Exercícios sobre pilha e subrotinas

① (Opção B)

②

Endereço	Conteúdo
0x7010	0x05
0x7008	0x01
0x7000	0x03
0x6FF8	0x0E
0x6FF0	0x07

A) ~~0 1 2 3 4 5 6 7 8 9~~

B) ~~0 X 1 2 3 4 5 6 7 8 9~~

C) ~~0 1 2 3 4 5 6 7 8 9~~

D) ~~0 1 2 3 4 5 6 7 8 9~~

X2 = 0x0E

③ a) A sub-rotina POLI devolve em X0 o valor resultante da soma entre o quadrado do valor inicial em X0, e o produto do valor inicial de X0 por 3, e 1. Ou seja $x^2 + 3x + 1$

b) *** → stp X29, X30, [SP, #-16]!
mov X29, SP

*** → ldp X29, X30, [SP], #16

c) Antes de <1>: contém os valores dos registos X29 e X30

Antes de <2>: não contém nada.

4a)

```
.text
.global f4_e4a
.type f4_e4a, %function
f4_e4a: stp X29,X30,[SP,#-16]!
        mov X29,SP
ciclo   ldrb W2,[X0],#1
        cbz W2,fim
        str W2,[SP,#-16]!
        b ciclo
fim:    cbz X1, movo_fim
        sub X1, X1, #1
        ldr W2,[SP],#16
movo_fim: ldp X29,X30,[SP],#16
          ret
```

b)

```
.text
.global f4_e4b
.type f4_e4b, "function"
f4_e4b: STP X29,X30,[SP,#-16]!
        MOV X29,SP
        MOV X1,#0
ciclo:  LDRB W2,[X0],#1
        CBZ W2, correto
        CMP W2, '('
        B.eq inicial
        CMP W2, '['
        B.eq inicial
        CMP W2, '{'
        B.eq inicial
        CMP W2, ')'
        B.eq curvos
        CMP W2, ']'
        B.eq retos
        CMP W2, '}'
        B.eq chavetas
        B ciclo
inicial: STR W2,[SP,#-16]!
         ADD X1,X1,#1
         B ciclo
curvos: CBZ X1, falha
        LDRB W3,[SP],#16
        SUB X1,X1,#1
        CMP W3,'('
        B.ne falha
        B ciclo
retos:  CBZ X1, falha
        LDRB W3,[SP],#16
        SUB X1,X1,#1
        CMP W3,'['
        B.ne falha
        B ciclo
chavetas: CBZ X1, falha
          LDRB W3,[SP],#16
          SUB X1,X1,#1
          CMP W3,'{'
          B.ne falha
          B ciclo
```

```
falha: MOV X0,#1
repetir: CBZ X1, fim
         ADD SP,SP,#16
         SUB X1,X1,#1
         B repetir
correto: MOV X0,#0
fim: LDP X29,X30,[SP],#16
     RET
```

(5) a) extern int soma(int *a, int m) ← Em C
(a: X0, m: W1)

```
.text
.global soma
.type soma, "function"
soma:   STP X29, X30, [SP, #-16]!
        MOV X29, SP
        MOV X2, #0
ciclo:  CBZ W1, fim
        LDR W3, [X0], #4
        ADD X2, X2, X3
        SUB W1, W1, #1
        B ciclo
fim:    MOV X0, X2
        RET
```

b) extern int media (int *a, int n) ← Em C
(a: X0, n: W1)

```
.text
.global media
.type media, "function"
media:  STP X29, X30, [SP, #-16]!
        MOV X29, SP
        STP X0, X1, [SP, #-16]!
        BL soma
        MOV X2, X0
        LDP X0, X1, [SP], #16
        SDIV X0, X2, X1
        LDP X29, X30, [SP], #16
        RET
```

c) extern int maxmed (int *a, int b, int *c, int d)
(a: X0, b: W1, c: X2, d: W3)

```
.text
.global maxmed
.type maxmed, "function"
maxmed: STP X29, X30, [SP, #-16]!
        MOV X29, SP
        STP X20 X21, [SP, #-16]!
        MOV X20, X2
        MOV X21, X3
        BL media
        STR X22, [SP, #-16]!
        MOV X20, X0
        MOV X0, X20
        MOV X1, X21
        BL media
        CMP X0, X22
        B.GT fim
        MOV X0, X22
fim:    LDR X22, [SP], #16
        LDP X20 X21, [SP], #16
        LDP X29, X30, [SP], #16
        RET
```