

## Algoritmos e Programação de Computadores

Atividades a serem desenvolvidas nas sessões de Laboratório

### Sessão 4:

#### Objetivos:

1. Aprender a elaborar programas que utilizam as estruturas de repetição **while**, **for** e **do...while**, além de estruturas de repetição aninhadas.
2. Aprender a solicitar os dados do usuário, por meio de mensagens explicativas.
3. Aprender a imprimir os resultados para o usuário, usando mensagens explicativas.

#### Atividades:

1. O programa abaixo lê a variável SEXO. Enquanto o valor fornecido não for dentro do esperado (M,m,F,f) exibe uma mensagem de erro e solicita novamente a leitura. Quando o valor fornecido estiver correto, o programa lê a variável ANO, e escreve ANO e SEXO. Digite o programa abaixo.

```
#include <stdio.h>
int main () {
    int ano;
    char sexo;

    printf("Informe o sexo: ");
    scanf("%c",&sexo);
    while ((sexo != 'f') && (sexo != 'F') && (sexo != 'm') &&
           (sexo != 'M')) {
        printf("Erro! ");
        printf("Informe o sexo novamente: ");
        getchar();
        scanf("%c",&sexo);
    }
    printf("Informe o ano: ");
    scanf("%d",&ano);
    printf("Saida - Ano = %d e Sexo = %c", ano, sexo);
    getchar();
    getchar();
    return (0);
}
```

}

- a. a) Corrija todos os erros sintáticos.
- b. b) Insira as quebras de linha dentro dos printf's, onde necessário. Compile e execute novamente.
- c. c) Caso seja desnecessário você pode retirar os comandos getchar do seu programa.

## 2. Criar o programa abaixo

```
/* descrição: imprimir os números de 1 a 5, cinco vezes.
Entrada: não tem
Saída: 1
      1 2
      1 2 3
      1 2 3 4
      1 2 3 4 5    */
```

```
#include <stdio.h>
int main () {
    int i, j;

    for (j=1;j<=5;j++) {
        i = 0;
        while (i<j) {
            i++;
            printf ("%d ",i);
        }
        printf("\n ");
    }
    getchar();
    return (0);
}
```

- a. Corrija todos os erros sintáticos;
- b. Execute o programa duas vezes, das seguintes formas:
  - ¥ exatamente como está, observando a saída;
  - ¥ retire o comando `i = 0`, escrevendo-o **ANTES** do comando **for** e observe a saída;
- a. Substituir o comando **while** pelo **do...while**, da seguinte forma:

<pre>while (i&lt;j) {     i++;     printf ("%d ",i); }</pre>	<pre>do {     i++;     printf ("%d ",i); } while(i&lt;j);</pre>
--	---

- a. Execute novamente o programa observando a saída (que deve ser a mesma da execução utilizando o comando **while**).

**Em todos os programas abaixo, quando necessário, solicite os dados de entrada e imprima os dados de saída usando mensagens explicativas adequadas. Lembre-se de validar a entrada do usuário, informando-o de valores de entrada inadequados (valores numéricos inesperados).**

3. Imprima o valor de  $\sum_{i=1}^k i$ , com valores de k sendo fornecidos pelo usuário. Não se sabe a priori quantos valores de k serão informados, sendo que a indicação de final de leitura é dada pelo valor de k menor do que zero. (Dica: uma das alternativas é utilizar um **while** ou **do ... while**, e dentro dele um **for**.)

Execute o programa com os seguintes dados:

¥ k = 10	resposta correta: 55
¥ k = 5	resposta correta: 15
¥ k = 7	resposta correta: 28
¥ k = -1	final do programa

4. Imprima o valor de n! para um valor n, e repita este cálculo t vezes. Os valores de t e n deverão ser informados pelo usuário.

Execute o programa, com os seguintes dados:

¥ t = 4	
¥ n = 0	resposta correta: 1
¥ n = 1	resposta correta: 1
¥ n = 3	resposta correta: 6
¥ n = 5	resposta correta: 120

5. Imprima o valor de  $x^k$ , para valores de x e k (ambos maiores ou iguais a zero) informados pelo usuário. Informe quatro pares de valores. (Dica: utilize “**for (i=1;i<=4;i++)**” para ler os 4 pares de valores, dentro dele utilize **do... while** para garantir que os valores de x e k sejam  $\geq 0$ , e “**for (j=1;j<=k;j++)**” para implementar a exponenciação após a leitura de cada par de valores.)

Execute o programa, com os seguintes dados:

¥ x = 4, k = 2	resposta correta: 16
¥ x = 3, k = 4	resposta correta: 81
¥ x = 10, k = 0	resposta correta: 1
¥ x = 2, k = 8	resposta correta: 256

6. Modifique o problema anterior para, além do que foi solicitado:

- aceitar o coeficiente k negativo (exemplo:  $2^{-3}=1/2^3=1/8=0.125$ );
- aceitar os valores de x e k negativos (exemplos:  $(-2)^2=4$ ,  $(-2)^3=-8$ ,  $(-2)^{-2}=1/(-2)^2=1/4=0.25$ ).

7. Escreva um algoritmo que leia uma sequência de valores inteiros (assuma pelo menos

dois elementos na sequência) e determine a maior soma de dois elementos consecutivos. Escreva o valor da soma. O valor zero indica o fim da sequência e não entra nos cálculos da sequência.