

Arquitectura de Computadores

IGE/ETI 2011/2012 1ºSemestre

3º Trabalho de avaliação - Programação em *assembly*

Introdução

Pretende-se implementar um programa que, utilizando o *assembly* do processador MAC-1, permita realizar um conjunto de funcionalidades sobre uma imagem, representada por uma matriz de caracteres com dimensão máxima 10x10 (ver figura 1). Cada pixel da imagem é representado por um carácter na matriz correspondente. Um pixel branco corresponde ao espaço e um preto corresponde ao carácter "x".

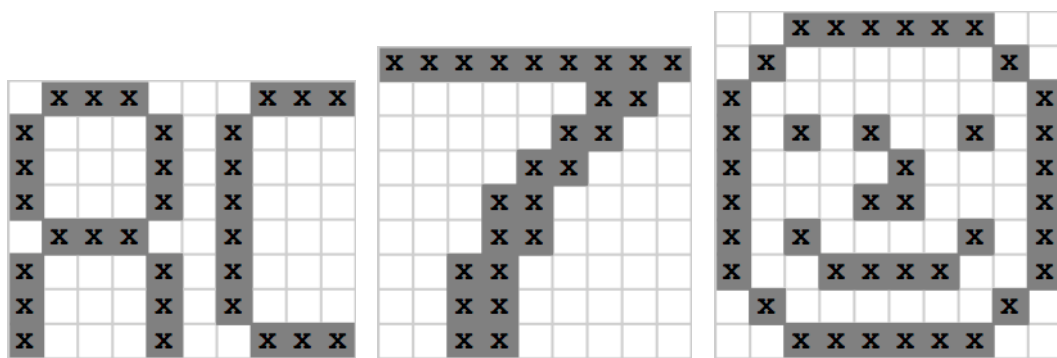


Figura 1 – Exemplos matrizes de caracteres, com dimensões 8x10, 9x9 e 10x10.

O trabalho deverá ser realizado em várias etapas, por forma a facilitar o seu desenvolvimento. No final, deverão estar implementadas certas funcionalidades para manipular uma imagem a preto e branco, tais como:

1. Imprimir a imagem
2. Contar e mostrar no ecrã o número de pixéis brancos e pretos
3. Inverter a imagem (trocar espaços por "x")
4. *Mirror* vertical

O seguinte pedaço de código *assembly* contém a representação da primeira imagem da figura 1, juntamente com informação sobre o número de linhas e colunas da matriz.

```
trabid:  .string "Trabalho 3 de AC"
        10
        0
matriz:  .string " xxx  xxx"
        .string "x   x x  "
        .string "x   x x  "
        .string "x   x x  "
        .string " xxx x  "
        .string "x   x x  "
        .string "x   x x  "
        .string "x   x  xxx"
n_lin:   8
n_col:   10
```

Etapa 1 - Funções básicas

- a) Crie uma função para calcular o endereço de memória correspondente à posição (i, j) da matriz. A função recebe os argumentos i e j e retorna a posição de memória correspondente, que se obtém pela expressão:

$$pos = matriz + i \times n_col + j,$$

em que n_col é o número de colunas da matriz e $matriz$ é a posição de memória onde se encontra o primeiro valor da matriz (canto superior esquerdo).

- b) Faça uma função que receba como argumento um número inteiro positivo e imprima esse número no ecrã.

Etapa 2 – Endereçamento indirecto

- a) Crie uma função que recebe como argumento um endereço de memória e escreve o conteúdo desse endereço e de todos os seguintes, até encontrar uma posição de memória com o valor 0. Use-a para escrever a mensagem que se encontra na posição de memória designada por `trabid` (ver introdução).
- b) Crie uma função que recebe os argumentos i e j e devolve o valor que se encontra nessa posição da matriz. A posição de memória correspondente pode ser calculada utilizando a função a) que desenvolveu na etapa 1.
- c) Crie uma função que recebe os argumentos i, j e um carácter c , e escreve esse carácter nessa posição da matriz.
- d) Crie uma função para mostrar a matriz no ecrã.

Etapa 3 – Recursividade

- a) Crie uma função recursiva que recebe como argumento um endereço de memória e escreve o conteúdo desse endereço e de todos os seguintes, até encontrar uma posição de memória com o valor 0, por ordem inversa. Use-a para escrever a mensagem, que se encontra na posição de memória designada por `trabid`.

Etapa 4 – Programa final

Fazendo uso das funções anteriormente criadas, faça um programa que produza os seguintes resultados, pela ordem indicada:

1. Imprima no ecrã a mensagem que está no endereço `trabid`.
2. Imprima a imagem (matriz) original

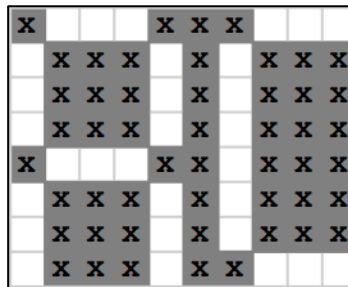
3. Indique quantos pixéis brancos e pretos se encontram na imagem.

Exemplo:

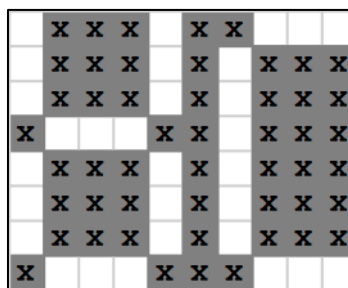
Pretos: 30

Branco: 50

4. Inverta a imagem, trocando os pixéis brancos por pretos e restantes pixéis por branco.



5. Aplique um *mirror* vertical e imprima a nova imagem



6. Imprima no ecrã a mensagem, que está no endereço `trabid`, ao contrário recorrendo a uma função recursiva.

Exemplo: "Trabalho de AC" → "CA ed 3 ohlabarT"

Entrega do trabalho

A entrega do trabalho deverá ser feita até ao dia 22 de Dezembro (quinta-feira).

Para entregar o trabalho deve efectuar os seguintes passos:

- 1) Deve registar o seu grupo em
http://ac.dcti.iscte.pt/www/myscripts/Grupos/inscrever_grupos.php
- 2) Submeter o trabalho em (precisa da *password* usada no registo)
http://ac.dcti.iscte.pt/www/myscripts/Trabs/submeter_trab03.php

É apenas necessário entregar um ficheiro ".zip", contendo o ficheiro com o código fonte (.asm) e o ficheiro compilado correspondente (.hex).

Trabalhos entregues em atraso não serão avaliados.

O programa deverá realizar as tarefas descritas na etapa 4, embora deva conter todas as funções mencionadas nas três primeiras etapas. No ficheiro .asm deverão existir os comentários necessários à compreensão do código, assim como o desenho da pilha sempre que isso se justificar.

A avaliação do trabalho é feita com base nos seguintes critérios:

- Funcionamento, estrutura do código e das rotinas desenvolvidas;
- Apresentação do código e relevância dos comentários;
- Apresentação dos resultados (output);
- Originalidade (trabalhos muito semelhantes são penalizados ou anulados).