Nº de ordem: _____

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação
Microprocessadores e Computadores Pessoais
Teste 2

1º ano
2017-05-03
Duração 1:30
Sem consulta

Nome: _____ **Nº de estudante:** _____

Atenção: Este teste tem 12 questões em 5 páginas, num total de 200 pontos.

Parte I — Questões de Escolha Múltipla

Cada questão tem uma e uma só resposta certa. Respostas erradas não descontam.
Apenas as respostas assinaladas com × na grelha abaixo serão consideradas para efeitos de avaliação.

	Questão													
Opção	1	2	3	4	5	6	7	8a	8b	8c	8d	9	10	11
A		×		×					×					
B	×		×					×				×		
C					×					×			×	
D						×	×				×			×

Pontos: _____ / 140

[10] 1. Considere as duas declarações seguintes para uma mesma rotina:

```
ROT2 PROC STDCALL APT:PTR REAL4, LEN:DWORD
ROT2 PROC C APT:PTR REAL4, LEN:DWORD
```

Qual das seguintes afirmações é verdadeira?

A. Só haverá diferenças no prólogo das rotinas
B. Haverá diferenças no epílogo e no código gerado para a invocação
C. Só haverá diferenças no epílogo das rotinas
D. Haverá diferenças no prólogo e no código gerado para a invocação

[10] 2. Considere a seguinte declaração assumindo que, inicialmente, o conteúdo de `ESP` é múltiplo de 4:

```
ROT1 PROC USES EDI ESI P1:PTR BYTE, P2:WORD
     LOCAL X:WORD, Y:DWORD
```

Quantos bytes da pilha são usados pelo prólogo da rotina?

A. 20 B. 12 C. 16 D. 18

[10] 3. Assuma que VX e VY são do tipo `REAL4`. Após execução do fragmento de código seguinte, indique o que pode afirmar-se de VY relativamente a VX.

```
rcpss  xmm0, VX
mulss  xmm0, xmm0
sqrtss xmm0, xmm0
movss  VY, xmm0
```

A. $VY = |VX|$ **B. $VY = |1/VX|$** C. $VY = VX$ D. $VY = 1/VX$

[10] 4. Considere uma sub-rotina de tratamento de imagem semelhante às estudadas nas aulas práticas:

```
afunc1 proc pixels:ptr byte, largura:dword, altura:dword
```

Relembrando: Cada ponto da imagem (pixel) é representado por quatro bytes consecutivos, com o seguinte significado: 1. valor da componente B (azul), 2. valor da componente G (verde), 3. valor da componente R (vermelho) e 4. valor da transparência.

Qual dos seguintes fragmentos coloca o primeiro pixel da segunda linha da imagem a verde?

A.
```
mov ecx, largura
mov edi, pixels
shl ecx, 2
add edi, ecx
mov dword ptr [edi], 0000FF00h
```

B.
```
mov edi, pixels
add edi, largura
mov dword ptr [edi], 00FF0000h
```

C.
```
mov ecx, largura
mov edi, pixels
shl ecx, 1
add edi, ecx
mov dword ptr [edi], 0000FF00h
```

D.
```
mov ecx, largura
mov edi, pixels
shl ecx, 2
add edi, ecx
mov dword ptr [edi], 00FF0000h
```

[10] 5. Considere o fragmento de código seguinte:

```
  .data
seq1  byte  "aaa bbb ccc dddd",0

  .code
 cld
 xor ecx, ecx
 mov esi, offset seq1
```

```
.repeat
  lodsb
  .if (al!=' ')
    inc ecx
  .endif
.until al==0
dec ecx
```

Qual é o valor de `ECX` após a execução das instruções?

A. 14 B. 10 **C. 13** D. 3

[10] 6. Considere o seguinte código, correspondente ao epílogo de uma rotina:

```
POP EDI
POP ESI
LEAVE
RET 4
```

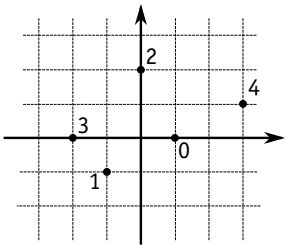
Qual das afirmações é verdadeira acerca da rotina?

A. Tem 4 parâmetros
B. Usa apenas os registos ESI e EDI
C. Usa 4 bytes de variáveis locais
D. Usa a convenção de chamada STDCALL

[10] 7. Um valor em memória é copiado do endereço de origem para o endereço de destino. Isso pode ser feito pela instrução `MOVSD` ou pela instrução `LODSD` seguida da instrução `STOSD`. Qual das seguintes afirmações é verdadeira acerca do registo `EAX`:

A. Ambas o alteram
B. Apenas `MOVSD` o altera
C. Nenhuma o altera
D. Apenas a sequência `LODSD/STOSD` o altera

8. Um conjunto de `n` pontos está disposto sobre uma grelha. A distância entre dois pontos é dada pelo número de "passos" mínimo necessário para ir de um ponto a outro "caminhando" sobre a grelha (uma unidade da grelha equivale a um "passo"). Por exemplo a distância entre o ponto 4 e o ponto 1 é de 6 "passos". Configuração exemplo de 5 pontos (0,. . . , 4) e sua representação em *assembly*:




```
.data
;coordenadas horizontais
coordX  SDWORD  1, -1, 0, -2, 3
;coordenadas verticais
coordY  SDWORD  0, -1, 2,  0, 1
```

Para este exemplo, a sub-rotina `rotSD` (apresentada abaixo) é invocada da seguinte forma:

```
invoke rotSD, offset coordX, offset coordY, 5, 2
```

```
rotSD PROC uses esi edi sX:PTR SDWORD,
       sY:PTR SDWORD, n:DWORD,
       ind_ponto:DWORD
  LOCAL pX: SDWORD, pY: SDWORD

  mov  eax, ind_ponto
  mov  esi, sX
  mov  edx, SDWORD PTR [esi+4*eax]
  mov  pX, edx
  mov  edi, sY
  mov  edx, [edi+4*eax]
  mov  pY, edx
  xor  eax, eax
  xor  edx, edx
```

```
  .WHILE (edx < n)
    mov  ecx, [esi+4*edx]
    sub  ecx, pX
    jns  @F
    neg  ecx            ; (*)
@@: add  eax, ecx
    mov  ecx, [edi+4*edx]
    sub  ecx, pY
    jns  @F
    neg  ecx
@@: add  eax, ecx
    inc  edx
  .ENDW
   ret
rotSD ENDP
```

[10] (a) Qual é o valor de `pY` durante a execução do ciclo `.WHILE/.ENDW`?
A. -1 **B. 2** C. 0 D. 1

[10] (b) Quantas vezes é executada a instrução da linha assinalada com `(*)`?
A. 2 B. 1 C. 4 D. 3

[10] (c) Assumir que o valor de `sY` é `0045AB00h`. No final da execução, qual é o valor de `EDI`?
A. `0045AB10h` B. `0045AB14h` **C.** `0045AB00h` D. `0045AB28h`

[10] (d) O objetivo da sub-rotina `rotSD` é:
A. Calcular a soma das distâncias entre todos os pontos exceto o ponto de índice `ind_ponto`.
B. Determinar o índice do ponto mais afastado do ponto de índice `ind_ponto`.
C. Determinar a distância do ponto mais afastado do ponto de índice `ind_ponto`.
D. Calcular a soma das distâncias do ponto de índice `ind_ponto` a todos os pontos.

[10] 9. Na linha de código `REPNE SCASD` a instrução `SCASD` é repetida enquanto:

A. `ECX!=0` ou `EAX==[EDI]`
B. `ECX!=0` e `EAX!=[EDI]`
C. `ECX!=0` e `EAX==[EDI]`
D. `ECX!=0` ou `EAX!=[EDI]`

[10] 10. Sabendo que `A, B, D, E` e `RES` são do tipo `real8` e `F` do tipo `sdword`, indique qual das seguintes sequências de instruções calcula

$$RES = \frac{(A+B)\times D}{(F-A)+E}$$

A.
```
movss xmm0,A
addss xmm0,B
mulss xmm0,D
cvtsi2ss xmm1,F
subss xmm1,A
addss xmm1,E
divss xmm0,xmm1
movss RES,xmm0
```

B.
```
movsd xmm0,A
addsd xmm0,B
mulsd xmm0,D
cvtsi2sd xmm1,F
addsd xmm1,E
subsd xmm1,A
divsd xmm1,xmm0
movsd RES,xmm1
```

C.
```
cvtsi2sd xmm1,F
subsd xmm1,A
addsd xmm1,E
movsd xmm0,A
addsd xmm0,B
mulsd xmm0,D
divsd xmm0,xmm1
movsd RES,xmm0
```

D.
```
cvtsi2ss xmm1,F
subsd xmm1,A
addsd xmm1,E
movsd xmm0,A
addsd xmm0,B
mulsd xmm0,D
divsd xmm0,xmm1
movsd RES,xmm0
```

[10] 11. Considere a execução do seguinte fragmento de código.

```
  .data
X REAL4 2.25
Y REAL8 ?
  .code
```

```
subsd  xmm0, xmm0
movss  xmm0, X
addss  xmm0, xmm0
movsd  Y, xmm0
```

Indique a afirmação verdadeira sobre o valor de Y.

A. Y=0.0 B. Y=4.5 C. Y=2.25 **D. Nenhuma das restantes opções é verdadeira.**

Parte II — Exercício de programação

Atenção: Responder em folha separada.

12. Um parque de estacionamento funciona 12 horas por dia (720 minutos). Inicialmente o parque está vazio. Duas sequências de `N` elementos do tipo `WORD`, `tin` e `tout`, contêm, respetivamente os instantes em que ocorreram entradas ou saídas de viaturas para um dia completo. Por exemplo a sequência $t_{\text{in}} = \{0, 0, 0, 0, 3, 7, 7, \dots\}$ indica que entraram 4 viaturas durante o minuto 0, uma viatura durante o minuto 3, duas durante o minuto 7, etc. Para a saída de viaturas é semelhante. As sequências estão ordenadas por ordem crescente. O tempo ($0 \leq$ tempo < 720) é medido em minutos decorridos após a abertura. Ao fim do dia o parque está vazio.

[35] (a) Escrever uma sub-rotina que determina quantas viaturas estão no parque no início do minuto `M` ($0 \leq$ `M` < 720). A sub-rotina tem o seguinte protótipo:

```
nviaturas PROTO tin:PTR WORD, tout:PTR WORD, N:WORD, M:WORD
```

Solução:

```
nviaturas PROC uses esi tin:PTR WORD, tout:PTR WORD, N:DWORD, M:WORD
    mov   dx, M
    xor   eax, eax      ; Número de viaturas
    mov   esi, tin      ; Processa entradas
    .while ([esi] < dx)
      inc   eax
      add   esi, type word
    .endw
```

```
    mov   esi, tout     ; Processa saídas
    .while ([esi] < dx)
      dec   eax
      add   esi, type word
    .endw
    ret
nviaturas ENDP
```

[25] (b) Sem recorrer à alinea anterior escrever uma sub-rotina que determina se em algum minuto do dia o número de viaturas no parque é negativo. A sub-rotina retorna 1 nesse caso ou 0 no caso contrário. Se existirem várias entradas e saídas no mesmo minuto, as entradas são processadas antes das saídas. A sub-rotina tem o seguinte protótipo:

```
coerente PROTO tin:PTR WORD, tout:PTR WORD, N:DWORD
```

Solução:

```
coerente PROC uses esi edi tin:PTR WORD, tout:PTR WORD, N:DWORD
    mov   esi, tin
    mov   edi, tout
    mov   ecx, N
@@: cmpsw       ; Se [esi]>[edi] (i.e., min. entrada > min. saída)
    ja    @F    ; não há coerência.
    loop  @B
    xor   eax, eax
    ret
@@: mov   eax, 1
    ret
coerente ENDP
```

Fim.