

Arquitectura de Computadores
Licenciatura em Engenharia Informática
1º teste (D) – 2008/04/23 – Duração: 1h30m + 30m

Nome: _____	Número: _____
Classificação: _____	

Teste sem consulta.

A interpretação do enunciado faz parte da avaliação. Explícite nas suas respostas todas as hipóteses assumidas. Por favor, tente focalizar a sua respostas para que estas se enquadrem na zona delimitada.

1 Teórica - Escolha Múltipla

Deve assinalar com um **X** a resposta correcta. Cada resposta errada desconta 25% da cotação da pergunta.

Para todas estas questões considere que está no contexto de um computador Intel com o ISA IA-32 e que o assembly utilizado é o NASM.

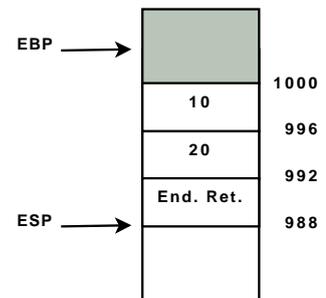
Q-1 [1.0 val.] Considere a definição da seguinte variável: a: **resb 32**

Qual das seguintes seqüências de instruções coloca o conteúdo do registo `cl` na última posição do array `a`.

1. **mov** ebx, a **mov** esi, 31 **mov** [ebx+esi*2], cl
2. **mov** ebx, [a] **mov** esi, 31 **mov** [ebx+esi*2], cl
3. **mov** ebx, a **mov** esi, 31 **mov** [ebx+esi*4], cl
4. **mov** ebx, [a] **mov** esi, 31 **mov** [ebx+esi*4], cl
5. Nenhuma das anteriores

Q-2 [1.0 val.] Considere que está a executar uma subrotina com o ambiente de execução ilustrado na figura à direita. Indique que seqüência de instruções é a correcta para terminar a execução da subrotina, de forma a voltar para o ponto de onde foi chamada e retornar o valor 10, de acordo com a convenção adoptada para o retorno de resultados.

1. **mov** eax, 10 **pop** ebp **ret**
2. **mov** eax, 10 **pop** esp **ret**
3. **pop** ebp **ret** 10
4. **mov** eax, 10 **ret**
5. Nenhuma das anteriores



Q-3 [1.0 val.] Qual é o comportamento do seguinte código C?

```
char a[] = 'hello', *p;  
for (p = a; *p != '\0'; p++)  
    *p = (*p) + 1;
```

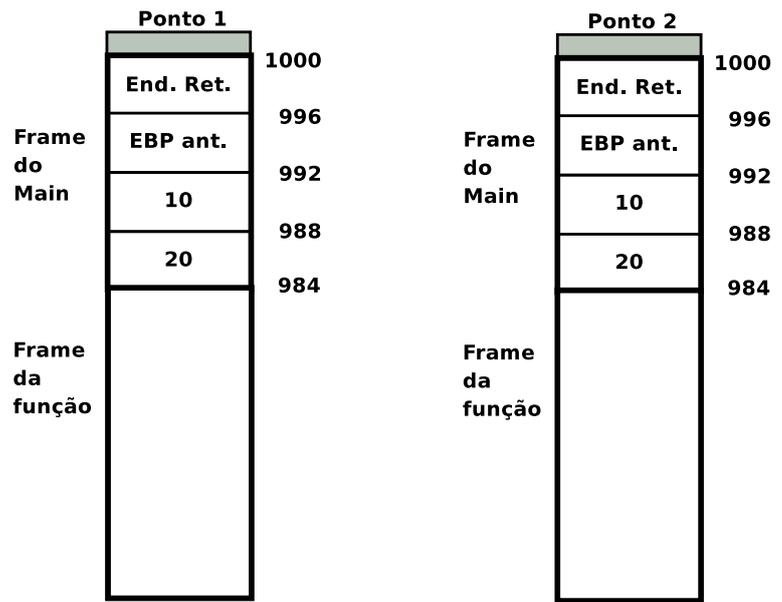
1. O ciclo executa normalmente, colocando em cada posição do array `a` o símbolo que, na tabela ASCII, sucede ao guardado nessa posição, por exemplo se `a[i] = 'a'` então o novo valor de `a[i]` será `'b'`
2. O ciclo executa normalmente, colocando em cada posição do array `a` o valor do elemento seguinte, por exemplo coloca em `a[1]` o valor de `a[2]`
3. O ciclo corre indefinidamente, pois a condição de paragem não está bem definida, devia ser `p != '\0'`
4. A condição inicial (`p = a`) está incorrecta, devia ser `p = &a`
5. Nenhuma das anteriores

Q-9 [2.0 val.] Considere o seguinte código C e assumo que pára a execução nos pontos 1 e 2 assinalados. Complete as figuras à direita, escrevendo qual é o conteúdo da pilha de execução (*stack*) em cada um dos pontos. Deve também indicar para onde apontam, nessa altura, os registos ESP e EBP.

```

void trocal(int a, int b) {
    int temp = a;
    a = b;
    b = temp; ← 2
}
void troca2(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp; ← 1
}
int main() {
    int x = 10, y = 20;
    trocal(y, x);
    troca2(&y, &x);
    return 0;
}

```



Q-10 [1.00 val.] Explique o porquê de se usar instruções diferentes para programar saltos condicionais sobre inteiros com e sem sinal. Por exemplo, **jb** e **ja**, respectivamente.

Q-11 [2.0 val.] Considere a seguinte sequência de acções em linguagem C e as sua numeração. Faça um esquema que ilustre cada acção numerada e use essa mesma numeração como legenda para a explicação.

```

(1) int *p;
    p = (int *) malloc (sizeof(int) * 20);
    (3)                               (2)
(4) p = p + 4;
(5) *p = 20;

```

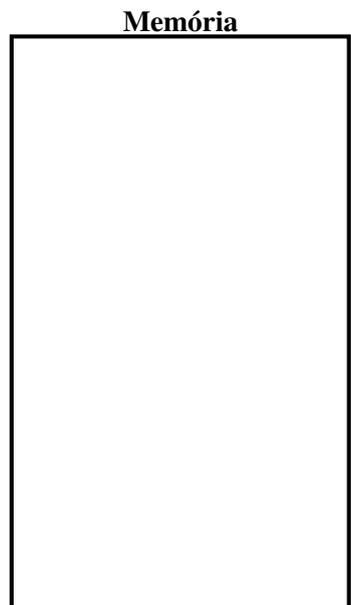
(1) _____

(2) _____

(3) _____

(4) _____

(5) _____



3 Prática - Assembly

Q-12 Esta pergunta **tem vários níveis de dificuldade**:

Para ter a cotação máxima (4 valores) deve usar as convenções de passagem de parametros e retorno de valores dada nas teóricas. Se usar qualquer outro método (à sua escolha) para passar parametros/retornar resultados, a pergunta será cotada para 80% (3.2 valores). Se não quiser usar funções pode responder tudo na alínea b), com um só programa equivalente ao que é pedido. Nesse caso terá no máximo a cotação da alínea b) (2.5 valores).

a) [1.5 val.] Programe em NASM IA/32, uma função `min`, que dados dois valores, **inteiros de 32 bits com sinal**, devolva o menor dos dois.

`min:`

b) [2.5 val.] Dado o seguinte programa (em pseudo código)

```
int v[21];
int m;
m = v[0];
for( int i = 0; i < 21; i++ )
    m = min(v[i],m);
```

apresente o pedaço de programa em NASM IA/32 correspondente a este código, incluindo a declaração das variáveis. Deve usar a função `min` definida na alínea anterior.

...

`section .bss`

`section .text`

`_start:`

```
mov eax, SYS_EXIT
mov ebx, 0
int LINUX_SYSCALL
```

4 Prática - Linguagem C

Q-13 Os espaços existentes no programa principal em C abaixo indicado deverão ser preenchidos de forma a definir uma matriz quadrada, de 8x8 inteiros que, depois de executada a função preenche, fique com os seguintes valores (onde v é um valor a "ler do teclado"):

```
v 0 0 0 0 0 0 0
v v 0 0 0 0 0 0
v v v 0 0 0 0 0
v v v v 0 0 0 0
v v v v v 0 0 0
v v v v v v 0 0
v v v v v v v 0
v v v v v v v v
```

a) [1.0 val.] O programa abaixo, "lê v do teclado", chama a função preenche e depois, imprime o somatório dos elementos da matriz. Preencha as caixas de forma a criar um programa correcto, obedecendo às especificações dadas.

```
void preenche(int matriz[8][8], int dim, int v);
[ ] somatorio([ ]);

int main() {
    int matriz [ ]; // complete a definição da variável matriz
    [ ] // e outra(s) que considere necessária(s)

    [ ]; // coloque aqui a instrução para "ler v do teclado"
    preenche([ ], [ ], [ ]); // coloque aqui os argumentos da função
    printf("Somatório: %[\n", somatorio([ ])); // coloque os itens em falta
    return 0;
}
```

Nota: Na função preenche o parâmetro dim que indica a dimensão da matriz, e v o valor a passar para o seu preenchimento.

b) [2.0 val.] Agora, implemente a função preenche para que esta inicialize a matriz da forma indicada.
Nota: Não tem de usar apontadores.

```
[ ] preenche ([ ])
```

```
}
```