Consider the program below. What would be the minimum number of registers necessary to compile this program?

```
int foo(int a, int b) {
  c = 4 * a;
  d = b - c;
  e = d / c;
  a = 4 * d;
  x = a * e;
  return x;
}
```

Figure 1: A straight-line piece of code that perform simple arithmetic operations.

1. Which algorithm did you use to find your answer?

2. Assuming that you must deal with programs without loops, but involving any sequence of assignments, do you think that this question has polynomial-time solution?

3. Would your answer change if you are allowed to change the name of variables?

4. Would your answer change if you were allowed to move instructions around?

UFMG

DCC

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Compilers Laboratory

```
int foo(int a, int b) {
  c = 4 * a;
  int f = c % b;
  d = b - c;
  e = d / c;
  a = f * d;
  x = a * e;
  return x;
}
```

Figure 2: A slight variation of the program seen in Fig-1

5. What about this new program in Fig-2: how many registers would be necessary to compile it without having to map variables to memory?

6. Assuming that you have only two registers: R0 and R1, could you rewrite the program in Fig-2 in a format that is closer to machine instructions? You can only use the two registers, and instructions to access memory. To this effect, assume that you have access to:
   6-a: "store st[n] R", which stores register R into memory location st[n], given positive integer n.
   6-b: "load st[n] R", which loads the value stored in st[n] into register R, given positive integer n.