

## Trabalho prático sobre Algoritmos de Escalonamento de CPU

Considere os seguintes algoritmos de escalonamento de CPU:

1. Prioridades dinâmicas
  - a. A cada processo é atribuída uma mesma prioridade de escalonamento inicial que é alterada com o passar do tempo.
  - b. Se o processo não está no estado executando, o escalonador periodicamente aumenta a sua prioridade.
  - c. Quanto mais o processo recebe a posse da CPU mais o escalonador reduz a sua prioridade.
  - d. As prioridades são alteradas a cada instante de tempo.
  - e. O escalonador sempre seleciona o processo com a prioridade mais alta dentre aqueles no estado pronto
  - f. Processos com mesma prioridade, devem seguir a ordem de chegada na fila.
  - g. O escalonador DEVE forçar a preempção do processo em execução sempre que houver um processo de maior prioridade.
2. Loteria
  - a. Um valor entre 0 e n-1 é atribuído a cada processo, onde n é o número de processos.
  - b. Sorteia-se um valor no intervalo.
  - c. O processo escolhido é colocado em execução por um quantum.
  - d. Ao final de cada quantum, o processo é repetido.
3. Alternância Circular(Round Robin com quantum=2)
  - a. Todos os processos são armazenados em uma fila circular.
  - b. A cada processo é atribuído uma unidade de tempo chamada quantum.
  - c. OBS: Considere a fila ordenada pelo ID de processos com mesmo instante de chegada.

Desenvolva um programa que implemente estes 3(três) algoritmos e leia da entrada padrão uma lista de processos com seus respectivos tempos de chegada e de duração. O programa deverá imprimir na saída padrão uma tabela com os seguintes valores:

- **Tempo de retorno médio** – Refere-se ao tempo transcorrido entre o momento da entrada do processo no sistema e o seu término.
- **Tempo de resposta médio** – Intervalo de tempo entre a chegada do processo e o início de sua execução.
- **Tempo de espera médio** – Soma dos períodos em que um processo estava no seu estado pronto.

A entrada é composta por uma série de pares de números inteiros separadas por um espaço em branco indicando o instante de chegada do processo e a duração de cada processo.

Exemplo de entrada:

```
0 2
0 3
1 2
1 4
```

A saída é composta por linhas contendo a sigla de cada um dos três algoritmos e os valores das três métricas solicitadas. Cada linha apresenta a sigla do algoritmo e os valores médios (com uma casa decimal) para tempo de retorno, tempo de resposta e tempo de espera, exatamente nesta ordem, separados por um espaço em branco.

Exemplo de saída:

```
PRI 7.50 1.00 4.75
LOT 7.00 2.50 4.25
RR 6.50 2.50 3.75
```

**Importante:** A corretude do seu programa será testada automaticamente. Portanto, o programa deve ler a entrada e imprimir a saída EXATAMENTE como no formato especificado acima. A impressão de qualquer outro caractere ou de dados fora da ordem solicitada implicará em uma saída incorreta.

**DICA:** Lendo da entrada padrão em C. O exemplo a seguir permite a execução com redirecionamento da entrada no formato: `./executavel<file.txt`

```
#include <stdio.h>

#define BUFFERSIZE 100

int main (int argc, char *argv[])
{
    char buffer[BUFFERSIZE];
    fgets(buffer, BUFFERSIZE , stdin);
    printf("Read: %s", buffer);
    return 0;
}
```