

## ATIVIDADE 4



### Processamento de arquivos de GPS

*Automatic Vehicle Systems* utilizam um módulo *GPS* para a coleta de dados e um módulo *GPRS* para transmissão das informações. As vantagens de um *AVS* incluem:

- Aderência aos horários;
- Economia de combustível; e
- Automatizam a coleta dos dados sobre distância e duração (Gillen; Chang; Johnson, 200).

É possível implementar o *AVS* em um *smartphone* para coletar os dados do *GPS* e armazenar em um arquivo local ou banco de dados na internet (Dolazzi; Scalco; 2014).

Implemente um programa capaz de:

- Ler os dados do arquivo de entrada (formato CSV);
- Calcular e exibir a distância total (em km) percorrida por um caminhão rastreado por *GPS*; e
- Criar um arquivo com os dados lidos do arquivo de entrada, acrescidos da velocidade instantânea, em km/h.

Cada linha do arquivo CSV armazena as seguintes informações:

1. Latitude (em graus);
2. Longitude (em graus);
3. Altitude (m); e
4. Tempo (s).

**Obs: os arquivos CSV utilizados utilizam o padrão brasileiro, ou seja, consideram a vírgula (,) como separador decimal e ponto e vírgula como separador de valores (;).**

É obrigatório o desenvolvimento das seguintes subrotinas:

- **troca\_char** - Procedimento responsável pela troca de caracteres em uma *string*. Recebe como parâmetros uma *string*, o caractere a ser buscado e substituído e o novo caractere, que deve entrar em seu lugar.

Exemplo:

```
troca_char("12,5;13,76", ',', '.') -> "12.5;13.76"
```

- **le\_ponto** - Procedimento responsável pela leitura de um ponto *GPS* a partir de uma *string* que representa uma linha. Recebe como parâmetros uma *string* e um ponteiro para uma variável do tipo *Gps*.

Exemplo:

```
le_ponto("24,177;47,024;98;0", &p)  
p -> {24.177, 47.024, 98.0, 0}
```

- **calcula\_distancia** - Função responsável por calcular a distância (m) entre dois pontos. Recebe como parâmetros duas variáveis do tipo *Gps* e retorna a distância entre elas. Para calcular a distância, deve-se usar as seguintes fórmulas:

$$dh = R_{\oplus} \cdot \arccos \left( \begin{aligned} &\sin(\text{Lat}_A) \cdot \sin(\text{Lat}_B) + \\ &\cos(\text{Lat}_A) \cdot \cos(\text{Lat}_B) \cdot \cos(\text{Lon}_A - \text{Lon}_B) \end{aligned} \right)$$
$$dv = \text{Alt}_B - \text{Alt}_A$$
$$\text{Dist}(A,B) = \sqrt{dh^2 + dv^2}$$

- sendo o raio da Terra:  $R_{\oplus} = 6372795,477598\text{m}$ .

- **calcula\_velocidade** - Função responsável por calcular a velocidade (km/h) do deslocamento descrito por dois pontos. Recebe como parâmetros duas variáveis do tipo *Gps* e retorna a velocidade instantânea entre eles. Use:

$$\text{velocidade}[\text{km/h}] = 3,6 \cdot \frac{\text{Dist}(P_{\text{Atual}}, P_{\text{Anterior}})[\text{m}]}{\text{Tempo}_{\text{Atual}} - \text{Tempo}_{\text{Anterior}}[\text{s}]}$$

- **ler\_arquivo** - Função responsável pela leitura de um arquivo CSV que contém os pontos *GPS* que descrevem a trajetória do caminhão. Recebe como parâmetros o nome do arquivo de entrada (caminho no sistema operacional) a ser lido e uma referência a um array de pontos *Gps*. O array deve ser preenchido com os valores lidos em cada linha do arquivo. A função deve retornar a quantidade de pontos lidos ou -1, caso a leitura não tenha sido bem sucedida.
- **escreve\_dados** - Função responsável por calcular a velocidade instantânea dos pontos de um array de pontos *Gps* e gravá-las como uma nova coluna em um novo arquivo CSV. Recebe como parâmetros o nome do arquivo de saída, um array de pontos *Gps* e o tamanho do array a ser gravado. Retorna a quantidade de pontos registrados ou -1, caso a escrita não tenha sido bem sucedida.

**Atenção! Os dados de velocidade devem ser gravados apenas para o os pares de pontos com localização diferente OU cujo instante de tempo são diferentes entre si. Note que o arquivo de saída poderá ter menos dados do que o arquivo de entrada.**

O programa principal deve somente fazer a leitura dos pontos e, a seguir, gravar as velocidades em outro arquivo.

Exemplo de output:

```
4 pontos lidos!  
4 velocidades gravadas!  
  
Não foi possível ler os pontos...  
  
4 pontos lidos!  
Erro ao gravar as velocidades...
```

## Referências

GILLEN, D; CHANG, E; JOHNSON, D. Productivity benefits and cost efficiencies from its applications to public transit: the evaluation of AVL. In: BEKIARIS, E. NAKANISHI, Y. J. (Org.). **Economic Impacts of Intelligent Transportation Systems**: Innovations and Case Studies. Amsterdam: Elsevier, 2004. P.549-570.

DOLAZZI, L. R. M.; SCALCO, R. Sistema para gerenciamento das tarefas de funcionários utilizando aplicativo em smartphone. São Caetano do Sul, 2014. Relatório (Anteprojeto de Pesquisa). Escola de Engenharia Mauá.

