Lecture 2 - Arrays

Compiling

Encryption is the act of hiding plain text from prying eyes. Decrypting, then, is the act of taking an encrypted piece of text and returning it to a human-readable form. An encrypted piece of text may look like the above.

For example, compiling this...

```c
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

- VS Code utilizes a compiler called clang or c language.
- If you were to type make hello, it runs a command that executes clang to create an output file that you can run as a user.
- VS Code has been pre-programmed such that make will run numerous command line arguments along with clang for convenience as a user.
- To use cs50.h library, you should run clang -o hello hello.c -lcs50 to compile hello.c.

Compiling involves major steps, including the following:

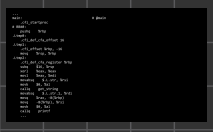
1 Preprocessing

```c
...
string get_string(string prompt);
int printf(string format, ...);
...

int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

The header files in your code are effectively copied into your file (Ex. #include <cs50.h>).

2 Compiling

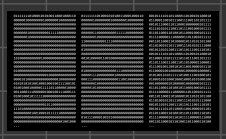


Your program is converted into assembly code.

3 Assembling

The compiler converts your assembly code into machine code.

4 Linking



Code from your included libraries are converted also into machine code and combined with your code. The final executable file is then outputted.

Debugging

- VS Code has a built-in debugger.
- In the CS50 codespace, the debugger has been preconfigured to you

1. Set a breakpoint by clicking to the left of a line of your code. When you click you'll see a red dot (•).
2. Run debug50 ./<programname>, replacing <programname> with you compiled code name.
3. At the top of your window, you can click the step over button to move through your code.

Arrays

Each data type requires a certain amount of system resources:

- bool → 1 byte
- int → 4 bytes
- long → 8 bytes
- float → 4 bytes
- double → 8 bytes
- char → 1 byte
- string → ? bytes

Running an example...

```c
int score1 = 72;
int score2 = 73;
int score3 = 33;
```

72 scores	73 scores			
33 scores				

Each int is using 4 bytes of memory to store each score value.

The solution...

Arrays are a way of storing data back-to-back in memory such that this data is easily accessible.

```c
int scores[3];
scores[0] = 72;
scores[1] = 73;
scores[2] = 33;
```

→ it tells the compiler to provide three back-to-back places in memory of size int (4 bytes) to store three scores.

This examines the value at this location by indexing into the array called scores at location 0, for example, to see what value is stored there.

Strings

A string is simply an array of variables of type char.

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

A string is an array of characters that begins with the first character and ends with a special character called a NUL character.

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

Imagining this in decimal, your array would look like above.

If we execute this code, the two strings can be visualize as follow:

```c
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    string s = "HI!";
    string t = "BYE!";

    printf("%s\n", s);
    printf("%s\n", t);
}
```

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]							

Useful libraries

- string.h
- ctype.h

Command-Line Arguments

- Are those arguments that are passed to your program at the command line.
- For example, all those statements you typed after clang are considered command line arguments.

```c
#include <cs50.h>
#include <stdio.h>

int main(int argc, string argv[])
{
    if (argc == 2)
    {
        printf("hello, %s\n", argv[1]);
    }
    {
        printf("hello, world\n");
    }
}
```

argc: the number of command line arguments

argv[]: An array of the characters passed as arguments at the C.L.I.

Using the syntax of this program, executing the following: ./great David would result in the program saying hello, David.

Exit Status

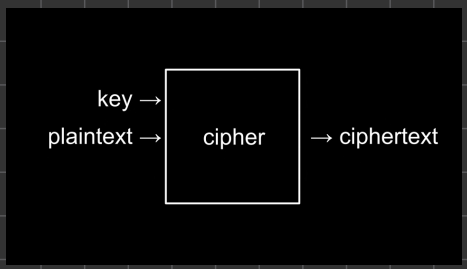
- When a program ends, a special exit code is provided to the computer.
- When a program exits without error, a status code of 0 is provided the computer.
- Often, when an error occurs that results in the program ending, a status of 1 is provided by the computer.

```c
int main(int argc, string argv[])
{
    if (argc != 2)
    {
        printf("Missing command-line argument\n");
        return 1;
    }
    else
    {
        printf("hello, %s\n", argv[1]);
        return 0;
    }
}
```

return 1; → error

return 0; → success

Cryptography



Cryptography is the art of ciphering and deciphering a message.

plaintext and a Key are provided to a cipher, resulting in a ciphered text.

The Key is a special argument passed to the cipher along with the plaintext. The cipher uses the Key to make decisions about how to implement its cipher algorithm.