

Funções — <tipo> <nome> (<lista argumentos>) {I}

32 Analise os seguintes programas e preveja a sua saída:

1.

```
#include <stdio.h>

int dobro(int a) {
    return(2*a);
}

int main() {
    int n;
    printf("Introduza um inteiro: ");
    scanf("%d",&n);
    printf("O dobro de %2d é %d\n", n, dobro(n));

    return(0);
}
```

2.

```
#include <stdio.h>

int trocar(float* a, float* b) {
    float aux;

    aux = *a;
    *a = *b;
    *b = aux;
}

int main() {
    float x = 2.5, y = 0.5;
    printf("x=%4.1f y=%4.1f\n", x, y);
    trocar(&x, &y);
    printf("x=%4.1f y=%4.1f\n", x, y);

    return(0);
}
```

3.

```
#include <stdio.h>
#include <math.h>

void lerDados(float* base, int* expoente) {
    printf("Escreva a base: ");
    scanf("%f", base);
    printf("Escreva o expoente: ");
    scanf("%d", expoente);
}

float potenciaInteira(float b, int e) {
    return(pow(b, e));
}
```

```

}

void escreveResultado(int e, float b, float r) {
    printf("Resultado: %4.2f^%d=%4.2f\n", b, e, r);
}

int main () {
    float b, r;
    int e;

    lerDados(&b, &e);
    r = potenciaInteira(b, e);
    escreveResultado(e, b, r);
}

```

33 Escreva um procedimento que troque, entre si, o valor de duas variáveis, mas sem usar uma variável auxiliar (só com as operações aritméticas elementares).

34 Implemente as seguintes funções:

1. $\log : \mathbb{R}^+ \times A \rightarrow \mathbb{R}$
 $(x, a) \mapsto \log_a(x) = \frac{\ln(x)}{\ln(a)}$
 com $A = \{x \in \mathbb{R} : x > 0 \text{ e } x \neq 1\}$
2. $\sinh : \mathbb{R} \rightarrow \mathbb{R}$
 $x \mapsto \sinh(x) = \frac{e^x - e^{-x}}{2}$
3. $f : \mathbb{R} \rightarrow \mathbb{R}$
 $x \mapsto \begin{cases} \sqrt{x} & \text{se } x \geq 0 \\ \sqrt{-x} & \text{se } x < 0 \end{cases}$
4. $\operatorname{argsech} :]0, 1] \rightarrow \mathbb{R}$
 $x \mapsto \operatorname{argsech}(x) = \log\left(\frac{1+\sqrt{1-x^2}}{x}\right)$

- 35**
1. Dados dois números inteiros positivos escreva um sub-programa que verifique se os dois números são ou não amigos, isto é, se cada um deles é igual à soma dos divisores próprios do outro.
 2. Usando a alínea anterior faça um programa que escreva todos os pares de números amigos existentes entre 1 e 1000. Tente otimizar este programa.