

Exercícios — Programação C e Estruturas de Dados

1. Escreva um programa C que imprima o seu nome e os parâmetros de linha de comando, um por linha, na ordem inversa. Exemplo de execução:

```
$ ./invcmd um 2 tres 4
Programa: ./invcmd
Parametros:
4
tres
2
um
```

2. Escreva um programa C que receba um inteiro como parâmetro na linha de comando, e mostre quantos bits 1 tem a representação binária desse número. Exemplos de execução:

```
$ ./contabits 20
O numero 20 possui 2 bits em 1.

$ ./contabits 65535
O numero 65535 possui 16 bits em 1.
```

3. Escreva um programa C que leia uma sequência de números inteiros da entrada padrão e calcule a média (μ) e o desvio padrão (σ) desses números, conforme as expressões abaixo (onde x_i representa cada número lido e N a quantidade de números lidos):

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \qquad \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

A leitura deve prosseguir até que o número lido seja -1 (que não deve ser considerado no cálculo).

O seu programa não deve assumir nenhum limite máximo de números a serem lidos, e não deve fazer uso de `realloc()`.

Exemplo de execução:

```
$ ./media-desv
3
2
1
7
4
-1
Media = 3.40
Desvio padrao = 2.06
```

4. Escreva um programa C que leia uma sequência de números inteiros da entrada padrão e armazene esses números em uma fila circular. O tamanho da fila circular é determinado por um parâmetro de linha de comando. Sempre que o número lido for -2 , o programa deve imprimir o conteúdo da fila. A leitura deve prosseguir até que o número lido seja -1 .

Exemplos de execução:

\$./filacirc 3	\$./filacirc 4
2	3
6	2
4	-2
3	Fila: 3 2
7	1
5	7
5	4
-2	-2
Fila: 7 5 5	Fila: 2 1 7 4
-1	-1

5. Escreva um programa C que leia da entrada padrão uma série de pares nome prioridade, onde nome é uma *string* e prioridade um valor numérico inteiro maior que zero. O programa deve manter todos os nomes com a mesma prioridade em uma fila, sendo que o número máximo de prioridades distintas não é conhecido de antemão.

Quando nome for IGUAL, o programa deve imprimir todos os nomes com prioridade igual a prioridade, na ordem em que apareceram na entrada. Quando nome for MENIG, o programa deve imprimir todos os nomes com prioridades inferiores ou iguais a prioridade; os nomes devem ser listados em ordem de prioridade e, dentro de cada prioridade, em ordem de chegada. Quando nome for FIM, a execução deve ser encerrada.

Na solução, o tempo gasto para listar um conjunto de nomes deve ser proporcional à quantidade de nomes **com as prioridades em questão**. Isso significa que soluções baseadas em uma única fila contendo todas as prioridades não são aceitáveis.

Exemplo de execução:

```
$ ./prioridades
A 2
B 1
C 4
D 2
E 84
F 1
IGUAL 2
Prioridade 2: A D
G 2
H 747
I 1048576
MENIG 3
Prioridade 1: B F
Prioridade 2: A D G
FIM
```