



Programação 2

2º Semestre

Trabalho Prático P3B

Efectue as tarefas de programação descritas abaixo, usando a linguagem C++ em ambiente Linux.

Grupo 7

Uma estrada, com uma faixa de rodagem, termina em uma intersecção controlada por um semáforo. Durante o tempo de luz verde, os veículos deixam a estrada, na ordem de chegada. Pretende-se implementar um programa em C++ que ilustre, de forma simplificada, este cenário.

```
class Veiculo
{
public:
    int comprimento;
    double velocidade;
    Veiculo(int c, double v): comprimento(c), velocidade(v){}
};

class ViaSemaforizada
{
    queue<Veiculo> veiculos;
    int tempoDeVerde;
    int tempoParaProximo(Veiculo v);
public:
    void setTempoDeVerde(int tempo){tempoDeVerde = tempo;}
    void adicionarVeiculo(Veiculo v);
    void abrirSemaforo();
    double densidadeInstantanea();
    double fluxoSaturacao();
    void print()
    {
        for(int i=0;i<veiculos.size();i++) cout << "."; cout << endl;
    }
};
```

a) Implemente o membro-função

```
void ViaSemaforizada::adicionarVeiculo(Veiculo v)
```

Que insere um novo veículo na via. Todos os veículos da via são guardados na fila *veiculos*, a fim de se manter a ordem de chega e saída. Implemente também o membro-função

```
void ViaSemaforizada::abrirSemaforo();
```

A função simula o tempo de abertura do semáforo, definido pelo *tempoDeVerde*. Considere que, num ciclo **for**, por exemplo, cada iteração equivale a um segundo. Quando o semáforo abre, o primeiro veículo da fila deixa a via e o veículo seguinte terá que percorrer a distância igual ao comprimento do primeiro veículo para sair da via também, e assim sucessivamente para os veículos seguintes, durante o tempo de verde. Para esse efeito, utilize uma variável *temporizador*, para guardar temporariamente o tempo que o veículo seguinte levará para

percorrer a distância deixada pelo primeiro veículo, baseado na sua velocidade. O temporizador é decrementado de 1 segundo e quando o seu valor for zero, significa que o veículo seguinte chegou ao semáforo e está pronto para deixar a fila. Implemente portanto, a função-membro

```
int ViaSemaforizada::tempoParaProximo(Veiculo v)
```

Que actualizará o temporizador para cada próximo veículo da fila, enquanto o semáforo estiver aberto.

b) Implemente o membro-função

```
double ViaSemaforizada::densidadeInstantanea()
```

A densidade instantânea de uma via é definida como o número de veículos por comprimento de via, dada em veículos por quilómetro (Veic/Km).

c) Implemente o membro-função

```
double ViaSemaforizada::fluxoSaturacao();
```

O fluxo de saturação de uma via é definido como o máximo número de veículos que são capazes de deixar a via durante o tempo de verde, dado em veículos por hora (Veic/h). Modifique convenientemente a definição da classe *ViaSemaforizada* para facilitar a implementação desta função, nomeadamente para saber quantos veículos efectivamente deixam a via após um ciclo do semáforo aberto.

Teste a aplicação no programa principal, com a adição de novos veículos e algumas chamadas à função *abrirSemaforo*. Experimente também diferentes valores para velocidades e comprimentos dos veículos adicionadas à via (dados em Km/h e m, respectivamente). Por fim, execute as funções *densidadeInstantanea* e *fluxoSaturacao*.