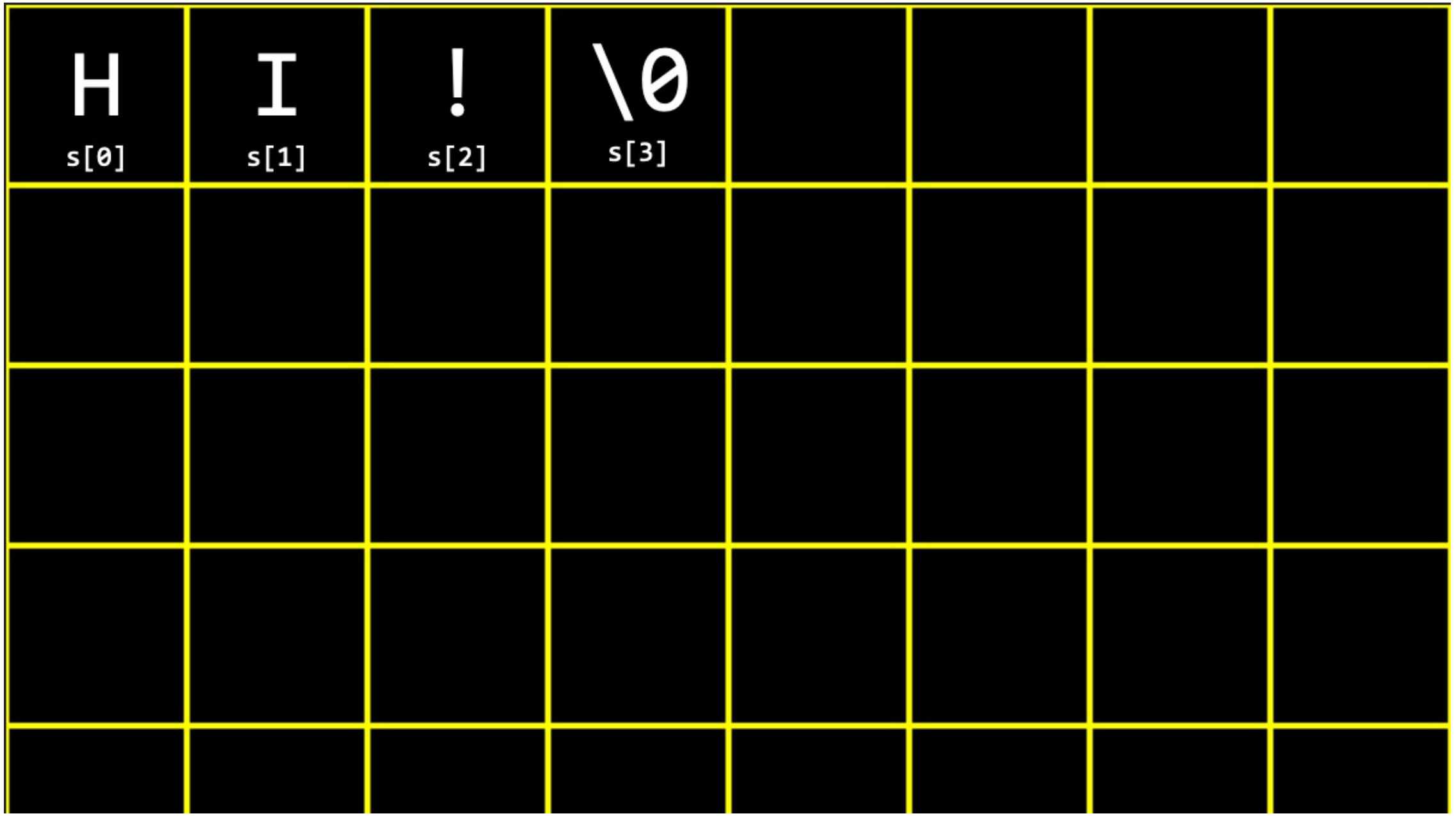
```
#include <iostream>
using namespace std;

int main()
{
        char s[] = “HI!”;

        cout << s<< endl;

        return 0;
}
```

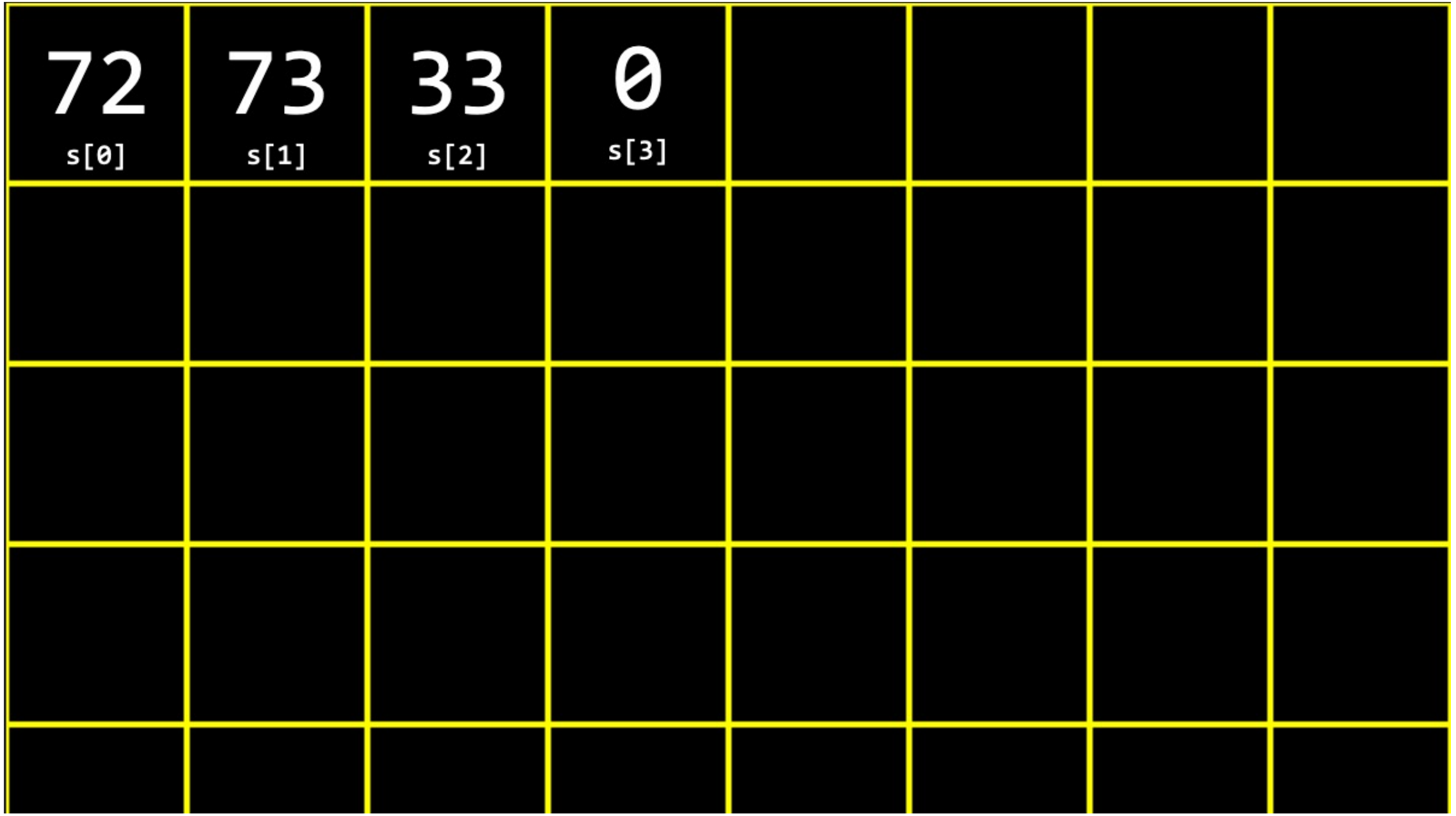
```
HI!
```

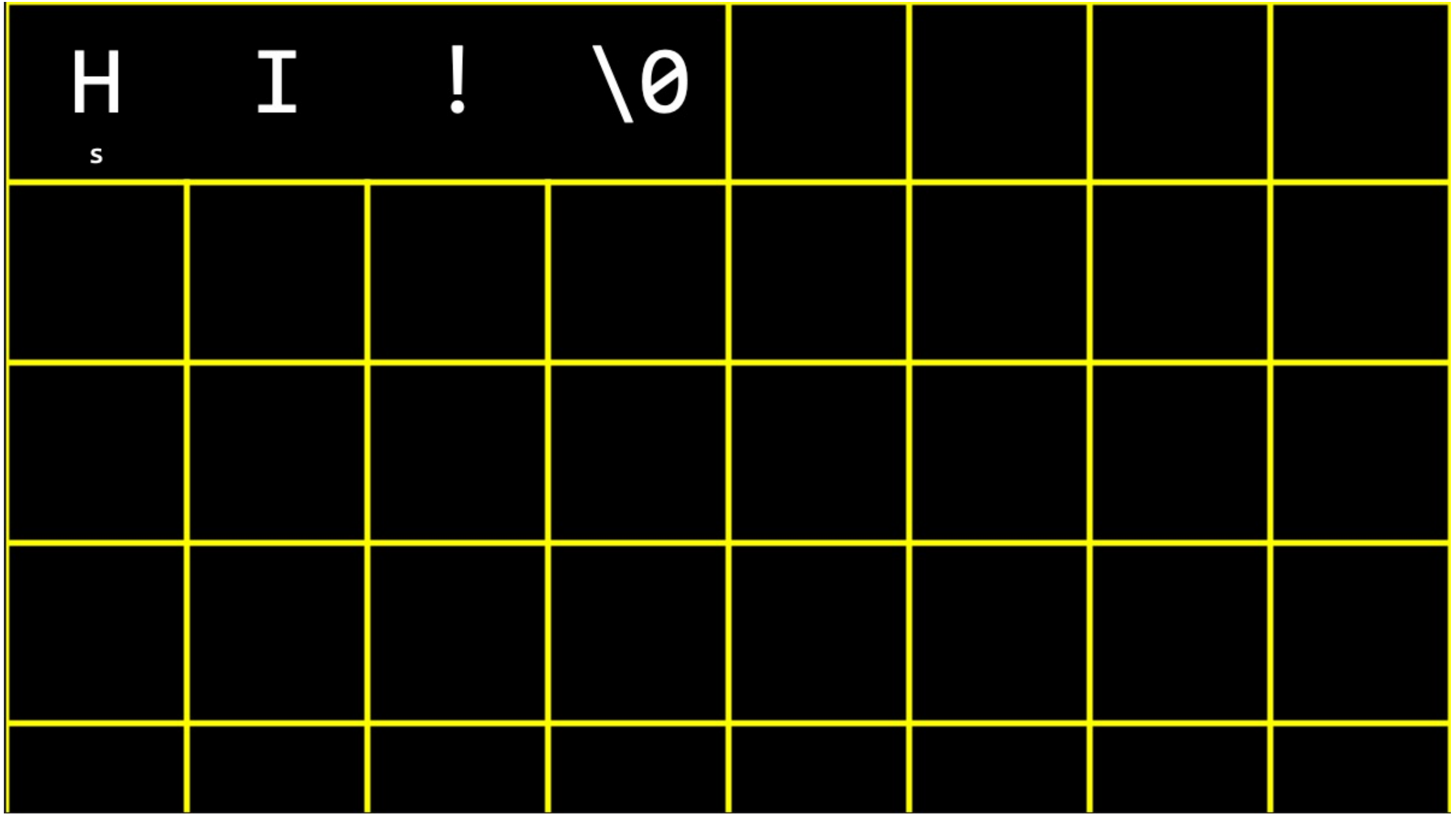



H I !
s


H
I
!
s[0]
s[1]
s[2]


H
s[0]
I
s[1]
!
s[2]
\0
s[3]


72
s[0]
73
s[1]
33
s[2]
0
s[3]


H
I
!
\0
s

Output Tracing

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

int main()
{
        // Declare a string of size 10
        char myString[10];

        char letter = 'A';
        // Initializing the array with alphabets values
        for (int i = 0; i < 9; i++)
        {
                myString[i] = letter;
                letter += 1;
        }
        myString[9] = '\0';
        cout << myString << endl;
}
```

```
??
```

Output Tracing

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

int main()
{
        // Declare a string of size 10
        char myString[10];

        char letter = 'A';
        // Initializing the array with alphabets values
        for (int i = 0; i < 9; i++)
        {
                myString[i] = letter;
                letter += 1;
        }
        myString[9] = '\0';
        cout << myString << endl;
}
```

```
ABCDEFGHI
```

Output Tracing

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

int main()
{
        // Declare a string of size 10
        char myString[] = “United States of America";

        unsigned int stringLength = 0;
        while(myString[stringLength] != ‘\0’)
                stringLength += 1;

        cout << stringLength << endl;
        return 0;
}
```

```
??
```

Output Tracing

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

int main()
{
        // Declare a string of size 10
        char myString[] = “United States of America";

        unsigned int stringLength = 0;
        while(myString[stringLength] != ‘\0’)
                stringLength += 1;

        cout << stringLength << endl;
        return 0;
}
```

```
24
```

Output Tracing

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

int main()
{
        // Declare a string of size 10
        char myString[] = “United States of America";

        unsigned int stringLength = 0;
        while(myString[stringLength])
                stringLength += 1;

        cout << stringLength << endl;
        return 0;
}
```

```
24
```

Character Handling

- `cctype` library (in C++; `ctype.h` for C language) is used to manipulate characters inside a string.
- Common functions include –
  - Boolean return function types: `isalpha`, `isupper`, `islower`, `ispunct`, `isspace`, etc.
  - Character conversation function types: `toupper` and `tolower`

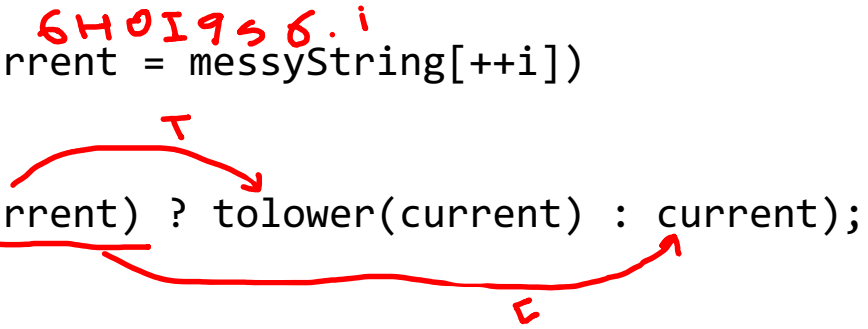
```
#include <iostream>
#include <cctype>
using namespace std;

int main()
{
        char s[] = "hello";
        int i = 0;
        while(s[i])
        {
            s[i] = toupper(s[i]);
            i++;
        }
        cout << s << endl;
        return 0;
}
```

```
HELLO
```

Reference (further read): https://www.cplusplus.com/reference/cctype/

Output Tracing

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

int main() {
   char messyString[] = "t6H0I9s6.iS.999a9.STRING";

   char current = messyString[0];
   for(int i = 0; current != '\0'; current = messyString[++i])
   {
       if(isalpha(current))
            cout << (char)(isupper(current) ? tolower(current) : current);
       else if(ispunct(current))
            cout << ' ';
   }
   cout << endl;
   return 0;
}
```



?? this is a string

Output Tracing

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

int main() {
   char messyString[] = "t6H0I9s6.iS.999a9.STRING";

   char current = messyString[0];
   for(int i = 0; current != '\0'; current = messyString[++i])
   {
       if(isalpha(current))
            cout << (char)(isupper(current) ? tolower(current) : current);
       else if(ispunct(current))
            cout << ' ';
   }
   cout << endl;
   return 0;
}
```

```
this is a string
```

String Handling

- `cstring` library (in C++; `string.h` for C language) is used to manipulate strings.
- Common functions include –
  - Copy string: `strcpy`.
  - Concatenate strings: `strcat`
  - Compare two strings: `strcmp`
  - Get string length: `strlen`

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

int main()
{
        char s1[100] = “CSC 1300-001”;
        char s2[100];
        char s3[] = “ Intro to Programming”

        strcpy(s2, s1);
        strcat(s1, s3);

        cout << s2 << endl;
        cout << s1 << endl;
        return 0;
}
```

```
CSC 1300-001
CSC 1300-001 Intro to Programming
```

Reference (further read): https://www.cplusplus.com/reference/cstring/

Output Tracing

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

int main() {
   char fragment1[] = "I'm a s";
   char fragment2[] = "tring";
   char fragment3[20];
   char finalString[20] = "";

   strcpy(fragment3, fragment1);
   strcat(finalString, fragment3);
   strcat(finalString, fragment2);

   cout << finalString << endl;
   return 0;
}
```

```
??
```

Output Tracing

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

int main() {
   char fragment1[] = "I’m a s";
   char fragment2[] = "tring";
   char fragment3[20];
   char finalString[20] = "";

   strcpy(fragment3, fragment1);   // fragment3 -> "I’m a s"
   strcat(finalString, fragment3); // finalString -> "I’m a s"
   strcat(finalString, fragment2); // finalString -> "I’m a string"

   cout << finalString << endl;
   return 0;
}
```

```
I’m a string
```

Read a “word” from the user

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

int main()
{
        char word[100];
        cout << “Enter a string: ”;
        cin >> word;
        cout << word << endl;

        cout << “Enter another string: ”;
        cin >> word;
        cout << word << endl;
        return 0;
}
```

```
Enter a string: _
```

Read a “word” from the user

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

int main()
{
        char word[100];
        cout << “Enter a string: ”;
        cin >> word;
        cout << word << endl;

        cout << “Enter another string: ”;
        cin >> word;
        cout << word << endl;
        return 0;
}
```

```
Enter a string: CSC1300-001
```

Read a “word” from the user

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

int main()
{
        char word[100];
        cout << “Enter a string: ”;
        cin >> word;
        cout << word << endl;

        cout << “Enter another string: ”;
        cin >> word;
        cout << word << endl;
        return 0;
}
```

```
Enter a string: CSC1300-001
CSC 1300-001
Enter another string: _
```

Read a “word” from the user

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

int main()
{
        char word[100];
        cout << “Enter a string: ”;
        cin >> word;
        cout << word << endl;

        cout << “Enter another string: ”;
        cin >> word;
        cout << word << endl;
        return 0;
}
```

```
Enter a string: CSC1300-001
CSC 1300-001
Enter another string: CSC1300-001 Introduction to Programming
```

Note: Space “ ” has a terminating character.

Read a “word” from the user

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

int main()
{
        char word[100];
        cout << “Enter a string: ”;
        cin >> word;
        cout << word << endl;

        cout << “Enter another string: ”;
        cin >> word;
        cout << word << endl;
        return 0;
}
```

```
Enter a string: CSC1300-001
CSC 1300-001
Enter another string: CSC1300-001 Introduction to Programming
CSC1300-001
```

Read a “line of text” from the user

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

int main()
{
        char word[100];
        cout << “Enter a string: ”;
        cin.get(word, 100);
        cout << word << endl;
        cin.ignore(); // To clear the buffer for the next input
        cout << “Enter another string: ”;
        cin.get(word, 100);
        cout << word << endl;
        return 0;
}
```

```
Enter a string: CSC1300-001
CSC 1300-001
Enter another string: CSC1300-001 Introduction to Programming
CSC1300-001 Introduction to Programming
```

Strings

- In C++, we can also create a string object to work with strings.
- Unlike using character arrays, it has got no fixed length.

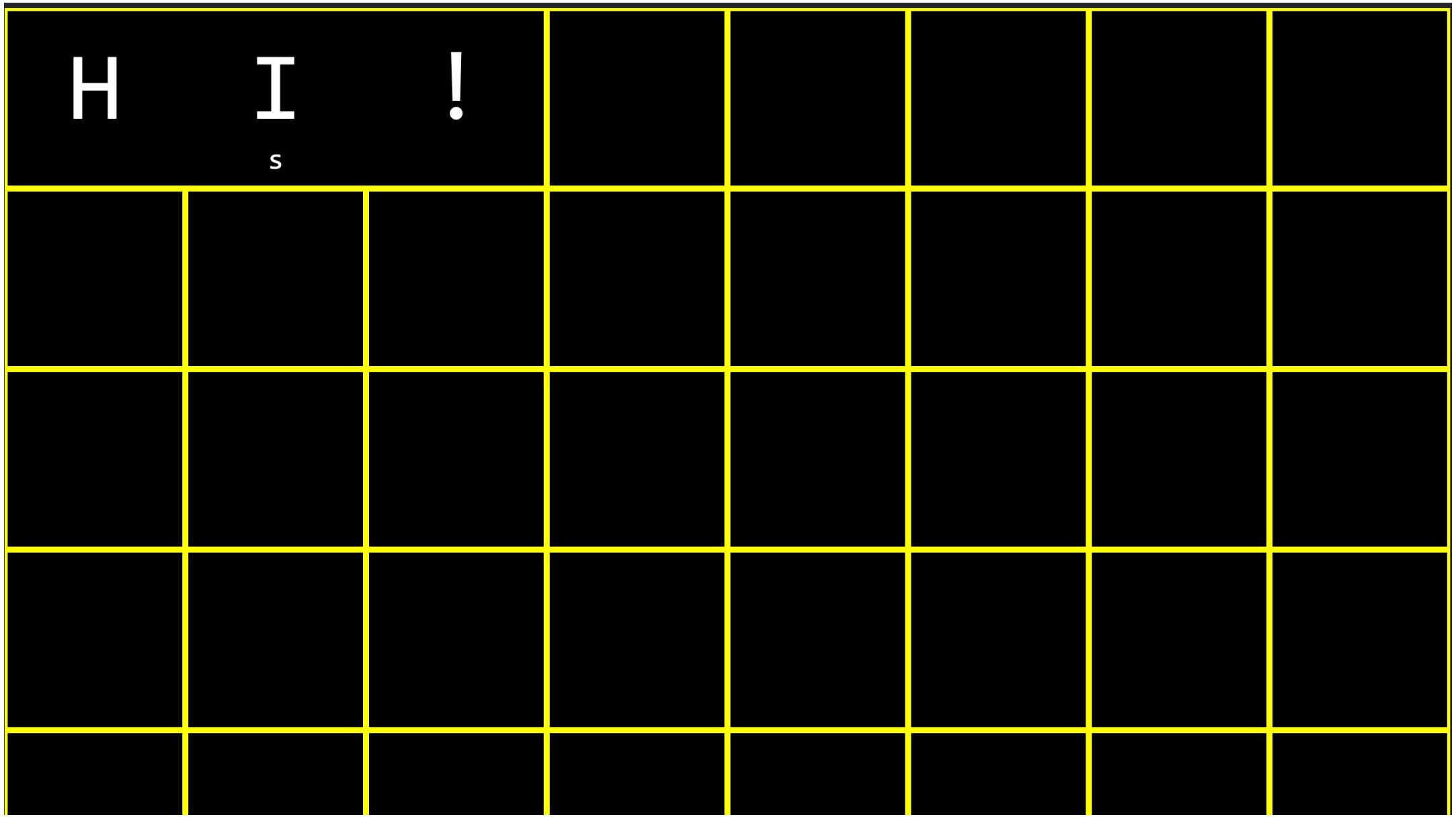
```
#include <iostream>
using namespace std;

int main()
{
        string s1= “Hello”;
        string s2 = “Hi”;
        string s3 = "Hey";

        cout << s1 << endl;
        cout << s2 << endl;
        cout << s3 << endl;

        return 0;
}
```

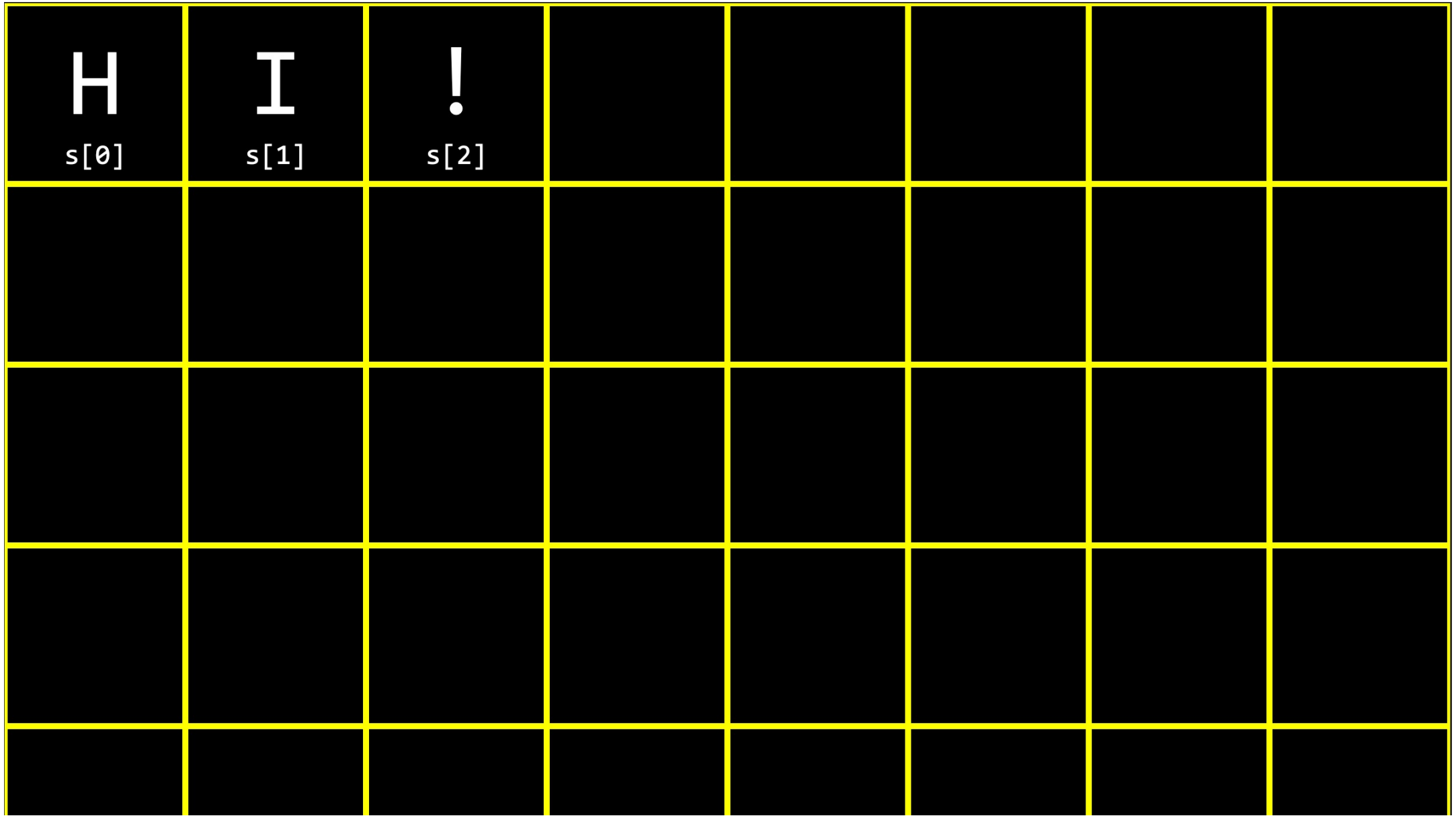
```
Hello
Hi
Hey
```

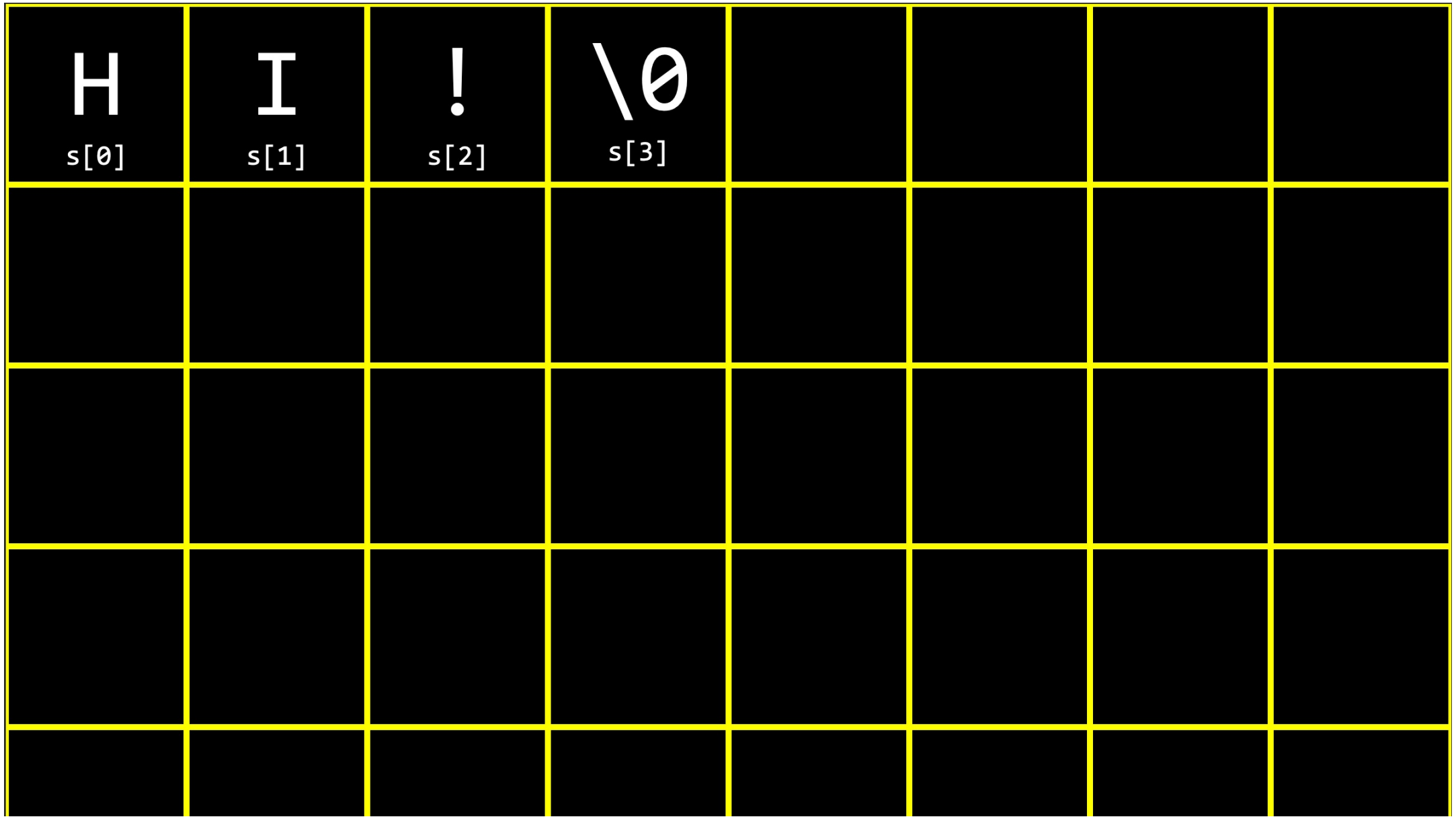
Strings Object

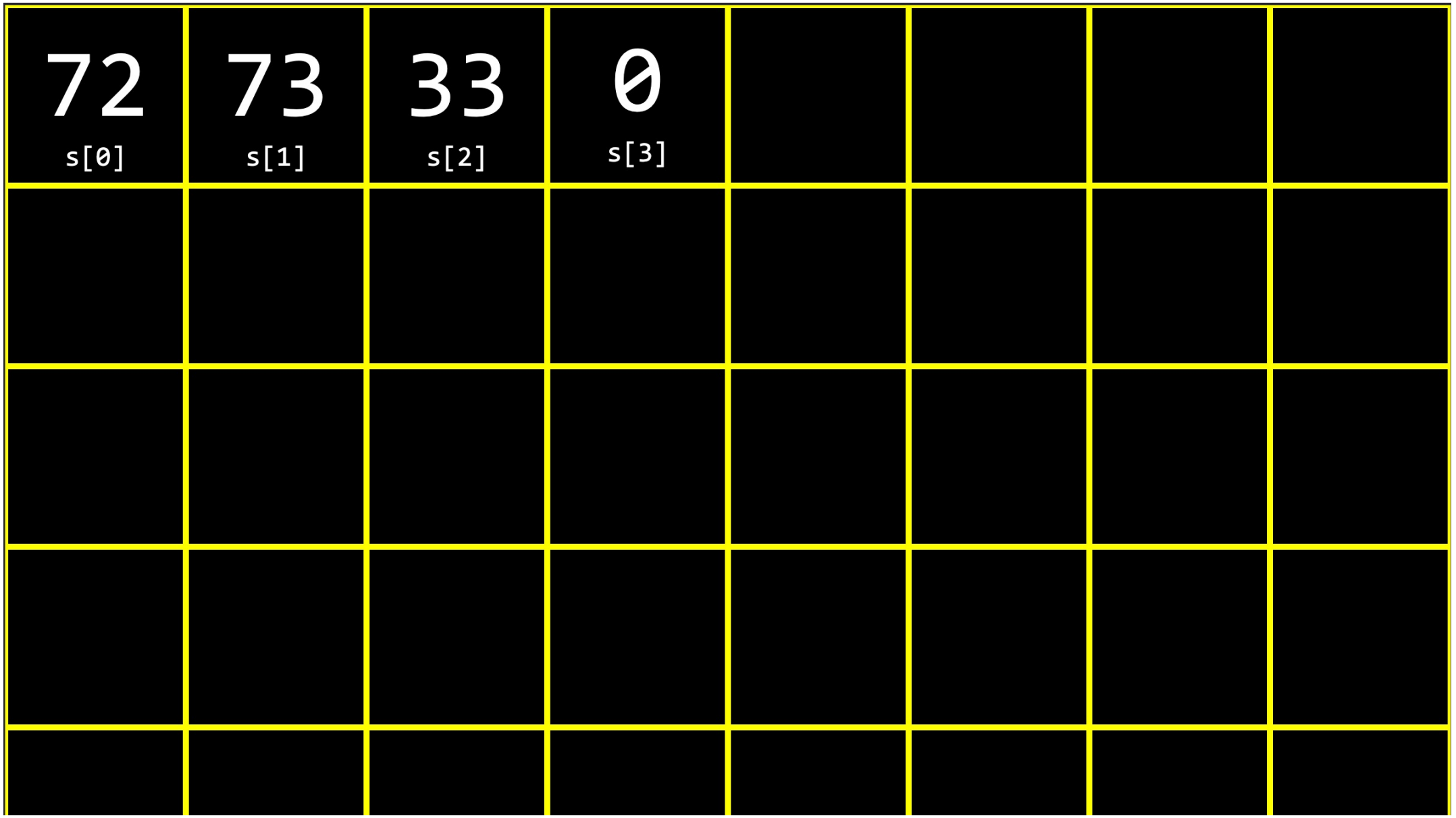
```
#include <iostream>
using namespace std;

int main()
{
        string s = “HI!”;

        cout << s << endl;

        return 0;
}
```

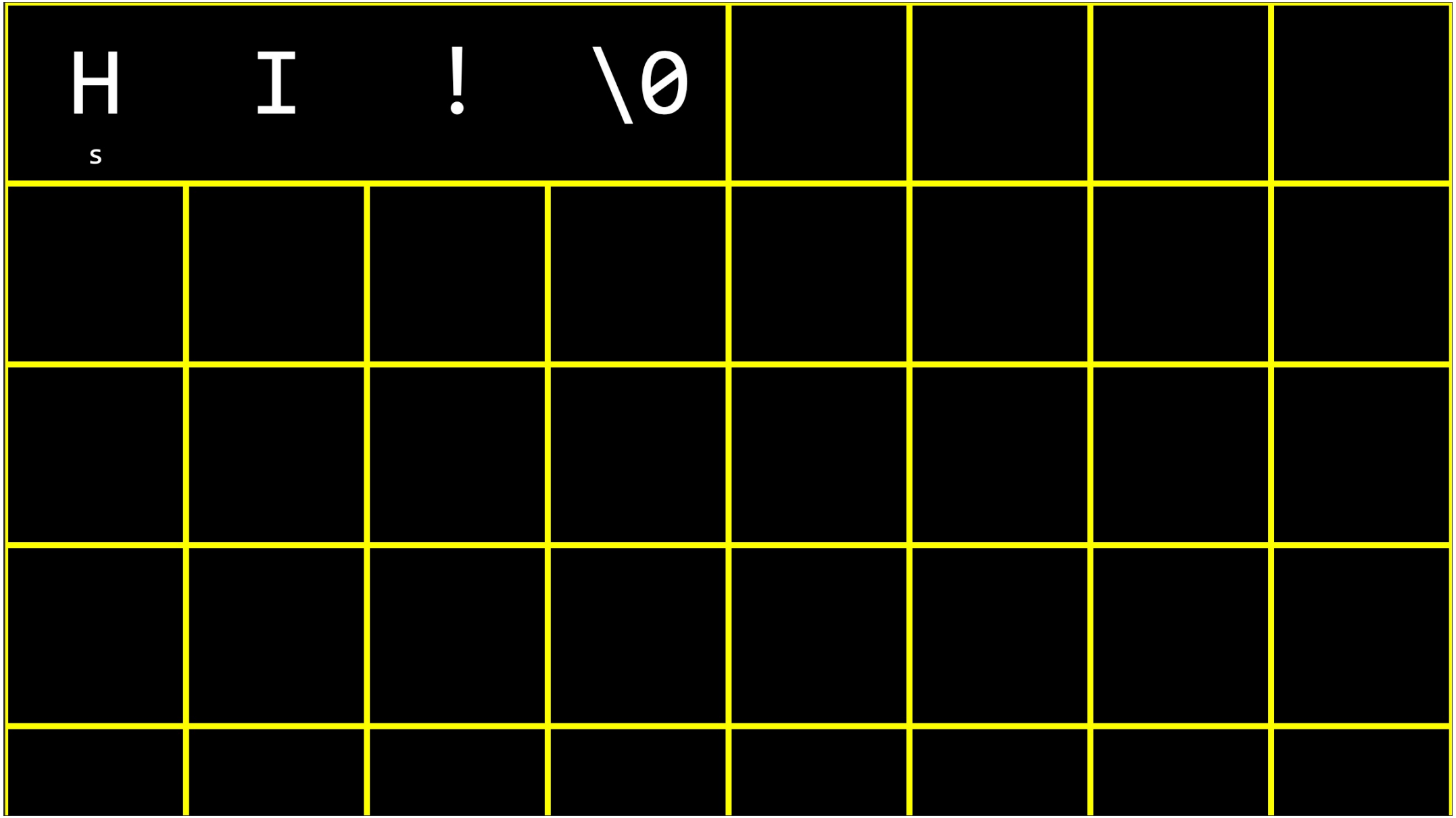
```
HI!
```


H I !
s


H
I
!
s[0]
s[1]
s[2]


H
s[0]
I
s[1]
!
s[2]
\0
s[3]

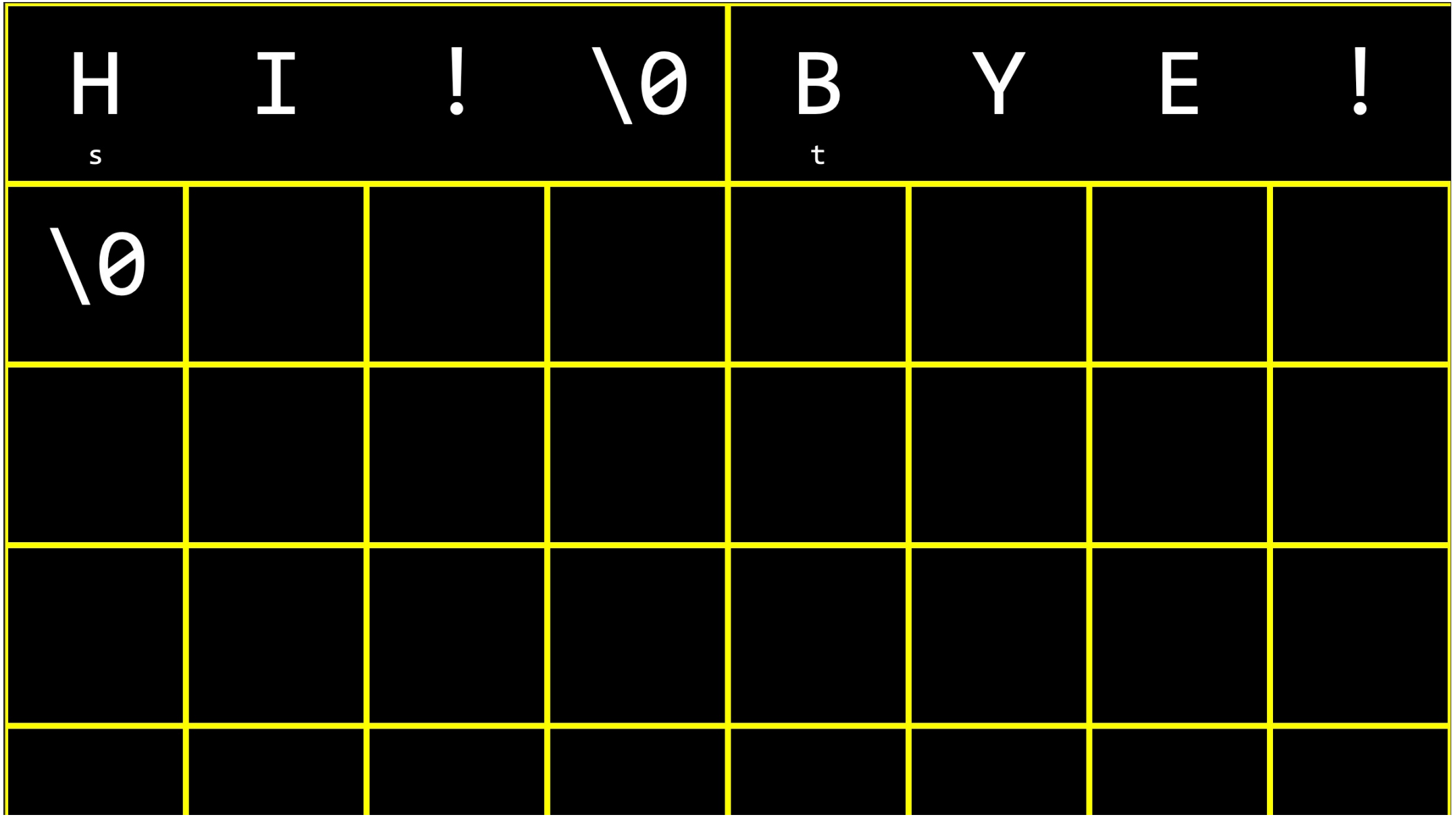

72
s[0]
73
s[1]
33
s[2]
0
s[3]

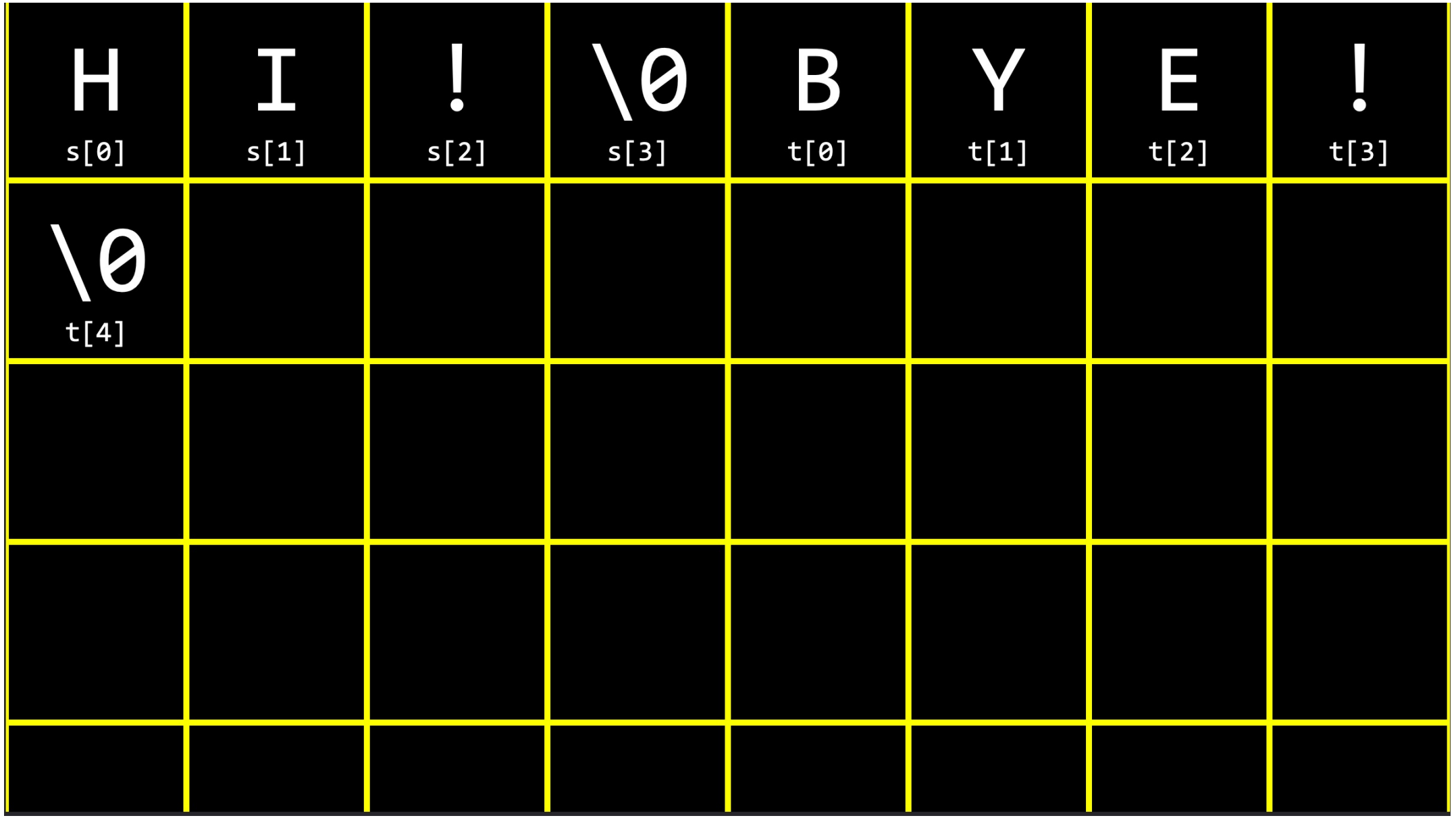

H
I
!
\0
s

Strings Object

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

int main()
{
        string s = “HI!”;
        string t = “BYE!”;

        cout << s << endl;
        cout << t << endl;

        return 0;
}
```

```
HI!
BYE!
```


H I ! \0
s
B Y E !
t
\0


H
s[0]
I
s[1]
!
s[2]
\0
s[3]
B
t[0]
Y
t[1]
E
t[2]
!
t[3]
\0
t[4]

Read a “line of text” from the user

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

int main()
{
        string message;
        cout << “Enter a string: ”;
        getline(cin, message);
        cout << message << endl;
        cin.clear(); // To clear the buffer for the next input
        cout << “Enter another string: ”;
        getline(cin, message);
        cout << message << endl;
        return 0;
}
```

```
Enter a string: CSC1300-001
CSC 1300-001
Enter another string: CSC1300-001 Introduction to Programming
CSC1300-001 Introduction to Programming
```

String Functionalities

- Common functions include –
  - Convert string to integer: `stoi`
  - Convert string to unsigned integer: `stoul`
  - Convert string to long int: `stol`
  - Convert string to double: `stod`
  - Convert to string: `to_string`

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

int main()
{
        string str1 = “104”;
        int val1 = stoi(str1);

        long long val2 = -2153156942601347941;
        string str2 = to_string(val2);

        cout << val1 << endl;
        cout << str2 << endl;

        return 0;
}
```

```
104
-2153156942601347941
```

Reference (further read): https://www.cplusplus.com/reference/string/

String Operations

- Common functions include –
  - Copy sequence of characters from string: `copy`
  - Find content in string: `find`
  - Find last occurrence of content in string: `rfind`
  - Generate substring: `substr`
  - Compare strings: `compare`

Reference (further read): https://www.cplusplus.com/reference/string/string/

String Operations: Copy

```
#include <iostream>
using namespace std;
int main ()
{
        char buffer[20];
        string str = "Test string...";
        int length = str.copy(buffer,6,5);
        buffer[length]='\0’;
        cout << "buffer contains: " << buffer << endl;
        return 0;
}
```

```
buffer contains: string
```

Note: Returns the number of characters copied to the array.

String Operations: Find

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

int main () {
        string str1 = "There are two needles in this haystack with needles.";
        string str2 = "needle";

        int found = str1.find(str2);
        if (found > 0)
                cout << "first 'needle' found at: " << found << endl;

        found = str1.find("haystack");
        if (found > 0)
                cout << "'haystack' also found at: " << found << endl;
        return 0;
}
```

```
first 'needle' found at: 14
'haystack' also found at: 30
```

Note: Returns the position of the first character of the first match.

String Operations: Substring

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

int main ()
{
        string str = "We think in generalities, but we live in details.";
        string str2 = str.substr(3,5);   // "think"

        int pos = str.find("live");      // position of "live" in str
        string str3 = str.substr(pos);   // get from "live" to the end

        cout << str2 << ' ' << str3 << endl;
        return 0;
}
```

```
think live in details.
```

String Operations: Compare

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

int main () {
        string str1 = "Apple”, str2 = "apple";
        if (str1.compare(str2) == 0)
                cout << str1 << " is " << str2 << endl;
        else
                cout << str1 << " is not " << str2 << endl;

        if (str1.compare("Apple") == 0)
                cout << str1 << " is Apple" << endl;
        else
                cout << str1 << " is not Apple" << endl;

        return 0;
}
```

```
Apple is not apple
Apple is Apple
```

String Operations: Compare

value	relation between *compared string* and *comparing string*
0	They compare equal
<0	Either the value of the first character that does not match is lower in the *compared string*, or all compared characters match but the *compared string* is shorter.
>0	Either the value of the first character that does not match is greater in the *compared string*, or all compared characters match but the *compared string* is longer.

Image Source: https://www.cplusplus.com/reference/string/string/compare/

Passing String to a Function: Pass by Value

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

void DisplayMessage(string msg)
{
        cout << msg << endl;
}
int main()
{
        string str = "This is CSC 1300–01";
        DisplayMessage(str);
        return 0;
}
```

```
??
```

Passing String to a Function: Pass by Value

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

void DisplayMessage(string msg)
{
        cout << msg << endl;
}
int main()
{
        string str = "This is CSC 1300–01";
        DisplayMessage(str);
        return 0;
}
```

```
This is CSC 1300-001
```

Passing String to a Function: Pass by Value

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

void DisplayMessage(string msg)
{
        cout << msg << endl;
}
string BuildString(string name)
{
        string str = “My name is ” + name;
        return str;
}
int main()
{
        string str = BuildString(“Ahsan”);
        DisplayMessage(str);
        return 0;
}
```

```
??
```

Passing String to a Function: Pass by Value

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

void DisplayMessage(string msg)
{
        cout << msg << endl;
}
string BuildString(string name)
{
        string str = “My name is ” + name;
        return str;
}
int main()
{
        string str = BuildString(“Ahsan”);
        DisplayMessage(str);
        return 0;
}
```

```
My name is Ahsan
```

Let’s dig deep into Strings now…

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

int main()
{
        string s = “HI!”;
        // Printing out a string
        cout << s << endl;
        return 0;
}
```

```
HI!
```

Print the address of a string

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

int main()
{
        string s = “HI!”;
        // Printing out a string
        cout << &s << endl;
        return 0;
}
```

```
0x7ff7b20ed3b0
```

Print the address of a string (in C language)

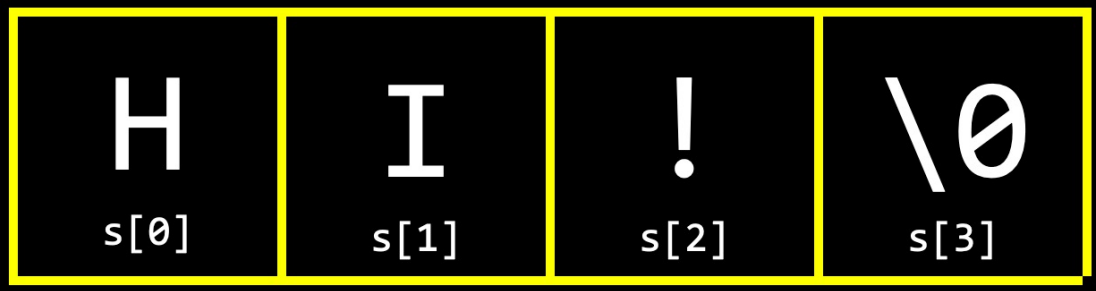
```
#include <iostream>
using namespace std;

int main()
{
        string s = “HI!”;
        // Printing out a string
        printf(“%p\n”, &s);
        return 0;
}
```

```
0x7ff7b20ed3b0
```

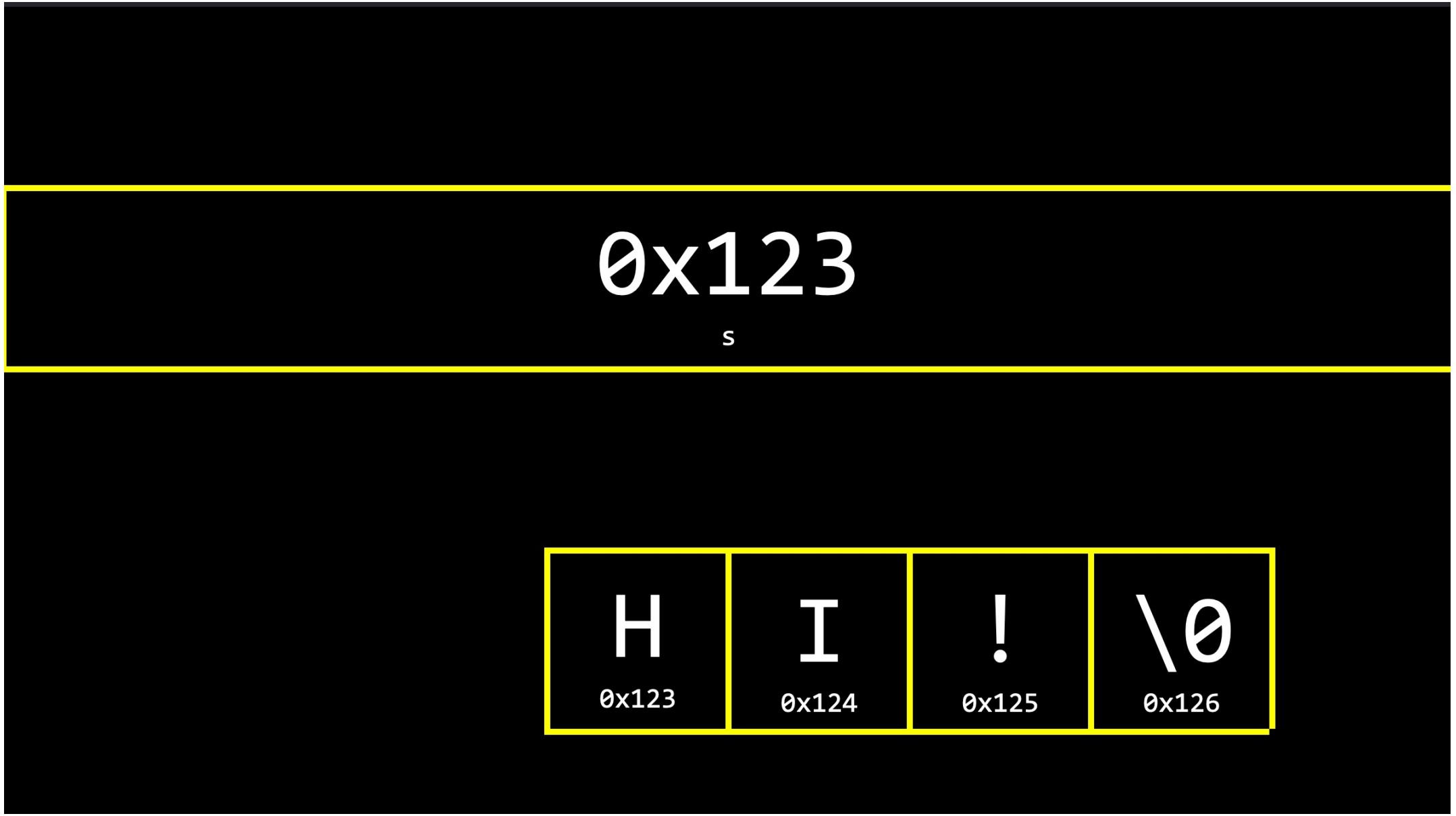
Print the addresses of all the elements (in C)

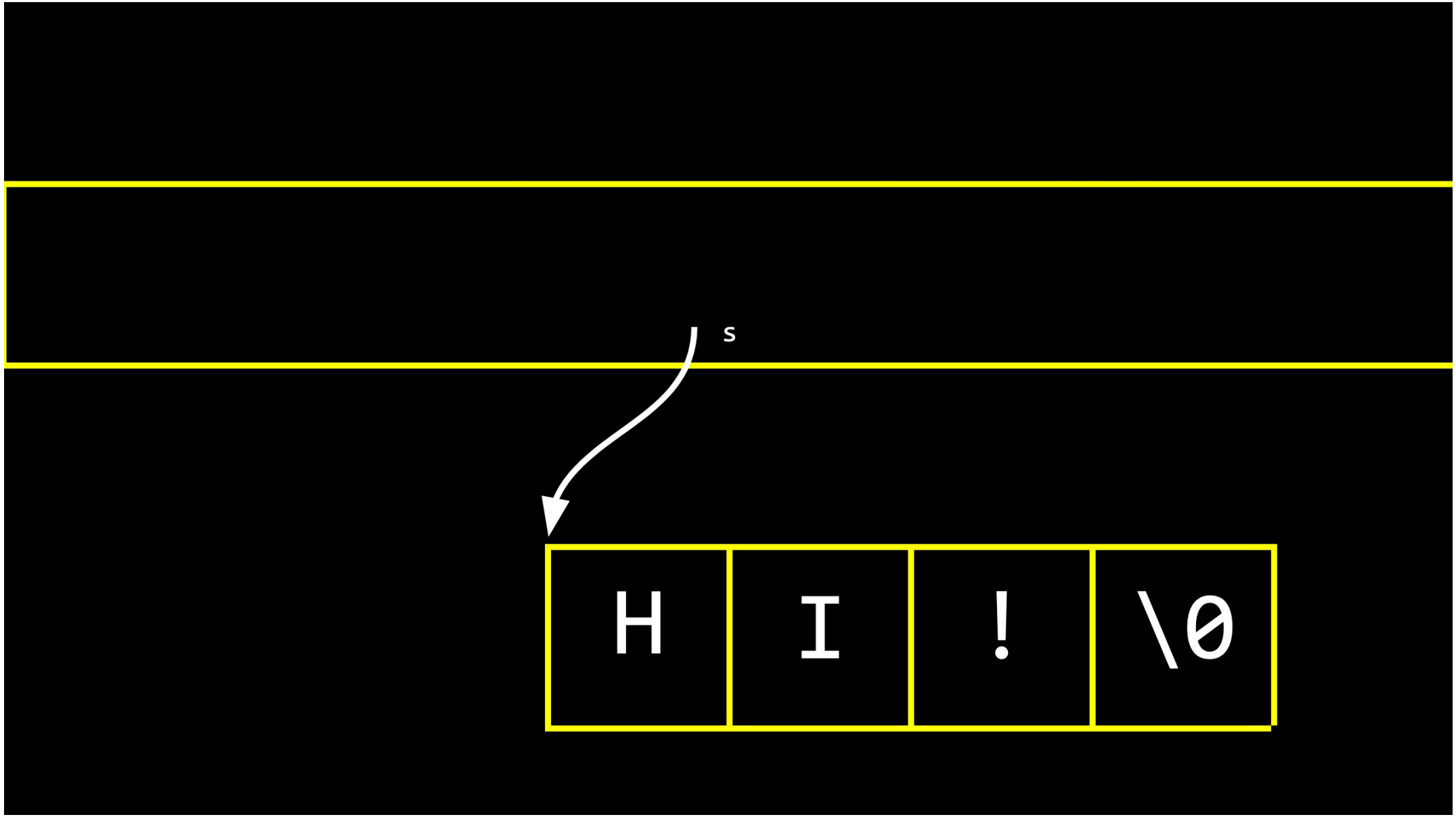
```
#include <iostream>
using namespace std;

int main()
{
        string s = “HI!”;
        // Printing out the elements’ addresses
        printf(“%p\n”, &s[0]);
        printf(“%p\n”, &s[1]);
        printf(“%p\n”, &s[2]);
        return 0;
}
```

```
0x7ff7b20ed3b1
0x7ff7b20ed3b2
0x7ff7b20ed3b3
```


H
s[0]
I
s[1]
!
s[2]
\0
s[3]


H
0x123
I
0x124
!
0x125
\0
0x126


0x123
s
H
0x123
I
0x124
!
0x125
\0
0x126


s
H
I
!
\0

String using Pointers

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

int main()
{
        char *s = “HI!”;
        // Printing out the entire string
        cout << s << endl;
        return 0;
}
```

```
HI!
```

String using Pointers

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

int main()
{
        char *s = “HI!”;
        // Printing out each element of a string
        cout << s[0] << endl;
        cout << s[1] << endl;
        cout << s[2] << endl;
        return 0;
}
```

```
H
I
!
```

String using Pointers

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

int main()
{
        char *s = “HI!”;
        // Printing out each element of a string
        cout << *(s+0) << endl;
        cout << *(s+1) << endl;
        cout << *(s+2) << endl;
        return 0;
}
```

```
H
I
!
```

Passing String to a Function: Pass by Reference

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

void CaptilizeString(string &str)
{
        str[0] = toupper(str[0]);
}

int main()
{
    string msg = "this is CSC 1300-001";
    CaptilizeString(msg);
    cout << msg << endl;
    return 0;
}
```

```
This is CSC 1300-001
```

Strings Arrays

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

int main()
{
        string words[2];
        words[0] = “HI!”;
        words[1] = “BYE!”

        cout << words[0] << endl;
        cout << words[1] << endl;

        return 0;
}
```

```
HI!
BYE!
```

Strings Arrays

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

int main()
{
        char *words[2];
        words[0] = “HI!”;
        words[1] = “BYE!”

        cout << words[0] << endl;
        cout << words[1] << endl;

        return 0;
}
```

```
HI!
BYE!
```

Command Line Argument

example.cpp

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

int main(int argc, char *argv[])
{
        for(int i = 0; i < argc; i++)
                cout << argv[i] << endl;

        return 0;
}
```

```
$
```

Command Line Argument

example.cpp

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

int main(int argc, char *argv[])
{
        for(int i = 0; i < argc; i++)
                cout << argv[i] << endl;

        return 0;
}
```

```
$ g++ -o example example.cpp
```

Command Line Argument

example.cpp

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

int main(int argc, char *argv[])
{
        for(int i = 0; i < argc; i++)
                cout << argv[i] << endl;

        return 0;
}
```

```
$ g++ -o example example.cpp
$ ./example hello hi hey hola
```

Command Line Argument

example.cpp

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

int main(int argc, char *argv[])
{
        for(int i = 0; i < argc; i++)
                cout << argv[i] << endl;

        return 0;
}
```

```
$ g++ -o example example.cpp
$ ./example hello hi hey hola
./example
hello
hi
hey
hola
```

Command Line Argument

example.cpp

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

int main(int argc, char *argv[])
{
        for(int i = 1; i < argc; i++)
                cout << argv[i] << endl;

        return 0;
}
```

```
$ g++ -o example example.cpp
$ ./example hello hi hey hola
??
```

Command Line Argument

example.cpp

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

int main(int argc, char *argv[])
{
        for(int i = 1; i < argc; i++)
                cout << argv[i] << endl;

        return 0;
}
```

```
$ g++ -o example example.cpp
$ ./example hello hi hey hola
hello
hi
hey
hola
```

Remarks

- Reference
  - MIT 6.096 Introduction to C++
  - This is CS50x, Dr. David J. Malan. https://cs50.harvard.edu/x/2020/