Universidade Federal de Viçosa
Centro de Ciências Exatas e Tecnológicas
Departamento de Informática
INF340 – Teoria da Computação
12/06/2018

Trabalho Prático 3

Entregar até 26/06/18

Verifique, pela lógica de Hoare, a correção (parcial) das seguintes funções. Quando não houver a pré-condição inicial, tente determinar a mais fraca possível para que se obtenha a pós-condição final dada. Escreva, nas lacunas dos comentários, as asserções válidas, usando (a linguagem da) lógica de primeira ordem.

1. Cálculo do máximo divisor comum, onde o invariante do laço principal é dado por $I \equiv \text{mdc}(a, b) = \text{mdc}(x, y)$.

```
int mdc(int x, int y)
{
    /* (x > 0) ∧ (y > 0) */
    a = x;
    /* ____________________________________________ */
    b = y;
    /* ____________________________________________ */
    while (a != b)
    {
        /* ____________________________________________ */
        if (a > b)
            /* ____________________________________________ */
            a = a - b;
            /* ____________________________________________ */
        else
            /* ____________________________________________ */
            b = b - a;
    }
    /* ____________________________________________ */
    return a;
    /* a = b = mdc(x, y) */
}
```

Sugestão: Talvez você precisará das seguintes relações conhecidas do mdc:

$$\text{mdc}(u, v) = \text{mdc}(u - v, v), \text{ se } u > v$$
$$\text{mdc}(u, v) = \text{mdc}(v, u)$$
$$\text{mdc}(u, u) = u$$

2. Produto eficiente de números inteiros, onde o invariante do laço principal é dado por $I \equiv (z + u * v = x * y) \wedge (u \geq 0)$. O predicado odd$(u)$ é satisfeito se u for um número ímpar.

```
int prod(int x, int y)
{
   /* ____________________________________________________ */
   int u, v, z;
   /* ____________________________________________________ */
   u = x;
   /* ____________________________________________________ */
   v = y;
   /* ____________________________________________________ */
   z = 0;
   /* ____________________________________________________ */
   while (u != 0)
   {
      /* ____________________________________________________ */
      if (odd(u))
         /* ____________________________________________________ */
         z = z + v;
      /* ____________________________________________________ */
      u = u / 2;
      /* ____________________________________________________ */
      v = 2 * v;
      /* ____________________________________________________ */
   }
   /* ____________________________________________________ */
   return z;
   /* z = x * y */
}
```