

# What is a program?

## Instructions

A set of commands that tells the computer what actions to perform.

Consider a well known example: PhotoShop. It has lots of features for working with images. It has sets of instructions to load and display an image, and perform many transformations of an images such as: crop an image, resize an image, change the contrast of an image, etc. Each of these features is performed by a set of instructions for the computers CPU.

## Data

The input material that the instructions work with and transform.

In the case of PhotoShop, the data consists of an image, or images, the input from the user via the keyboard and mouse, and other inputs such as metadata about the image and its source, information about the color space of the computer display, and other information.

## Instructions

The instructions that a CPU understands are very simple.

ADD	Add two registers
B	Branch (on condition code)
CMP	Compare
LDR	Load register from memory
MOV	Move register or constant
MUL	Multiply
STR	Store register to memory
SUB	Subtract

It is difficult, time consuming, and error prone to program a cpu directly using its native instruction set.

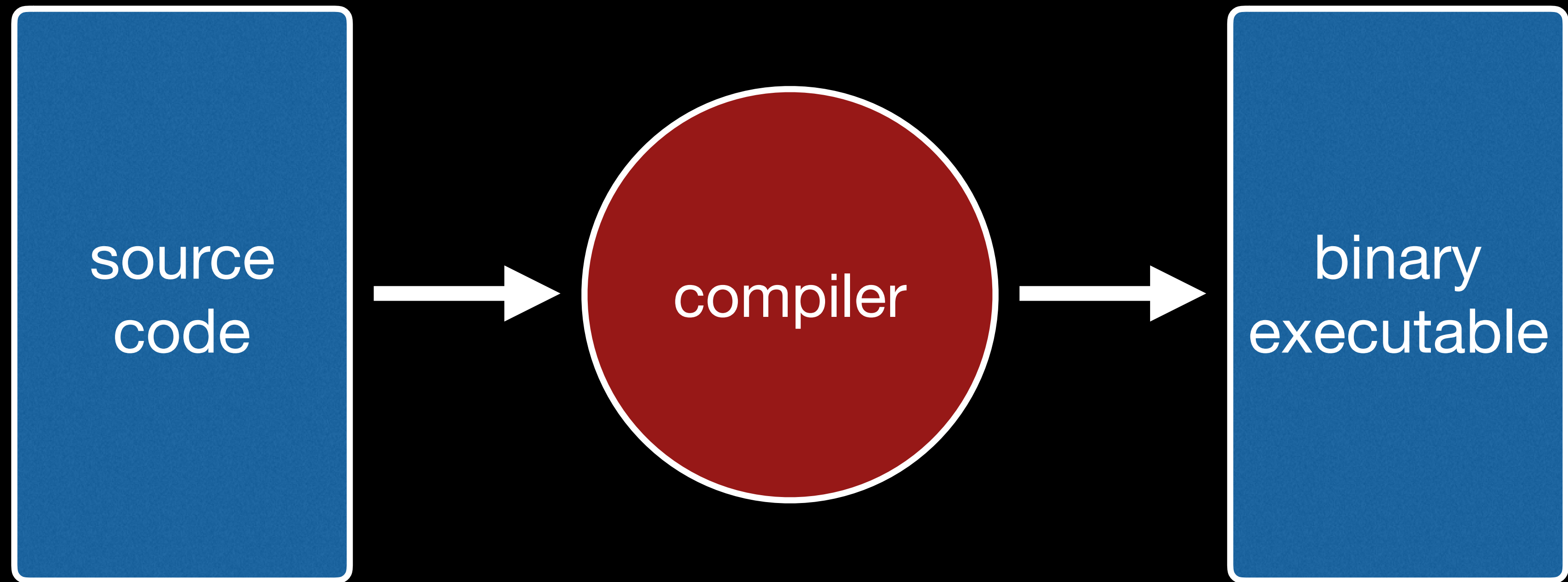
Instead of writing code in a native instruction set for an individual cpu, most programmers use a higher level programming language.

**Swift**



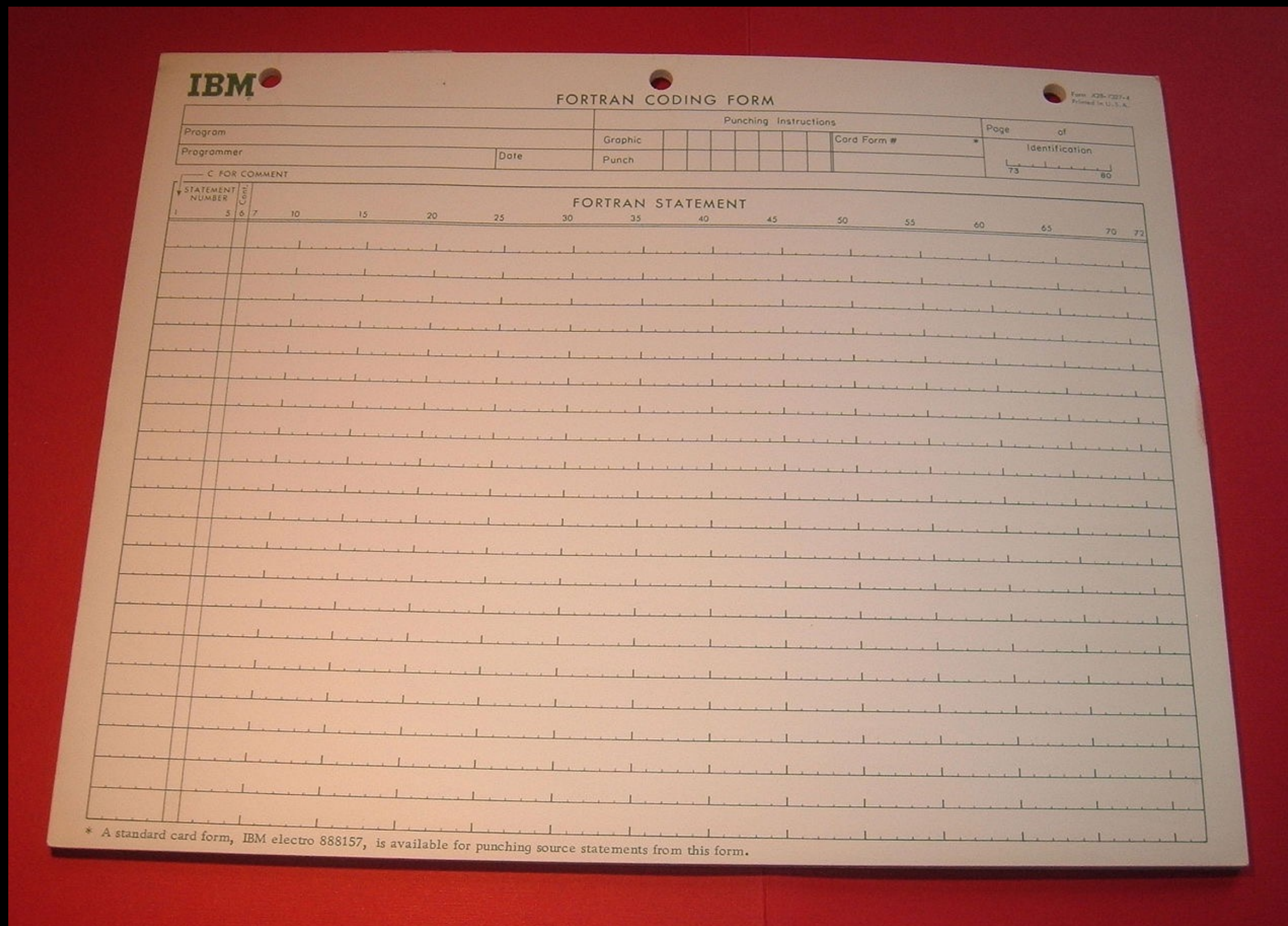
We will use Swift.

A compiler is a special program which is used to translate source code into native instructions for a cpu. (eg. a binary executable.)





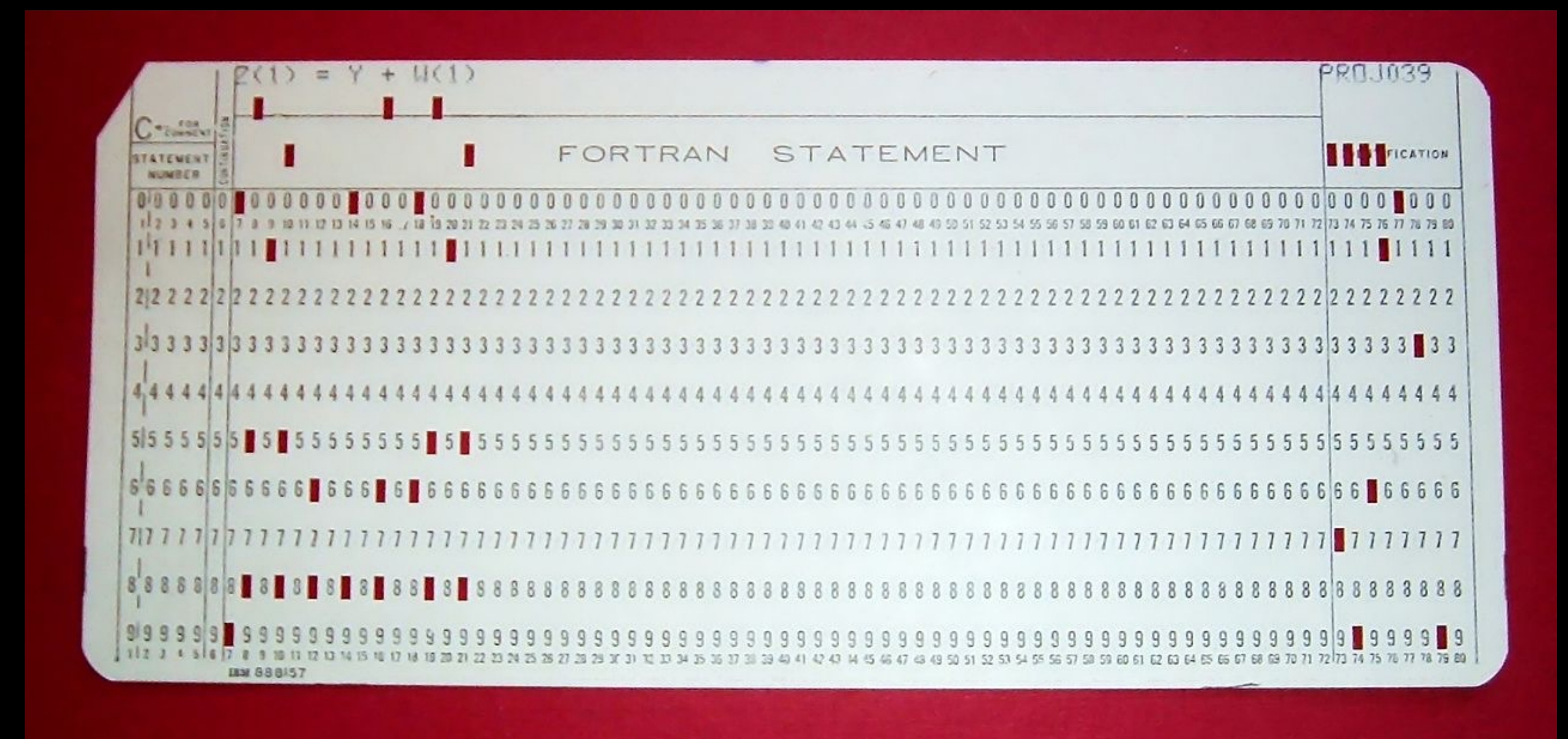
# Programming during the dawn of computing



The image shows a blank IBM Fortran Coding Form. At the top left is the IBM logo. The title "FORTRAN CODING FORM" is centered. Below the title are fields for "Program", "Punching Instructions", "Page of", "Programmer", "Date", "Punch", "Card Form #", and "Identification". The main body of the form is a large grid with columns numbered 1 to 72. The first column is labeled "STATEMENT NUMBER" and the rest are labeled "FORTRAN STATEMENT". A small note at the bottom left states: "\* A standard card form, IBM electro 888157, is available for punching source statements from this form."

coding form

Write your program by hand here first.



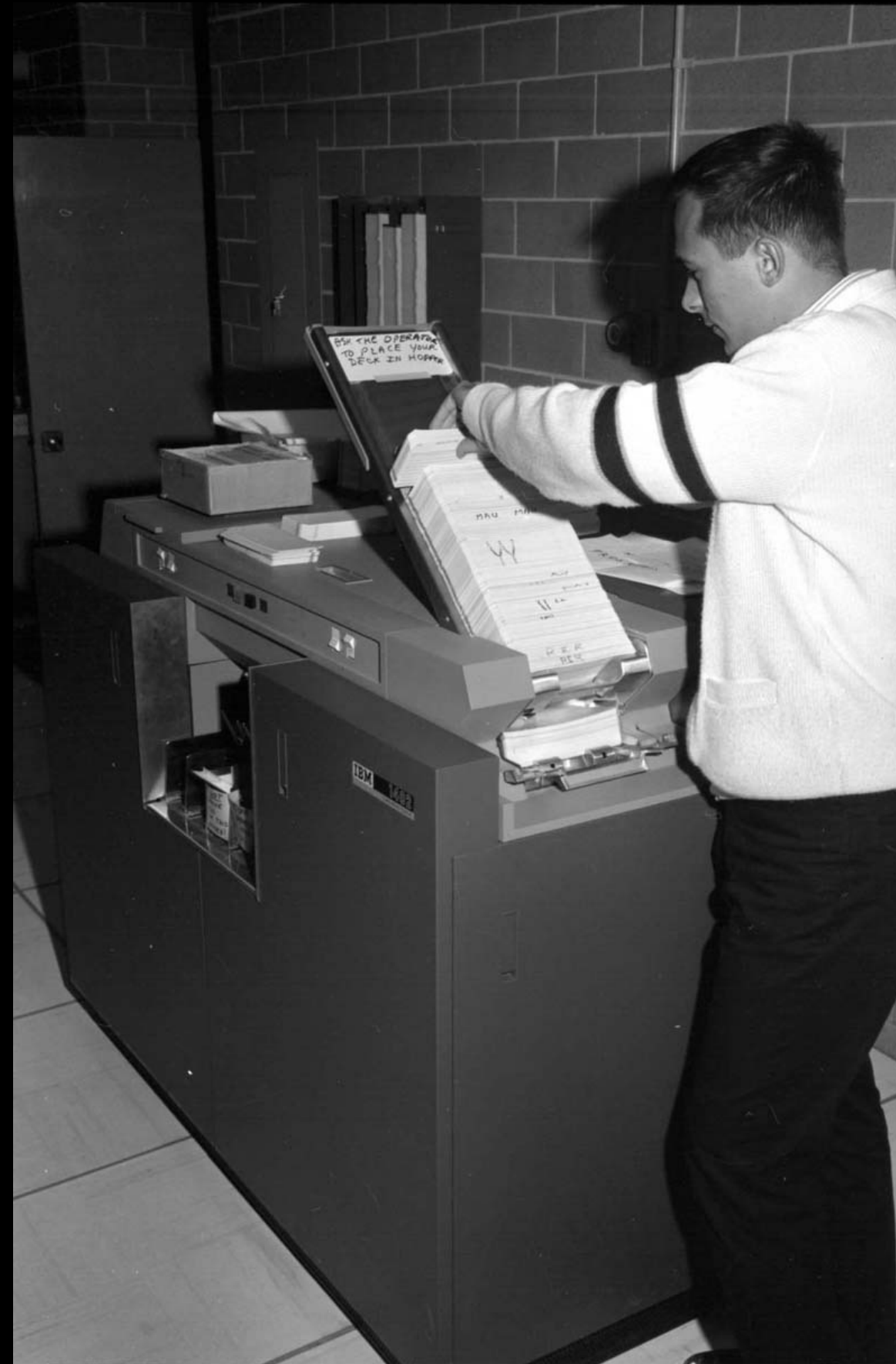
The image shows a punched Fortran card. At the top left is the IBM logo. The title "FORTRAN STATEMENT" is centered. Below the title are fields for "STATEMENT NUMBER" and "IDENTIFICATION". The main body of the card is a grid with columns numbered 1 to 80. The first column is labeled "STATEMENT NUMBER" and the rest are labeled "FORTRAN STATEMENT". The card contains the following Fortran code:  
$$Z(1) = Y + U(1)$$
  
The card is punched with holes corresponding to the characters in the code. A small note at the bottom left states: "IBM 888157".

punch card

Translated from coding form to holes in card



# Programming during the dawn of computing



card reader

Stack all cards in reader to execute program.

# Programming during the dark early days of computing

```
#include <stdio.h>
#include <stdlib.h>
int main( int argc, char **argv ) {

    printf("Hello world.\n");
    exit(0);
}
```

Use a text editor (vi) to create and edit source code file.

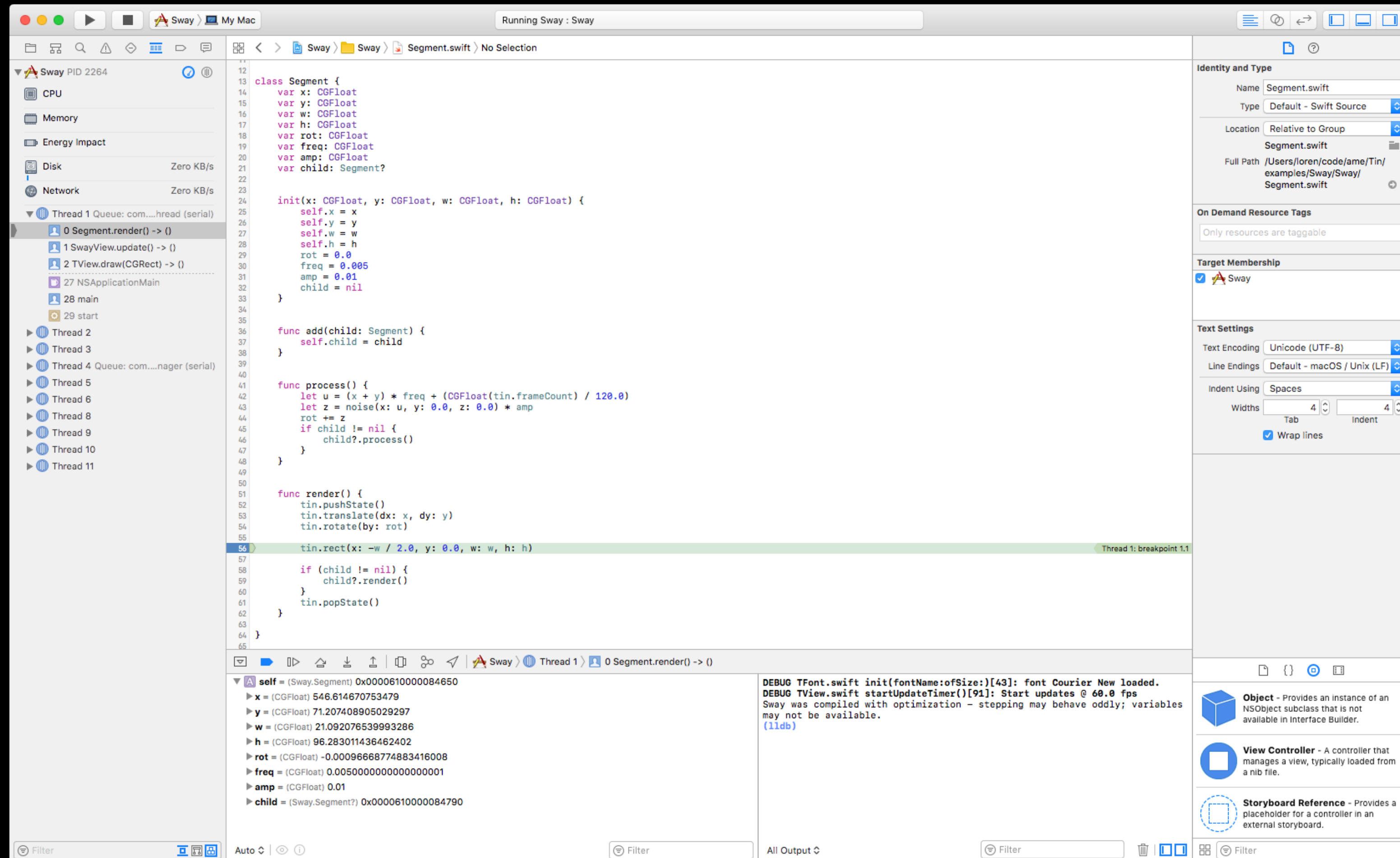
```
cc hello.c -o hello
```

In terminal, type command to compile and link program to create binary executable.

```
./hello
```

In terminal, type command to run and test program.

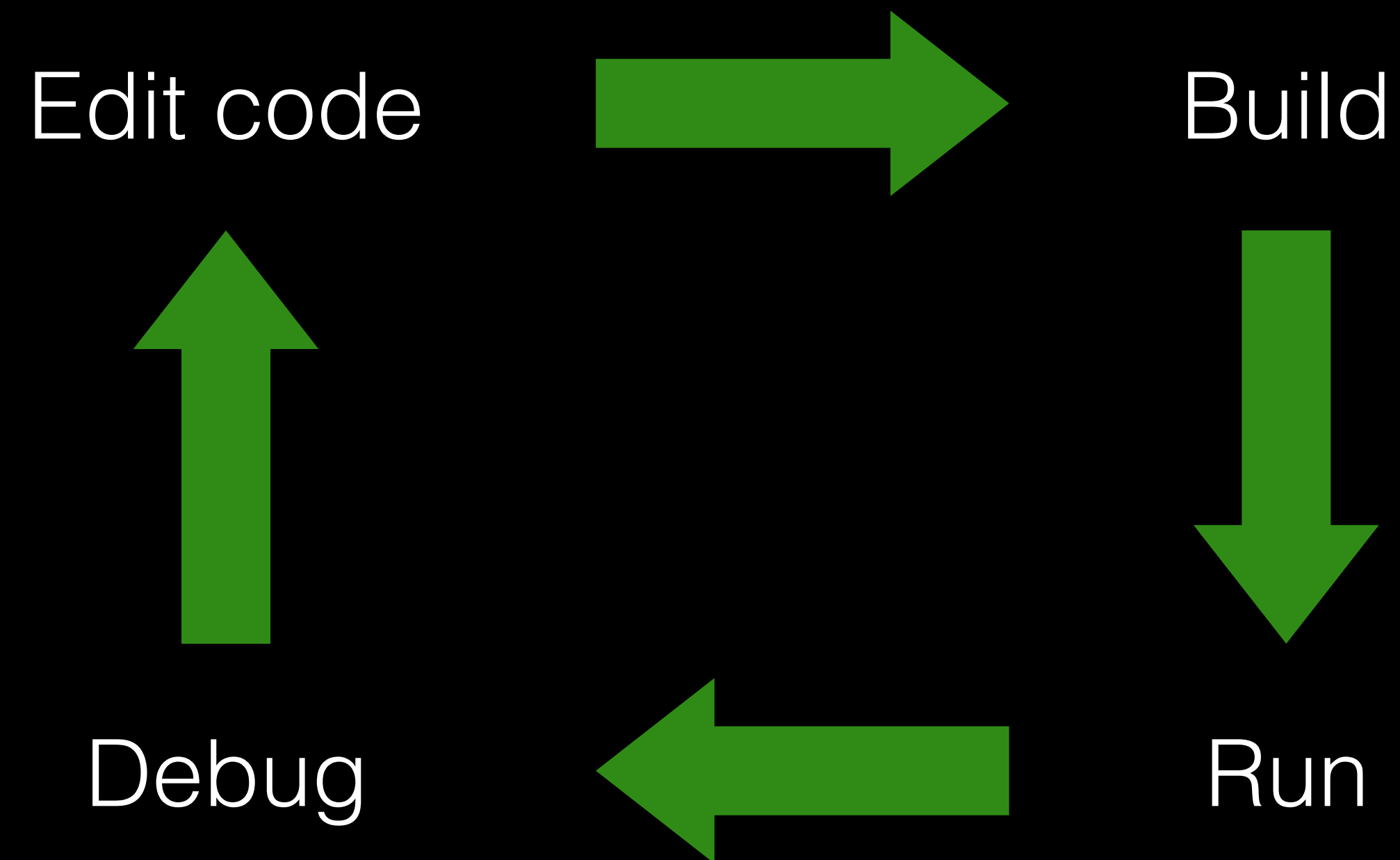
# Modern Integrated Development Environment (IDE)



Xcode



# Modern Integrated Development Environment (IDE)



Development cycle

*Programmers want this cycle to be easy, and fast!*