 <b>FEUP</b>	<p>Faculdade de Engenharia de Universidade do Porto</p> <p>Mestrado Integrado em Engenharia Informática e Computação</p> <p><b>Microprocessadores e Computadores Pessoais</b></p>	<p>2º miniteste</p> <p>17/12/2008</p> <p>Duração: 1 hora</p> <p>Sem consulta</p>
--	---	--

**Atenção: 1) Justifique todas as respostas**  
**2) O miniteste é constituído por 5 questões, sendo as 2 últimas, 5A e 5B, em alternativa.**

## 1

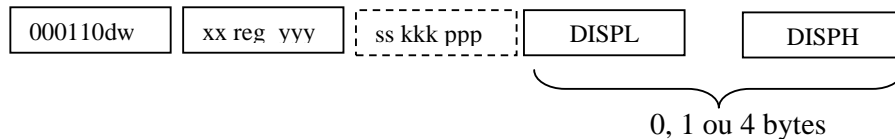
Descreva a evolução da pilha durante a execução do fragmento de programa abaixo. Apresente os diagramas ilustrativos do estado da pilha depois das instruções assinaladas com (\*\*).

```
.code
yy PROC USES edi apt: DWORD PTR, s:BYTE, tam:DWORD
Local temp:dword
    mov edi, apt
    ...
    mov ecx, tam ; **
    ...
    ret
yy endp

start:
    invoke troca, addr apt1, addr apt2
    ...
    mov ecx, 2
@@:  push ecx ; **
    invoke yy , addr ap1, k, ecx
    ...
    pop ecx
    loop @B
    ...
    invoke ExitProcess, Null
end start
```

## 2

O formato da instrução SUBB no microprocessador da família IA-32 da Intel é :



em que:

w – indica se o operando da instrução é do tipo *byte*(0) ou *doubleword*(1)  
d – destino reg (0) ou destino *xyyy* (1)  
xx – modo de endereçamento  
yyy – registo usado no modo de endereçamento  
ss – escala (00<sub>B</sub>:1, 01<sub>B</sub>:2, 10<sub>B</sub>:4, 11<sub>B</sub>:8)  
kkk – registo de índice  
ppp – registo de base  
O campo assinalado a traço interrompido é opcional.

Codifique a instrução: **SUBB Byte Ptr [EDI + 1FFF<sub>H</sub>], DL**

### 3

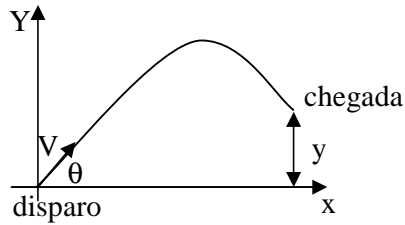
Descreva e explique o funcionamento da rotina **xxx**.

```
xxx PROC Uses edi apt_or: DWORD PTR, apt_d: DWORD PTR, dim: DWORD
    mov esi, apt_or
    mov edi, apt_d
    mov ecx, dim
    cld
    rep movsd
    ret
xxx ENDP
```

### 4

Com base nas instruções do coprocessador aritmético escreva a rotina Tempo que determina o tempo  $t$ , gasto por uma bala a percorrer o percurso desde o ponto do disparo até atingir o ponto de cota  $y$  (ver figura). Considere que a velocidade inicial da bala é  $V$  e que esta foi disparada com um ângulo  $\theta$  (em radianos). Despreza-se o atrito no deslocamento. O valor de  $t$  pode ser deixado no topo da stack do coprocessador.

$$t = \frac{1}{g} \left[ V \sin \theta + \sqrt{\frac{V^2 \sin^2 \theta}{g} - 2Y} \right]$$



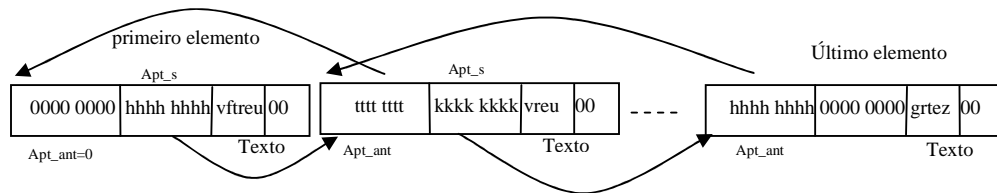
## 5A

Considere um array em que cada elemento contém 3 campos:

- o 1º campo é o apontador **apt\_ant** que aponta para o elemento anterior;
- o 2º é o apontador **apt\_seg** que aponta para o elemento seguinte;
- o 3º é um campo de texto que termina com o carácter de código 00.

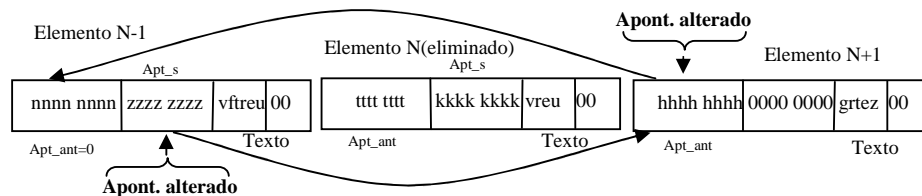
O apontador **apt\_ant** do 1º elemento está preenchido com 00..00 e o apontador **apt\_seg** do último elemento está preenchido com 00..00.

- a) Escreva a rotina **Procura** que identifica o elemento do array cujo texto tenha o comprimento máximo. No fim a rotina retorna em EAX o apontador desse elemento do array.



**Procura Proto apt: DWORD;** apt aponta para *Apt\_ant* do 1º elemento do array

- b) Escreva a rotina **Del** que elimina o elemento de índice N no array. Para se eliminar um elemento do array seguem-se as indicações da figura relativas aos apontadores.



## 5B

Considere um array X de 8\*N elementos em que cada elemento é um byte representado em complemento para 2.

- a) Com base em instruções MMX escreva a rotina **NEG** que calcula  $X = -X$ , isto é, cada elemento do array é o negativo do anterior.

Nota: calcular o negativo de um valor em complemento para 2 é complementar o valor e adicionar uma unidade.

**NEG Proto apt:byte ptr, dim: dword**

Apt aponta para o 1º elemento do array e dim é o número de bytes do array.

- b) Com base em instruções MMX escreva a rotina **MOD** que calcula o módulo de X, isto é,  $X = -X$  apenas se  $X < 0$ .

**MOD Proto apt:byte ptr, dim: dword**