

**Arquitectura de Computadores**  
**Licenciatura em Engenharia Informática**  
1º teste (B) – 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.]** Um endereço de memória:

1. tem sempre 32 bits
2. pode ter 8, 16 ou 32 bits
3. pode ter 16 ou 32 bits
4. tem se ser sempre guardado em registos específicos, como o EIP
5. Nenhuma das anteriores

---

**Q-2 [1.0 val.]** Considere a definição da seguinte variável: a: **resb 32**

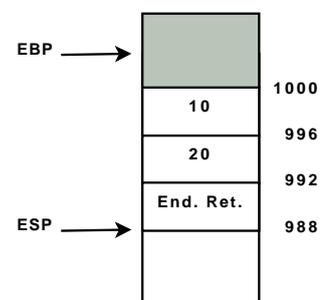
Qual das seguintes seqüências de instruções coloca o conteúdo do registo `cl` na última posição do array `a`.

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

---

**Q-3 [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-4 [1.0 val.]** No contexto da unidade de vírgula flutuante

1. O topo da pilha de operações é guardado no registo **FLAGS**
  2. Existem instruções da família **jump** que testam o conteúdo do registo **FSW**
  3. Os registos ST0 a ST7 têm 80 bits
  4. O estado da última operação efectuada é guardado no registo **FLAGS**
  5. Nenhuma das anteriores
-

**Q-5 [1.0 val.]** Dadas as seguintes variáveis (à esquerda) diga quais das instruções (à direita) são correctas quanto ao tipo de dados utilizado:

- |                  |                            |
|------------------|----------------------------|
| a: <b>dd</b> 321 | a) <b>mov</b> ah, [a]      |
| b: <b>db</b> 57  | b) <b>mov</b> ebx, [b]     |
| c: <b>dd</b> 1.0 | c) <b>mov dword</b> [a], 0 |
|                  | d) <b>fld dword</b> [a]    |
|                  | e) <b>fstp qword</b> [c]   |

1. b), c) e e)
2. b), d) e e)
3. a) e d)
4. c) e d)
5. Nenhuma das anteriores

---

**Q-6 [1.0 val.]** Considere o seguinte programa C guardado num ficheiro nome `prog.c`.

```
int f(void);
int main() {
    int x = f();
    return 0;
}
```

Qual é o resultado da seguinte linha de compilação: `gcc prog.c -o prog`

1. Um erro de ligação (“*linkagem*”) porque o código da função `f` não está disponível
2. Um erro de compilação porque o código da função `f` não está disponível
3. Um erro de compilação porque o protótipo da função `f` não está definido
4. Um erro de compilação porque a utilização da função `f` não está de acordo com o seu protótipo
5. Nenhuma das anteriores

---

**Q-7 [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