



1. (3v) Considere que temos as seguintes declarações:

```
1  int i = 2, j = 3, k = 4, l, m, n;  
2  double x = 12.3, y = 3.67, v, w;
```

Partindo do princípio que cada instrução é a única a ser executada após as declarações (p/ex. a alínea c) é feita independentemente das alíneas a) e b)), indique o valor que é colocado em cada variável abaixo, ou explique porque é que a afetação é inválida, caso seja esse o caso.

- (a) $v = x / i$;
- (b) $w = (\text{int}) y \% k$;
- (c) $n = (\text{int}) x / (\text{long}) y * i / 2$;
- (d) $m = n + i * j$;
- (e) $n = k / (j * i) * x + y$;
- (f) $i = i + 1$;

2. (6v) Usando ciclos for, escreva código capaz de *calcular* os seguintes valores acumulados, com um máximo de 5 linhas de código para cada troço:

- (a) $1 + 3 + 5 + \dots + 99$
- (b) $1 - 2 + 3 - \dots + 49 - 50$
- (c) $1.0 * 1.1 * 1.2 * \dots * 2.0$
- (d) $1.0/2 + 1.0/4 + 1.0/8 + \dots + 1.0/256$

Assuma que o resultado deve ser colocado numa variável chamada *res*, que já terá sido apropriadamente declarada mas que tem de inicializar, e não faça nenhum Input/Output. Se precisar, pode usar outras famílias de instruções de controle como if-else.

3. (4v) Repita o grupo 2 (todas as alíneas), usando ciclos while.
4. (3v) Considere a definição de número primo, em que um inteiro N é um número primo se só for divisível por 1 e por si mesmo. Defina um método (de classe) boolean primo (int n) que retorna true se n for primo e false caso contrário.
5. (4v) De igual modo, diz-se que dois números são primos entre si se o seu maior divisor comum for 1. Comente, interprete e escreva um método (de classe) boolean primos_entre_si (int a, int b) que implemente esta propriedade.