

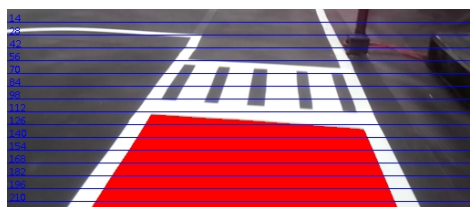
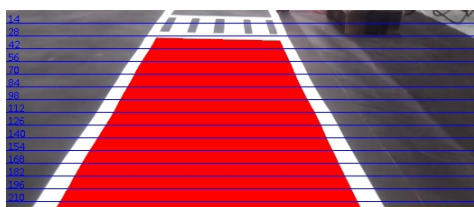
### Instruções

A Parte III do Projeto de LAEDI será realizada na aula prática de 26/03/21 e poderá ser realizada em duplas (a mesma dupla das partes anteriores). A entrega poderá ser realizada até 08/04/21 (quinta-feira, sem prorrogação), pelo sistema run.codes.

A solução deverá ser implementada pelos(as) próprios(as) alunos(as) em sistema Linux na linguagem C ou C++ e não será permitido a utilização de trechos de códigos de outras pessoas ou retirados da internet.

### Deteccão de faixa de pedestres em uma imagem da pista

Na segunda etapa do projeto, foi implementada a identificação do padrão de uma faixa de pedestres dada uma linha da imagem. Considere que agora não há apenas um, mas **L** linhas selecionadas na imagem, como mostrado na figura abaixo, em que cada linha está colorida de azul.



O objetivo da terceira etapa do projeto é realizar a deteção de faixa de pedestres em uma imagem da pista de forma simples, considerado as informações de vários perfis (linhas) da imagem. Dessa forma, cada perfil da imagem deverá ser analisado e o resultado (Pista normal, Faixa de pedestres ou Padrão não identificado) deverá ser armazenado em uma estrutura de dados do tipo lista.

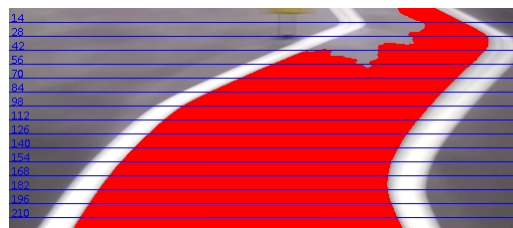
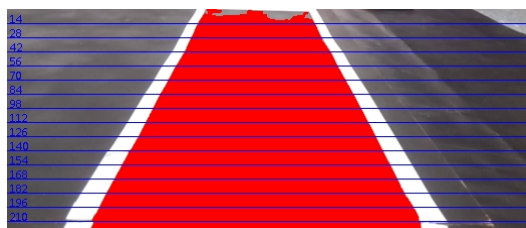
Para armazenar o resultado da análise de cada perfil em uma lista, utilize uma das implementações do livro texto do Ziviani para o tipo abstrato de dados Lista (por meio de arranjo ou apontadores), disponíveis em:

<http://www2.dcc.ufmg.br/livros/algoritmos/implementacoes-03.php>

Cada item da lista, que representa um perfil, deverá conter: um campo **Chave** (número inteiro que será atribuído sequencialmente a cada perfil lido do arquivo) e um campo **Resultado** (Pista normal, Faixa de pedestres ou Padrão não identificado).

É obrigatório utilizar as funções da implementação do Ziviani e não é permitido alterá-las, com exceção da função main e das definições solicitadas neste enunciado. Funções adicionais poderão ser criadas, se necessário.

Cada grupo deverá escolher qual método será utilizado para realizar a detecção de faixa de pedestres (por exemplo, pode ser considerado que há faixa de pedestres se em pelo menos um perfil for detectado o padrão da faixa). Lembrem-se de levar em consideração que há perfis diversos na imagem, alguns somente com o padrão da pista completa, outros com falhas devido à iluminação ou movimento do robô, etc.



Para a implementação da Parte III, o código da Parte II poderá ser utilizado.

### **Formato de entrada dos dados**

Os valores dos  $N$  pixels (elementos) dos  $L$  perfis da imagem serão fornecidos em um arquivo texto com o seguinte formato: na primeira linha o valor de  $L$  e nas linhas seguintes: o valor de  $N$  e na próxima linha os  $N$  valores dos elementos do perfil separados por um espaço em branco. Todos valores são números inteiros.

O programa deverá solicitar o nome do arquivo de entrada.

### **Formato de saída dos dados**

O programa deverá produzir a impressão na tela conforme o modelo a seguir, de acordo com o resultado encontrado (não utilize acentuação nem cedilha na saída para esse programa):

```
Digite o nome do arquivo: teste.txt
Resultado: Pista sem faixa de pedestres.
```

ou

```
Digite o nome do arquivo: teste.txt
Resultado: Pista com faixa de pedestres.
```

### **Observações importantes:**

- Por favor, leia **todas** as informações do enunciado antes de enviar o programa para o run.codes.

- Caso tenha alguma dúvida ou dificuldade, entre em contato com antecedência, evitando deixar para a véspera da entrega.
- O exercício poderá ser realizado em dupla (a mesma das partes anteriores do projeto).
- O programa deverá solicitar ao usuário o nome do arquivo a ser testado. O programa deverá permitir testes com quaisquer arquivos no formato especificado (com nomes distintos, com quaisquer valores de L, de N e dos valores dos elementos).
- O programa deverá ser bem organizado, indentado e conter comentários explicativos relevantes.
- É obrigatório utilizar as funções da implementação do livro texto do Nivio Ziviani (Projeto de Algoritmos).
- Os testes do programa entregue serão realizados pelo sistema run.codes, conforme as instruções de utilização do sistema que podem ser consultadas pelo SIGAA.
- O programa deverá compilar sem erros ou avisos (*warnings*) com o compilador *gcc/g++*. Programas que não compilarem ou que apresentarem erros de execução (falhas de segmentação, etc) não serão corrigidos. Não serão aceitos arquivos enviados por e-mail e nem *'prints'* da execução do programa como prova de seu funcionamento.
- Se houver indícios de plágio no código fonte do programa, a nota final da atividade será zero e serão aplicadas as penalidades previstas no Regime Disciplinar Discente.