

Trabalho de Programação 1 - Processador RAMSES

1. Descrição Geral

O objetivo deste trabalho é escrever um programa para o processador Ramses (usando necessariamente o montador Daedalus para desenvolver o código fonte) que calcule a movimentação de um robô em um tabuleiro com 256 x 256 posições e informe sua posição final após executada uma série de movimentos a partir de uma posição inicial. As coordenadas X e Y das posições do tabuleiro variam de -128 a +127 e são representadas como valores de 8 bits em complemento de 2. As especificações dos movimentos e as coordenadas da posição inicial do robô serão os dados de entrada. As coordenadas da posição final do robô no tabuleiro serão os dados de saída.

2. Dados de entrada e saída

O programa receberá como dados de entrada as coordenadas (X_0, Y_0) da posição inicial do robô no tabuleiro e uma lista com as informações sobre os movimentos que o robô deve fazer a partir da posição inicial, na qual para cada movimento haverá duas informações: direção do movimento e distância a percorrer (em número de posições no tabuleiro).

A direção do movimento será indicada por um valor de 8 bits em complemento de 2, com o seguinte significado:

0	Movimento na direção POSITIVA do EIXO X
1	Movimento na direção POSITIVA do EIXO Y
128 (-128)	Movimento na direção NEGATIVA do EIXO X
129 (-127)	Movimento na direção NEGATIVA do EIXO Y

A distância a percorrer será um valor inteiro de 1 a 127, que deverá ser somado ou subtraído do valor da coordenada correspondente à direção do movimento para obter as coordenadas da nova posição do tabuleiro para onde o robô irá se mover.

Quando a coordenada alterada por um movimento ficar fora dos limites do tabuleiro (valores menores do que -128 ou maiores do que +127), o robô irá para a outra extremidade do tabuleiro, na mesma linha ou coluna, e continuará avançando a partir de lá. Isto significa que o programa não precisará verificar estas situações, pois ao somar um valor a uma coordenada positiva, se ele passar de 127 o resultado já será a coordenada correta, no lado negativo do eixo correspondente. O mesmo acontecerá se for feita uma subtração a partir de uma coordenada negativa: o resultado positivo já será a coordenada correta.

A quantidade de movimentos a executar é variável (de 0 a 14 movimentos) e o programa saberá que executou todos os movimentos quando encontrar uma especificação de movimento com o valor negativo no byte de distância a percorrer. Ao final da execução, a posição final do robô no tabuleiro será indicada pelas coordenadas (X_f, Y_f) e as coordenadas da posição inicial devem estar inalteradas. Portanto, inicialmente seu programa deverá mover os valores das coordenadas iniciais para as variáveis das coordenadas finais e, a partir daí, atualizar as coordenadas finais a cada movimento processado.

Para os dados de entrada e saída devem ser utilizadas, obrigatoriamente, as seguintes posições de memória:

Endereço	224	225	226	227	...	252	253	254	255
Dado	X_0	Y_0	DS_1	D_1		DS_{14}	D_{14}	X_f	Y_f

Onde DS_i indica a direção e sentido do movimento, conforme especificado anteriormente, e D_i a distância a ser percorrida (em número de posições do tabuleiro). O código do programa e as demais variáveis e constantes necessárias deverão ocupar os endereços 0 a 223 da memória.

3. Exemplo

Supondo que os dados de entrada para o programa sejam os seguintes (valores em decimal):

Endereço	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238
Dado	0	0	0	3	1	2	128	6	129	5	0	255			

Neste caso, após mover as coordenadas iniciais para as finais ($X_f = 0$, $Y_f = 0$), o robô deverá fazer 4 movimentos: 3 posições no sentido positivo do eixo dos X ($X_f = 3$, $Y_f = 0$), 2 posições no sentido positivo do eixo dos Y ($X_f = 3$, $Y_f = 2$), 6

posições no sentido negativo do eixo dos X ($X_f = -3$, $Y_f = 2$) e mais 5 posições no sentido negativo do eixo dos Y ($X_f = -3$, $Y_f = -3$), conforme ilustrado na Figura 1, onde os pares de coordenadas (X_i, Y_i) estão indicados entre parênteses e a direção e sentido do movimento e a distância a percorrer entre colchetes [DS_i, D_i].

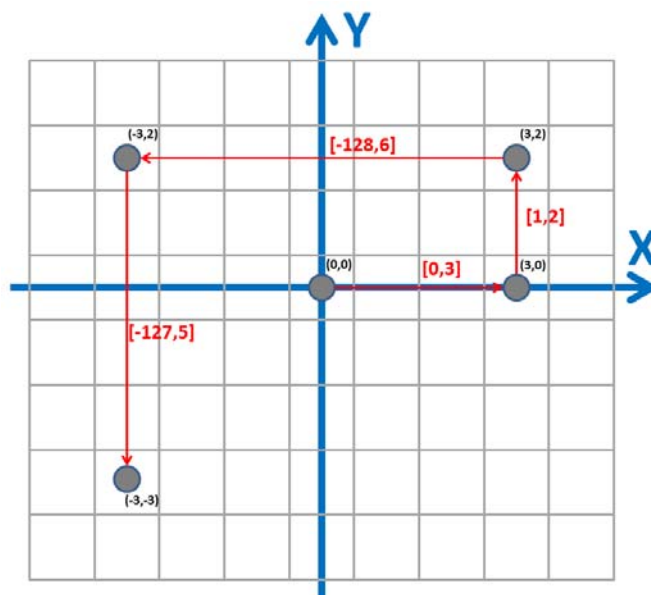


Figura 1 – Exemplo: sequência de 4 movimentos

Portanto, para estes dados de entrada, seu programa deverá fornecer nas palavras 254 e 255 o seguinte resultado:

Endereço	254 (X_f)	255 (Y_f)
Dado	253 (-3)	253 (-3)

Observe que o conteúdo dos endereços 224 e 225 não deve ser modificado.

4. Correção dos Trabalhos

Os arquivos fonte do RAMSES entregues serão montados usando o montador DAEDALUS. A seguir serão aplicados 20 casos de teste, de forma sequencial e contínua (sem recarga ou inicialização da memória). A nota final do trabalho será proporcional ao número de casos de teste em que o programa produzir a resposta correta (cada caso de teste vale 5 pontos de um total de 100).

5. Bônus

Os programas que fornecerem resultados corretos para todos os 20 casos de teste poderão concorrer a um Bônus de Desempenho, no valor de 10 pontos (10% da nota). Esse bônus será concedido para as soluções que utilizarem o menor número de acessos à memória (aqueles mais rápidos). O desempenho será medido somando os números de **acessos** usados para solucionar todos os 20 casos de teste.

6. Entregáveis: o que deve ser entregue?

Somente os arquivos fonte (arquivos .RAD) escritos na linguagem simbólica do RAMSES com as soluções do problema apresentado deverão ser entregues via Moodle da disciplina. **Esses arquivos serão montados com o DAEDALUS.** O programa fonte deverá conter comentários descritivos da implementação. Por exemplo, podem ser usados comandos em alto nível da linguagem “C”. Para nomear os arquivos, utilize todo o seu nome, usando maiúsculas e minúsculas, sem espaços em branco e sem acentos.

O trabalho deverá ser entregue até a data especificada no link de entrega no sistema Moodle. **Não serão aceitos trabalhos após o prazo estabelecido.**

7. Observações

Recomenda-se a troca de ideias entre os alunos. Entretanto, a identificação de cópias de trabalhos acarretará na aplicação do Código Disciplinar Discente e a tomada das medidas cabíveis para essa situação (**tanto o trabalho original quanto os copiados receberão nota zero**).

O professor da disciplina reserva-se o direito, caso necessário, de solicitar uma demonstração do programa, onde o aluno será arguido sobre o trabalho como um todo. Nesse caso, a nota final do trabalho levará em consideração o resultado da demonstração.

8. Casos de Teste

Para testar plenamente sua solução, use os casos de teste abaixo, da seguinte forma:

1. Carregue seu código (arquivo .MEM gerado pelo Daedalus a partir do .RAD) no simulador Ramses
2. Coloque os valores indicados nas palavras 224 a 252 (as palavras em branco não precisam ser preenchidas)
3. Coloque o valor 0 no PC
4. Execute o programa
5. Verifique se os resultados nas palavras 224, 225, 254 e 255 são os mesmos indicados no final da linha de cada caso de teste
6. Para testar os demais casos, volte à etapa 2 SEM RECARREGAR O CÓDIGO DE SUA SOLUÇÃO NO SIMULADOR

Caso	End. →	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	...	254	255
1	Dados	0	0	0	3	1	2	128	6	129	5	0	255								...	253	253
2		127	127	0	255																	127	127
3		128	128	0	127	0	127	0	2	1	127	1	127	0	255							128	126
4		0	0	0	10	1	5	0	10	1	5	0	10	1	5	0	10	0	255			40	15
5		10	20	0	5	129	10	128	15	0	255											0	10
6		0	0	0	50	1	50	128	50	129	50	0	255									0	0
7		0	0	0	100	0	50	1	50	1	100	0	255									150	150
8		0	0	128	50	129	100	128	100	129	50	0	255									106	106
9		128	128	0	127	0	127	1	127	1	127	0	10	1	10	0	255					136	136
10		127	127	128	127	128	127	129	127	129	127	128	20	129	30	0	255					109	99