UNISINOS - **UNIVERSIDADE DO VALE DO RIO DOS SINOS**
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS (C6/6) – Curso: Informática

LABORATÓRIO I – AULA : Programas com Laços – While/Repeat

Disciplina: Linguagem de Programação PASCAL **E-mail:** ***osorio@exatas.unisinos.br***
Professor responsável: ***Fernando Santos Osório*** **Web:**
Semestre: 2000/2 ***http://www.inf.unisinos.br/~osorio/lab1.html***
Horário: 63 **Xerox :** ***Pasta 54*** **– LAB. I (Xerox do C6/6)**

Lista de Exercícios - Lista número: 03 - Programas com Laços

⇒ ***Comando While/Do***

1. Faça um comando While equivalente ao descrito abaixo (que realize o mesmo tipo de procedimento e obtenha o mesmo resultado), mas sem utilizar o operador relacional "OR". Você pode usar outros operadores lógicos ou relacionais, exceto o "OR".

```
WHILE  ( Nota < 0.0 )  OR  ( Nota  > 10.0 )
DO  ReadLn (Nota);
```

2. Faça um programa que leia dois números inteiros, o primeiro é o valor inicial de um contador, e o segundo é o valor final do contador (testar se o valor inicial fornecido é inferior ao valor final). Usando o comando *While*, escreva na tela uma contagem que comece no primeiro número lido, escreva os números seguintes colocando sempre apenas um número em cada nova linha da tela, e terminando a contagem quando chegar ao valor final indicado.

3. Ler o nome de um aluno e suas duas notas A e B, e após calcular a média ponderada entre estas notas (A tem peso 1 e B tem peso 2). Repetir este procedimento para uma turma composta por cinco alunos, usando o comando *While*. Exemplo de tela de saída:

```
Entre com o nome do aluno: João da Silva
Entre com o grau A: 5.0
Entre com o grau B: 6.0
O aluno João da Silva tem uma média:5.66
```

4. Baseado no programa anterior, faça um novo programa de maneira que possamos trabalhar com turmas compostas por um número variável de alunos. Após calcular e imprimir a média de um aluno, exibir uma mensagem perguntando ao usuário se existem mais alunos (resposta: sim / não). Se tiver mais alunos, continuar o procedimento de leitura das notas e o cálculo da média até que o usuário responda 'não'. Usar o comando *While* e gerar uma saída conforme o exemplo de tela de saída abaixo:

```
Entre com o nome do aluno: João da Silva
Entre com o grau A: 5.0
Entre com o grau B: 6.0
O aluno João da Silva tem uma média:5.66
Continuar (sim/não) ? sim
```

5. Baseado no programa anterior, faça um novo programa de maneira a validar as notas fornecidas pelo usuário (notas devem ser valores positivos entre 0.0 e 10.0). Indicar ao usuário se a nota fornecida é inválida e pedir para fornecer uma nova nota, repetindo este processo até que o usuário informe uma nota correta. Usar um laço *While* na leitura das nota, e gerar uma saída conforme o exemplo de tela de saída abaixo:

```
Entre com o nome do aluno: João da Silva
Entre com o grau A: 15.3
ERRO: Nota inválida! Digite novamente a nota.
Entre com o grau A: 5.0
Entre com o grau B: 6.0
O aluno João da Silva tem uma média:5.66
Continuar (sim/não) ? não
```

6. Fazer um programa que calcule e imprima o fatorial de um número fornecido pelo usuário, usando o comando *While*. Repetir a execução do programa tantas vezes quantas o usuário quiser. Lembre-se que o resultado do cálculo de um fatorial pode ser um número "grande" (Exemplo: Fatorial de 8 = 40320). Exemplo de tela de saída:

```
Entre com um número: 5
O fatorial de 5 é 120
Outro número (sim/não) ? não
```

7. Escrever um programa que calcule todos os números inteiros divisíveis por um certo valor indicado pelo usuário, e compreendidos em um intervalo também especificado pelo usuário. O usuário deve entrar com um primeiro valor correspondente ao divisor e após ele vai fornecer o valor inicial do intervalo, seguido do valor final deste intervalo. Usar o comando *While*. Exemplo de tela de saída:

```
Entre com o valor do divisor: 3
Início do intervalo: 17
Final do intervalo: 29
Números divisíveis por 3 no intervalo de 17 à 29:
18  21  24  27
```

8. Faça um programa para o "jogo de adivinhar um número". O computador deve sortear um número entre 0 e 100 e pedir para o usuário tentar adivinhar este número. O usuário vai dizer o seu palpite, e o computador deve responder, se ele é maior ou menor que o número que ele sorteou. O programa termina somente quando o usuário acertar exatamente qual o número que o computador tinha sorteado, escrevendo uma mensagem de felicitações para o nosso usuário e indicando o número total de tentativas feitas. Dica: para gerar um número qualquer entre 0 e 100, use um comando como o deste exemplo indicado logo a seguir. Exemplo: numero_sorteado := random (100);

9. Faça um programa para o "jogo de adivinhar um número", mas invertendo os papéis desta vez. O computador que vai tentar adivinhar um número escolhido pelo usuário. O usuário deve escolher um número e para cada número apresentado pelo computador, responder se ele acertou, ou se o número apresentado é maior que o escolhido, ou se ele é menor que o escolhido. O programa termina quando o usuário responder que o computador acertou.

10. Faça um programa que obtenha e exiba na tela todos os números primos de 1 até 150. Os números primos são aqueles que só são divisíveis por 1 e por eles mesmos (exemplo: 1, 3, 5, 7, ...).

⇒ ***Comando Repeat/Until***

11. Faça um comando Repeat equivalente ao descrito abaixo (que realize o mesmo tipo de procedimento e obtenha o mesmo resultado), mas sem utilizar o operador relacional "And". Você pode usar outros operadores lógicos ou relacionais, exceto o "And".

```
REPEAT ReadLn (Nota);
UNTIL ( Nota >= 0.0 )  AND  ( Nota  <= 10.0 ) ;
```

12. Faça um programa que leia dois números inteiros, o primeiro é o valor inicial de um contador, e o segundo é o valor final do contador (testar se o valor inicial fornecido é inferior ao valor final). Usando o comando *Repeat*, escreva na tela uma contagem que comece no primeiro número lido, escreva os números seguintes colocando sempre apenas um número em cada nova linha da tela, e terminando a contagem quando chegar ao valor final indicado.

13. Fazer os exercícios do 3 ao 7 usando o comando REPEAT/UNTIL no lugar do while.

14. Refazer o exercício 5 considerando que a resposta do usuário a pergunta "continuar?" pode ser: 'S', 's', 'SIM', 'sim', 'Sim' (para continuar); 'N', 'n', 'NÃO', 'não', 'Não', 'NÃO', 'não' (para parar) e caso seja entrado algo diferente de continuar ou parar, avisar o usuário de que sua resposta foi inválida e ler novamente a resposta do usuário.

15. Apresentar na tela a tabuada de multiplicação dos números de 1 até 10. O programa deve exibir o resultado das multiplicações de 1x1, 1x2, ... até 1x10, pedir para o usuário teclar algo, e recomeçar com 2x1, 2x2, ... até 2x10, pedir novamente para teclar algo e seguir assim sucessivamente até chegar em 10x10.

16. Fazer um programa que leia o número total de bits usados para representar um valor, e exiba em seguida o maior valor inteiro e sem sinal que podemos representar com este número de bits. Lembre-se que um número binário com 8 bits pode representar valores entre 0 e 2^N-1 (255 = 2 na potência 8, menos 1 = 2*2*2*2*2*2*2*2 - 1).

17. Faça um programa que teste os reflexos do usuário de um computador. O programa deve inicialmente aguardar um certo tempo antes de começar uma contagem à partir de 1, sendo este tempo determinado de maneira aleatória. Ao terminar o tempo de espera inicial do programa, ele deve pedir para o usuário apertar uma tecla e começar imediatamente uma contagem à partir de 1, exibindo na tela os números a medida que for contando. Se o usuário pressionar uma tecla o programa deve terminar a contagem e exibir até onde ele conseguiu contar antes do usuário ter pressionado a tecla. Enquanto o usuário não apertar uma tecla o programa segue contando.

    Dicas: delay – Função do Pascal que permite esperar um certo tempo.
    random – Função do Pascal que permite escolher um número aleatório
    keypressed – Função do Pascal que permite saber se uma tecla foi pressionada ou não, sem no entanto ficar paralisado aguardando que uma tecla seja pressionada.