

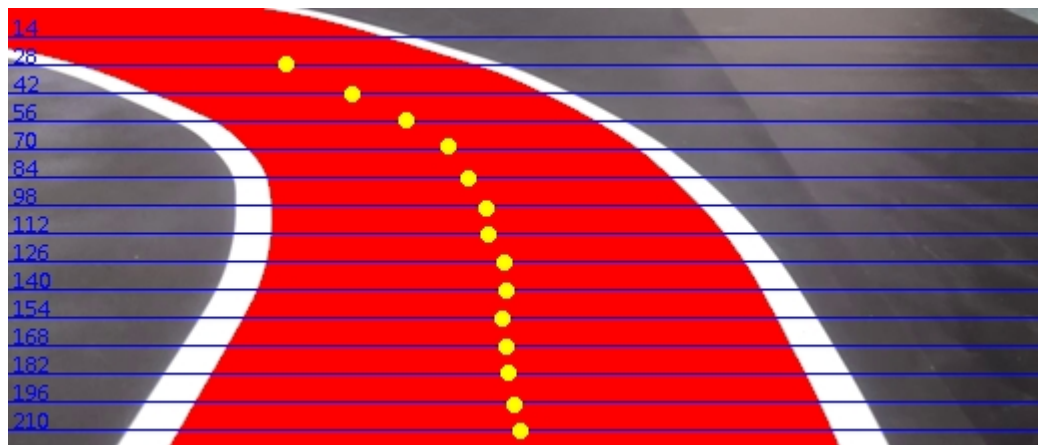
Instruções

A Parte III do Projeto de LAEDI será realizada nas aulas práticas de 06/11/20 e 13/11/20 e poderá ser realizada em duplas (a mesma dupla das partes anteriores). A entrega poderá ser realizada até 19/11/20, pelo sistema run.codes.

A solução deve 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.

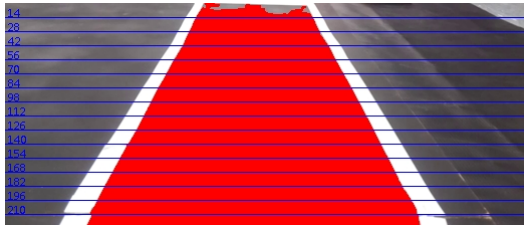
Estimativa do formato da pista

Na segunda etapa do projeto, foi implementado o cálculo do ponto médio da pista dada uma linha da imagem. Considere que agora não há apenas uma, mas **L** linhas selecionadas na imagem, como mostrado na figura abaixo, em que cada linha está colorida de azul. Os pontos médios da pista em cada linha estão marcados em amarelo.

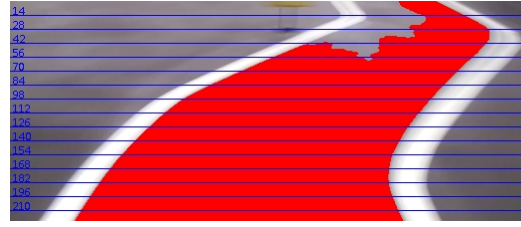


O objetivo da terceira etapa do projeto é estimar o formato da pista de forma simples, dentre três opções: pista em curva à esquerda, pista em curva à direita e pista em linha reta (veja outros exemplos destas situações nas figuras a seguir). Essa estimativa será baseada nas posições sucessivas dos pontos médios de cada linha. Observe que, na figura acima, os pontos médios tendem a se posicionarem cada vez mais à esquerda (de baixo para cima) quando há uma curva à esquerda. De forma semelhante, os pontos médios tendem a se posicionarem cada vez mais à direita quando há uma curva à direita. Quando a pista não está em curva, ou seja, aproximadamente em linha reta, os pontos médios tendem a ficar em posições próximas.

Cada grupo deverá escolher qual método será utilizado para estimar o formato da pista.



Pista aproximadamente em linha reta.



Pista em curva à direita.

Para a implementação da Parte III, o código da Parte II poderá ser utilizado. Considere que, para o formato da pista ser estimado, é necessário ter no mínimo 70% do total de linhas que contenham o padrão da pista (as outras 30% podem ser descartadas devido a imperfeições na imagem ou corte de parte da pista).

Formato de entrada dos dados

Os valores dos N pixels (elementos) das L linhas da imagem serão fornecidos em um arquivo texto com o seguinte formato: na primeira linha o valor de L e nas L linhas seguintes: o valor de N e na linha seguinte os N valores dos elementos da linha 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: Curva a direita.
```

ou

```
Digite o nome do arquivo: teste.txt
Resultado: Curva a esquerda.
```

ou

```
Digite o nome do arquivo: teste.txt
Resultado: Pista em linha reta.
```

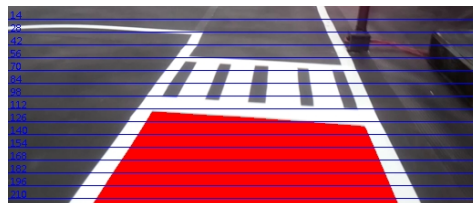
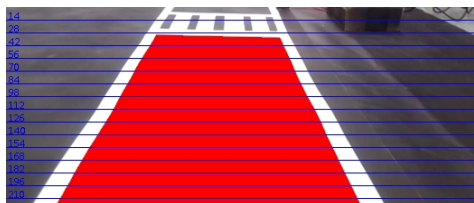
ou

```
Digite o nome do arquivo: teste.txt
Resultado: Formato da pista nao estimado.
```

Implementação extra (opcional): 5 pontos extras

Há vários outros desafios na interpretação das imagens da pista. Por exemplo, a detecção de obstáculos, de sinalização horizontal e de sinalização vertical. Os grupos que conseguirem implementar alguma dessas sugestões poderão obter pontos extras no projeto. Essa implementação extra não é obrigatória no projeto.

As figuras abaixo mostram partes da pista que contêm faixas de pedestres. Uma sugestão de implementação adicional seria buscar o padrão de faixa de pedestres na imagem, que geralmente aparece nas linhas com a sequência de cores: preta, branca, preta, branca, preta, branca, preta, branca, preta, branca, preta, branca, preta, branca, preta.



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, identado 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.