

# 1 Comportamento reativo (++++)



(++++)

Um robô móvel no formato de um carro com quatro rodas tem a tarefa de desviar de obstáculos. Para perceber o ambiente ele é equipado com 3 sensores de distância capazes de medir a distância do robô para um obstáculo à sua frente. Os sensores, denominados  $s_1$ ,  $s_2$  e  $s_3$ , estão posicionados na frente do robô, rotacionados entre si, de modo a perceber o ambiente de navegação. Tomando  $s_2$  como referência, este sensor aponta para a frente do robô,  $s_1$  está rotacionado  $30^\circ$  à esquerda e  $s_3$  está rotacionado a  $30^\circ$  à direita. Os três sensores estão calibrados para medir a distância do obstáculo ao qual está apontando até um ponto de referência que é comum a todos os sensores. Nas Figuras 1 esse ponto é indicado em vermelho.

O computador controlador do robô lê os dados dos três sensores simultaneamente e decide se deve PARAR, virar à ESQUERDA ou à DIREITA. O sensor  $s_2$  mede a distância direta do ponto logo à frente do robô devido ao seu alinhamento com o chassi do próprio robô. Por outro lado, como os sensores  $s_1$  e  $s_3$  estão rotacionados, deve ser considerado o ângulo de sua pose para calcular a distância perpendicular do ponto ao robô.

Considere as variáveis  $d_1$ ,  $d_2$  e  $d_3$  sendo as distâncias perpendiculares dos pontos dados pelos sensores  $s_1$ ,  $s_2$  e  $s_3$  respectivamente. O comando de VOLTAR é executado caso os três sensores identifiquem uma parede perpendicular ao robô, ou seja,  $d_1 = d_2 = d_3$ . O comando DIREITA é executado caso o robô identifique uma parede inclinada de modo que  $d_1 < d_2 < d_3$  e o comando ESQUERDA é executado caso os sensores indiquem  $d_1 > d_2 > d_3$ . O comando PARAR é executado caso contrário.

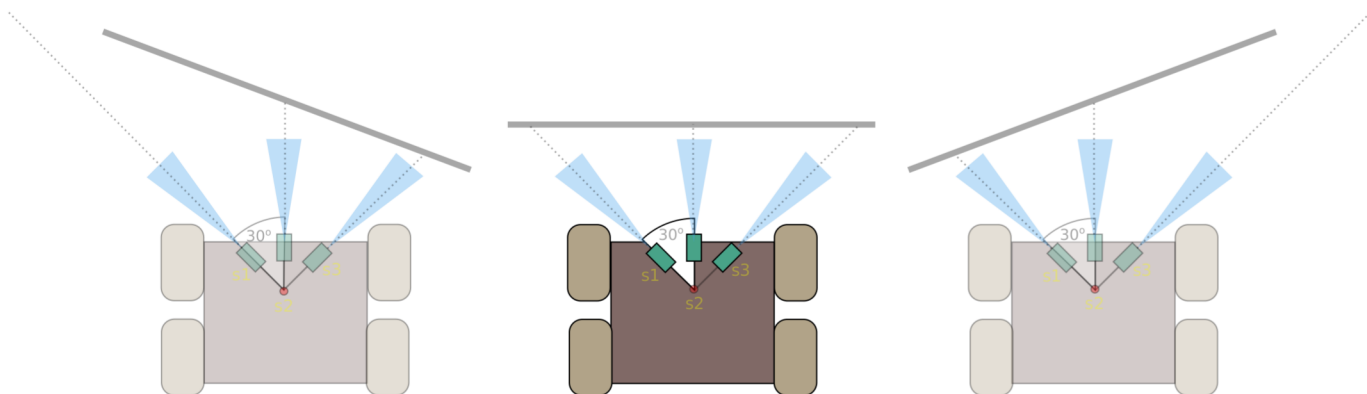


Figura 1: Robô.

## Entrada

O programa deve ler 3 números **double** que indicam as distâncias  $s_1$ ,  $s_2$  e  $s_3$  respectivamente.

## Saída

O programa deve escrever uma linha com comando a ser realizado pelo robô. Os comandos são: VOLTAR, ESQUERDA, DIREITA e PARAR.

## Exemplo

Entrada	Saída
100 100 100	PARAR

<b>Entrada</b>		<b>Saída</b>
1	50 600	DIREITA

## 2 ValidarSenha(++)



(++)

Escreva um programa em C que faça a validação de senha. A validação da senha é realizada pela categoria do usuário. Existem três categorias validas:

- (A)dministrador senha: 321456
- (G)erente senha: 9510753
- (U)suário senha: 78955632

Se a senha conter menos de 6 dígitos, emitir a mensagem "SENHA INVALIDA: *senha* - MINIMO 6 DIGITOS". Para senha com 6 ou mais dígitos e que não estiver correta, conforme categoria, deverá apresentar a mensagem "ACESSO NEGADO". Se a categoria e a senha estiverem corretas emitir a mensagem "ACESSO PERMITIDO A CATEGORIA *x*". Qualquer categoria diferente de A, G ou U resulta na mensagem "CATEGORIA INVALIDA: *x*".

### Entrada

O programa deve ler um valor inteiro correspondente a senha do usuário e um valor do tipo char, corresponde a categoria do usuário. As categorias podem ser A, G ou U. Qualquer valor diferente destes devem emitir um aviso "CATEGORIA INVALIDA".

### Saída

A saída deve apresentar uma das seguintes mensagens (seguir a ordem abaixo para validação):

- SENHA INVALIDA: *senha* - MINIMO 6 DIGITOS, onde *senha* é a senha informada.
- ACESSO PERMITIDO A CATEGORIA *x*, onde *x* é o código da categoria
- ACESSO NEGADO
- CATEGORIA INVALIDA: *x*, onde *x* é o código da categoria

Após a mensagem, o programa deve imprimir o caractere de quebra de linha.

### Exemplo

<b>Entrada</b>
321456 A
<b>Saída</b>
ACESSO PERMITIDO A CATEGORIA A

<b>Entrada</b>
9510753 W
<b>Saída</b>
CATEGORIA INVALIDA: W

<b>Entrada</b>
123 U
<b>Saída</b>
SENHA INVALIDA: 123 - MINIMO 6 DIGITOS

<b>Entrada</b>
123123 G
<b>Saída</b>
ACESSO NEGADO

### 3 ValidarData(++)



(++) Faça um programa para verificar se uma data é válida. Uma data é válida se o dia, mês

e ano forem informados corretamente. Por exemplo, na data 18/13/2010, o mês 13 não existe, portanto não é uma data válida. Lembre-se que há meses com 31 dias, 30 dias, 29 dias ou 28 dias.

Sabe-se que para descobrir se um ano é bissexto (29 dias) deve-se verificar se ele é divisível por 4. Entretanto, os anos divisíveis por 100 (apesar de divisíveis por 4) são exceção, eles não são bissextos. Os anos divisíveis por 400 (apesar de divisíveis por 100) são a exceção da exceção, eles também são bissextos.

Alguns exemplos para ilustrar o ano bissexto:

- 1996 é bissexto (divisível por 4)
- 1900 não é bissexto (divisível por 4, seria bissexto, mas é também divisível por 100)
- 2000 é bissexto (divisível por 100 portanto não seria bissexto, mas é também divisível por 400, portanto bissexto).

#### Entrada

O programa deve ler 3 valores inteiros corresponde ao dia, mês e ano. O ano deve estar entre 1900 e 2036.

#### Saída

O programa deve apresentar uma linha contendo a data no formato dia/mês/ano ou a mensagem "DATA INVALIDA". O programa deve imprimir a mensagem "DATA FORA DO INTERVALO ESTIPULADO", caso o ano informado seja inferior a 1900 ou superior a 2036.

No final da mensagem, imprima o caractere de quebra de linha.

#### Observações

Os meses de janeiro, março, maio, julho, agosto, outubro e dezembro tem 31 dias. Os meses de abril, junho, setembro e novembro tem 30 dias. O mês de fevereiro tem 28 dias ou 29 dias.

#### Exemplo

<b>Entrada</b>
10 11 1790
<b>Saída</b>
DATA FORA DO INTERVALO ESTIPULADO

<b>Entrada</b>
25 12 1987
<b>Saída</b>
25/12/1987

<b>Entrada</b>
30 2 2011
<b>Saída</b>
DATA INVALIDA