

Programação Imperativa – Exercícios de Controle de Fluxo de Execução

Prof. Alcides Calsavara – PUCPR

(B.1) Elabore um programa em C que leia dois valores inteiros: o primeiro define o limite inferior de um intervalo, enquanto o segundo o limite superior. Imprima todos os múltiplos de 3 no intervalo, incluindo os limites inferior e superior, se for o caso.

Testar as seguintes condições:

- 1) O limite inferior não pode ser maior que o limite superior
- 2) O limite inferior não pode ser igual ao limite superior.

O usuário deve ter até 3 chances para fornecer valores válidos para os limites inferior e superior. Se o usuário ultrapassar esse número de chances, o programa deve retornar o código 15 e encerrar.

IMPORTANTE: Os programas não podem conter o comando **break**, nem o comando **goto**. Também, não podem conter um comando **return** dentro de um bloco de repetição. Além disso, não podem usar array.

(B.2) Elabore um programa em C que calcule e imprima os n primeiros termos da **Sequência de Fibonacci**, onde n é um valor inteiro positivo fornecido pelo usuário. Muito embora a Sequência de Fibonacci seja, usualmente, definida de forma recursiva, o programa deve calcular os n termos por meio de um algoritmo iterativo, isto é, com uso de um comando de repetição. Mais especificamente, o programa deve utilizar o comando **for** da linguagem C, sendo que cada iteração do comando deve calcular e imprimir um termo da sequência. Não pode ser usado o comando **break**, nem o comando **continue**.

(B.3) Reescreva o programa abaixo fazendo a substituição do comando **switch** por um ou mais comandos **if** (e respectivo **else**, quando se aplicar).

```
#include <stdio.h>
int main()
{
    int k;
    char ch, cript;
    printf("Digite um número inteiro: ");
    scanf("%d", &k);
    getchar(); // consome o ENTER

    printf("Digite um caracter: ");
    ch = getchar();

    switch ( k )
    {
        case 1:
        case 2:
            cript = ch + 2;
            printf("%c\n", cript);
            break;
        case 3:
            ch = ch + 5;
        case 4:
            cript = ch - 2;
```

```
        printf("%c\n", crip);
        break;
default:
switch ( ch )
{
    case 'a': case 'e': case 'i': case 'o': case 'u':
        printf("Vogal\n");
        break;
    default:
        printf("Consoante\n");
}
break;
}
return 0;
}
```