

Projeto 1 - Arquivos (código de barras)

ATP II

17 de setembro de 2021

1 Definição

A leitura de código de barras é um processo algorítmico em que se determina a largura de barras e espaços para determinar dígitos decimais.

Considerando que um arquivo contém um conjunto de códigos de barras, escreva um programa que leia tais códigos e escreva em um segundo arquivo a tradução das barras em valores numéricos.

O texto “How UPC Bar Codes Work”, disponível em <https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/upc.htm> descreve como códigos de barras são reconhecidos (ver principalmente a partir de “Can I decode the bars”).

Aqui usaremos uma versão simplificada desse processo, considerando que os dados armazenados são apenas valores 0 e 1, sendo que um valor 0 está contido em uma barra e o valor 1 está contido em um espaço.

O arquivo contém um número desconhecido de linhas, sendo que cada código de barra ocupa de duas a cinco linhas, existindo sempre pelo menos duas linhas preenchidas com espaços (valores 1) entre dois códigos de barra.

Como indicado no documento citado acima, a primeira barra representa a largura base das barras e espaços. Assim, se essa barra tiver largura de dois valores 0, então podemos ter barras com largura de dois, quatro, seis ou oito 0 consecutivos. O mesmo vale para os espaços, com dois, quatro, seis ou oito valores 1. Esse tamanho base pode ser de um a quatro caracteres (pixels), variando de um código para outro.

2 O que deve ser feito

Escreva um programa que leia o arquivo de entrada, denominado “barras.dat” e escreva os dígitos correspondentes a cada código em um segundo arquivo, denominado “codigos.dat”. No arquivo de saída deve existir uma linha para cada código de barra, sendo que os dígitos devem estar separados por um espaço em branco.

3 Condições de teste

Seu programa deve executar considerando que o arquivo de entrada contém um conjunto de linhas compostas por zeros e uns, sem separação entre eles dentro de uma linha. O exemplo a seguir mostra o começo de cada linha do arquivo de entrada.

```
1111111111111111111111111111111111...
11101010010001110100110011010100001001...
11101010010001110100110011010100001001...
11101010010001110100110011010100001001...
1111111111111111111111111111111111...
1111111111111111111111111111111111...
1111111111111111111111111111111111...
11100110011000000110000111111000011001...
11100110011000000110000111111000011001...
1111111111111111111111111111111111...
```

Sendo que o primeiro código gera 8916..., enquanto o segundo código gera 70..., ou seja, no arquivo de saída teríamos:

```
8 9 1 6 ... ...
7 0 ... ...
```

4 Entrega

Entregar o código fonte do programa, devidamente documentado, no classroom.

PRAZO: 03/10, até 23h59 no classroom.