Subprograms in Turing

Introduction

What is a Program?

A program is a collection of <u>useful actions</u> that a computer performs to solve a problem (or several problems).

It is up to the programmer to choose the actions, put them into the correct order, and implement them using the appropriate language.

As problems get more complicated, programs contain more actions, becoming longer and more difficult to manage.

What is a Subprogram?

A <u>subprogram</u> is a program within a program. You may also hear terms such as <u>functions</u> and <u>procedures</u>, which are special types of subprograms.

There are also utility subprograms, such as those for input, output, and specialized math operations. Most of these are <u>functions</u>.

We will start with <u>procedures</u>.

Predefined Subprograms

- most programming languages provide many subprograms for common (and not-so-common) purposes

- for example:

```
x := sqrt(5) % x is 2.236068

y := length("Hello") % y is 5

randint(roll1, 1, 6)
```

Turing Structure with a Procedure
(must be above main part of program)

```
% Turing program to print stuff

procedure printStuff()
    put "This is stuff!"
end printStuff

%%% main program below %%%
printStuff()
printStuff()
printStuff()
```

Procedures as Replacement Code

The simplest version of a procedure is essentially replacing sections of code with the name of the procedure. For simple programs, this may actually be more work. For larger and more complex programs, procedures and subprograms become invaluable tools.

A procedure must be <u>declared</u> before we can use it. It is also important to note that declaring a procedure does not do anything on its own. It must be <u>called</u> from the main program for the code to be run.