The **focus** of this homework assignment is learning more of RISC-V assembly language programming and, in particular, implementing procedures (functions in C) using RISC-V ISA features.

```c
#include <stdio.h>
#include <stdint.h>      // for uint32_t
#include <inttypes.h>    // for PRIu32 in printf

// "Half or triple-plus-one" sequence, iterative.
uint32_t hotpo_i(uint32_t n) {
  printf("%" PRIu32 "\n", n);
  while (n > 1) {
    if (n % 2 == 0) n /= 2;
    else n = 3 * n + 1;
    printf("%" PRIu32 "\n", n);
  }
  return n;
}

// "Half or triple-plus-one" sequence, recursive.
uint32_t hotpo_r(uint32_t n) {
  static uint32_t recur_depth = 1;  // recursion depth
  uint32_t next_n;
  printf("%" PRIu32 "\t%" PRIu32 "\n", n, recur_depth);
  if (n == 1) return 1;
  recur_depth++;
  if (n % 2 == 0) next_n = hotpo_r(n/2);
  else next_n = hotpo_r(3*n + 1);
  recur_depth--;
  return next_n;
}

// Read an unsigned integer from stdin and invoke both versions of
// hotpo on it.
int main() {
  uint32_t init_n;
  scanf("%" PRIu32, &init_n);
  hotpo_i(init_n);
  putchar('\n');
  hotpo_r(init_n);
  return 0;
}
```

As did the previous assignments, this one will also use the *RARSM*[1] environment. The most closely related portions of the textbooks are Chapter 5 of the *RVAP*[2] book and Section

[1] Jean Privat and others, RARSM—RISC-V Assembler and Runtime Simulator (iMproved), `https://github.com/rarsm/rars`, 2024.

[2] Robert Winkler, *RISC-V Assembly Programming* (Robert Winkler, 2024).

2.8 of the *COAD*[3] book.

The **main programming task** to write an assembly language program corresponding to the above C program (using discussions in class and the discussion forum to clarify details of the correspondence).

Input-output: Mirroring this aspect of the previous assignment, the `hw03` program should read its input from the *standard input* stream and write its output to the *standard output* stream. Optional diagnostics may be written to the *standard error* stream. It is very important that the program read its input only from the standard input stream and that it write nothing except the specified output to the standard output stream.

The **packaging and submission** requirements are similar to those in the previous homework assignments, using the `hw03` tag instead of `hw02`. Similarly, the rules for using additional **resources** are the same as before.

[3]David A Patterson and John L Hennessy, *Computer Organization and Design RISC-V Edition*, 2nd edition (Morgan Kaufmann, 2020).