

## Trabalho de Programação – Processador RAMSES

### 1. Descrição Geral

Nesse trabalho você deverá desenvolver um programa para o processador RAMSES que seja capaz identificar a presença de um determinado *string* dentro de outro *string*. Os *strings* são compostos por bytes, da mesma forma que os *strings* usados para representar caracteres na linguagem “C”. Portanto, os elementos do vetor podem ter qualquer valor, sendo reservado o valor 0x00 para delimitar o final do *string*.

Ao iniciar a execução, seu programa deve acessar a memória de dados onde estão as informações dos dois *strings* de entrada: o *string* onde procurar e o *string* a ser procurado. Para referência, o *string* onde procurar será chamado de SFILE e o *string* a ser procurado será chamado de SLOOK.

Ao final da execução, seu programa deverá fornecer como resultado o local onde o *string* SLOOK foi encontrado dentro do *string* SFILE. Seu programa também deverá informar quando não for possível encontra o *string* SLOOK dentro do *string* SFILE. Para referência, o resultado da execução será colocado em SRES.

Segue na “Tabela de Exemplos” abaixo alguns casos, com os resultados esperados. Os resultados estão informados através da posição relativa no *string* SFILE onde o SLOOK foi encontrado, a partir da posição “0” (zero) de SFILE.

Exemplo	SFILE	SLOOK	SRES
1	“12abc”	“abc”	Posição 2 de SFILE
2	“12aabcd”	“abc”	Posição 3 de SFILE
3	“abc”	“abc”	Posição 0 de SFILE
4	“1234”	“abc”	<i>String</i> não encontrado
5	“12abc34”	“abc”	Posição 2 de SFILE
6	“12abcabc34”	“abc”	Posição 2 de SFILE
7	“12abababc34”	“abc”	Posição 6 de SFILE
8	“12ab”	“abc”	<i>String</i> não encontrado
9	“bc”	“abc”	<i>String</i> não encontrado
10	“”	Qualquer <i>string</i>	<i>String</i> não encontrado
11	“12abc”	“”	Posição 5 de SFILE

Tabela de Exemplos

Cuidado especial deve ser tomado para os casos em que um dos dois *strings* estiver vazio.

Se SFILE estiver vazio, o resultado deve ser “*String* não encontrado”. Este é o caso do exemplo 10 acima.

Se SFILE não estiver vazio mas SLOOK estiver vazio, o resultado deve ser a posição em que se encontra o “\0” no final de SFILE.

**Observação:** será usada a notação da linguagem “C” para representar os números em hexadecimal. Também será usada a notação “\0” para representar o valor 0x00.

### 2. Disposição dos dados na memória

Seu programa, para ser executado, necessita a localização dos *strings* SFILE e SLOOK. O endereço de início do *string* SFILE será encontrado no endereço 0xE0 da memória; o endereço de início do *string* SLOOK será encontrado no endereço 0xE1 da memória. Os dois *strings* estarão dispostos na memória, na região entre os endereços 0xE3 até 0xFF, e pode ocorrer dos *strings* estarem sobrepostos.

Depois de executado seu programa, quando for atingida a instrução de HLT, o resultado com o endereço de memória onde foi encontrado o *string* procurado deverá estar escrito no endereço 0xE2. Caso o *string* procurado não seja encontrado, o endereço 0xE2 deverá conter o valor 0x00.

Para correção de seu programa será verificado apenas o conteúdo do endereço 0xE2. Por exemplo, a partir do caso 1 da “Tabela de Exemplos”, serão realizadas as seguintes operações:

1) Antes de executar seu programa a memória de dados será preenchida da seguinte forma:

Endereço	0xE0	0xE1	0xE2	0xE3	0xE4	0xE5	0xE6	0xE7	0xE8	0xE9	0xEA	0xEB	0xEC
Conteúdo	0xE3	0xE9	???	0x31	0x32	0x61	0x62	0x63	0x00	0x61	0x62	0x63	0x00

2) Então, seu programa será executado. Durante essa execução, seu programa deverá utilizar as informações fornecidas nos *strings* de entrada.

3) Finalmente, após o término da execução do programa, o resultado escrito no endereço 0xE2 será verificado. No caso do exemplo, o resultado esperado é 0xE5, conforme mostrado abaixo:

Endereço	0xE0	0xE1	0xE2	0xE3	0xE4	0xE5	0xE6	0xE7	0xE8	0xE9	0xEA	0xEB	0xEC
Conteúdo	0xE3	0xE9	0xE5	0x31	0x32	0x61	0x62	0x63	0x00	0x61	0x62	0x63	0x00

### 3. Restrições e garantias

- Seu programa e todas as suas variáveis de trabalho devem estar restritos a região de memória entre 0x00 e 0xDF;
- Caso seja necessário inicializar suas variáveis internas, isso deve ser feito de forma explícita através de instruções do programa. Lembre-se que as diretivas do tipo “DB” são um mecanismo para a alocação de espaço de memória, não sendo adequado como mecanismo de inicialização;
- É garantido que os vetores de entrada estejam completamente contidos na região reservada para eles, incluindo a marca de final de *string*;
- É garantido que os vetores de entrada possuam, sempre, a marca de final de *string*.

### 4. Correção dos Trabalhos

Os arquivos fonte do RAMSES entregues serão montados usando o montador DAEDALUS.

Para a correção, serão aplicados 20 (vinte) casos de teste. Há apenas duas opções de correção dos casos de teste: correto ou errado.

Para cada caso de teste em que o programa fornecer a resposta correta, serão atribuídos 5 pontos. Portanto, os programas que fornecerem os resultados corretos para todos os casos de teste receberão 100 pontos.

Para a correção serão realizadas as seguintes etapas:

- 1) Carga do programa na memória;
- 2) Para cada um dos 20 (vinte) casos de teste será realizado o seguinte:
  - a. Alteração da memória de dados com as informações de entrada;
  - b. Reset do processador, colocando o PC em 0x00 (zero);
  - c. Execução do programa;
  - d. Verificação do resultado.

### 5. O que deve ser entregue?

Deverá ser entregue somente o arquivo fonte (arquivo .RAD) escrito na linguagem simbólica do RAMSES, com a solução do problema apresentado, no Moodle da disciplina.

O programa fonte deverá conter comentários descritivos da implementação. Por exemplo, nos comentários podem ser usados comandos da linguagem “C”.

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

## 6. 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. Inicialmente, nesses casos, **ambos os trabalhos: original e cópias, 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.