

## Pilha e subrotinas

1. Suponha que se realiza uma sequência "misturada" de operações *push* e *pop*. As operações *push* usam os números de 0 a 9 por ordem; a seguir a cada operação de *pop* o resultado é impresso. Qual das seguintes sequências não pode ocorrer?

- A. 4 3 2 1 0 9 8 7 6 5
- B. 4 6 8 7 5 3 2 9 0 1
- C. 2 5 6 7 4 8 9 3 1 0
- D. 4 3 2 1 0 5 6 7 8 9

2. Assumir os seguintes conteúdos de memória

Endereço	Conteúdo
0x7010	0x05
0x7008	0x01
0x7000	0x03
0x6FF8	0x00
0x6FF0	0x02

e de registos: X0=0x0E, X1=0x07 e SP=0x7000.

Mostrar as alterações que ocorrem no conteúdo da memória e dos registos após cada uma das instruções da sequência.

```
STR X0, [SP, #-16]!
LDR X2, [SP], #16
STP X1, X0, [SP, #-16]!
```

3. Considerar o seguinte programa.

```
// ===== prog.c =====

extern long int POLI(long int x);
int main(void)
{
    long int r;
    ...
    r = POLI(7);
    printf("Resultado = %d\n", r);
    return EXIT_SUCCESS;
}

// ===== subrot.s =====

POLI: ...          // ***
        MOV X10, X0 // <1>
        BL  QUAD
        MOV X20, X0
```

```
MOV  X0, #3
MUL  X10, X10, X0
ADD  X0, X20, X10
ADD  X0, X0, #1
...      //  ***
RET    //  <2>

QUAD: MUL  X0, X0, X0
      RET
```

- a) Analise o programa e descreva o que calcula a sub-rotina POLI. Indique o que aparecerá no monitor após a execução.
  - b) Completar o código nos locais assinalados por \*\*\*.
  - c) Indique o conteúdo da pilha do sistema imediatamente antes da execução das instruções assinaladas com <1> e <2>.
4. Utilizando a pilha, escrever um programa que:
- a) imprime uma cadeia de caracteres por ordem inversa.
  - b) verifica se uma sequência de caracteres tem parêntesis curvos, parêntesis retos e chavetas corretamente emparelhadas.
5. Escrever e testar as seguintes sub-rotinas:
- a) SOMA – calcula a soma dos elementos de um vetor.
  - b) MEDIA – calcula a média dos elementos de um vetor utilizando a sub-rotina SOMA.
  - c) MAXMED – calcula a média dos elementos de dois vetores e retorna a maior das duas médias.

Fim