Universidade de Caxias do Sul (UCS)
Área de Ciências Exatas e Tecnologia
Disciplina: Programação de Computadores I

Linguagem de Programação C

Comando de Repetição While

Prof. Dr. Ricardo Vargas Dorneles

Comandos de Repetição

- São utilizados para que um comando ou conjunto de comandos sejam executados mais de uma vez.
- A linguagem C possui 3 comandos de repetição:
  - O comando ***for***
  - O comando ***while***
  - O comando ***do...while***

Comando While

- O formato do comando While é:

  while (condição) comando;

  ou

  while (condição)

  {

  Lista de comandos

  }

- onde ***condição*** é uma expressão lógica, como as condições do comando if.

Comando While

- Ao entrar no comando while a condição é avaliada
- Se ela é verdadeira, a lista de comandos é executada
- Os comandos dentro da repetição são executados até que a condição seja falsa
- Exemplo:

```
a=1;
while (a<4){
    printf("%d\n",a);
    a=a+1;
}
```

Teste de Mesa

- Uma coluna para cada variável e uma coluna para os comandos de escrita
- Os comandos são executados sequencialmente, um a um
- A cada atribuição, a coluna da variável que recebeu o valor deve ser atualizada
- A cada comando de escrita, a coluna das escritas é atualizada
- Esse formalismo é usado há 60 anos e funciona muito bem
- Não tente inventar algo diferente

Teste de Mesa - Exemplo

```
a=1;
while (a<4){
    printf("%d\n",a);
    a=a+1;
    }
```

a	esc
1	

Teste de Mesa - Exemplo

```
a=1;
while (a<4){
    printf("%d\n",a);
    a=a+1;
    }
```

a	esc
1	1

Teste de Mesa - Exemplo

```
a=1;
while (a<4){
    printf("%d\n",a);
    a=a+1;
    }
```

a	esc
1	1
2	

Teste de Mesa - Exemplo

```
a=1;
while (a<4){
    printf("%d\n",a);
    a=a+1;
    }
```

a	esc
1	1
2	

Teste de Mesa - Exemplo

```
a=1;
while (a<4){
    printf("%d\n",a);
    a=a+1;
    }
```

a	esc
1	1
2	2

E continua até concluir a simulação ...

Comando de Repetição

- Uma estrutura de repetição, como o ***while***, também é chamada de **laço de repetição** (ou ***loop***)
- Cada repetição dos comandos dentro do ***while*** também é chamada de **iteração**.
- Por essa razão algoritmos ou métodos que utilizam estruturas de repetição são ditos **iterativos**

Teste de mesa

Os trechos a seguir escrevem o que na tela?

a)

```
i=1;
while (i<=2){
    printf("%d\n",i);
    i=i+1;
}
```

b)

```
i=1;
while (i<=2){
    i=i+1;
    printf("%d\n",i);
}
```

Teste de mesa

Os trechos a seguir escrevem o que na tela?

c)

```
i=1;
while (i<3){
    i=i+1;
    printf("%d\n",i);
}
```

d)

```
i=1;
while (i<3){
    printf("%d\n",i);
    i=i+1;
}
```

Teste de mesa

Os trechos a seguir escrevem o que na tela?

e)

```
a=1;
b=2;
while (a<200){
a=a+b;
b=a+b;
printf("%d␣%d\n",a,b);
}
```

f)

```
a=100;
while (a>0){
    printf("%d\n",a%2);
    a=a/2;}
```

Teste de mesa

Os trechos a seguir escrevem o que na tela?

g)

```
a=1234;
b=0;
while   (a>0){
    b=b*10+a%10;
    a=a/10;
}
printf("%d\n",b);
```

h)

```
a=100;
d=2;
while (a>1){
    if (a%d==0){
        printf("%d\n",d);
        a=a/d;}
    else d=d+1;
    }
```

Exemplo 1

- Desenvolva um programa que escreva 1000 vezes o nome da nossa disciplina.
  - A solução para esse problema deve, de alguma forma, repetir 1000 vezes o comando printf("Programação de Computadores I\n")
  - Neste caso, utiliza-se uma estrutura de repetição com uma variável que controle as 1000 repetições

Exemplo 1

```
i=1;
while (i <=1000){
    printf (" Internacional !!!\ n");
    i=i+1;
}
```

- Nesse programa, a variável ***i*** tem a importante função de controlar o número de repetições, para que o comando de escrita seja executado exatamente 1000 vezes, nem uma a mais, nem uma a menos.
- A variável que controla o número de repetições dá-se o nome de **variável de controle**

Laço de Repetição - Variável de Controle

Um laço controlado por variável de controle deve conter 3 elementos:

1. A definição do valor inicial (inicialização) da variável de controle, que deve obrigatoriamente ser feita fora do laço, e apenas uma vez, no início
2. O incremento (ou decremento) da variável de controle
3. Uma condição de saída que resulte falsa quando o número de iterações desejadas for completada.

Exemplo 2

- Desenvolva um programa que mostre na tela todos os números pares entre 1 e 50.
    - Esse exercício pode ser resolvido por dois caminhos diferentes.
    - 1ª solução: gerar todos os números entre 1 e 50 e escrevendo somente os que forem pares

```
a=1;
while (a <=50){
    if (a %2==0)
        printf (" %d\n",a);
    a=a+1;
}
```

Exemplo 2

- 2ª solução: gerar escrever somente os números pares

```
a=2;
while (a <=50){
      printf (" %d\n",a);
      a=a +2;
}
// o incremento da variável que controla o laço pode ser diferente de 1
```

Erros Comuns

- Alguns erros comuns ao implementar laços controlados são:
  - Falta da inicialização da variável de controle
  - Inicialização da variável de controle dentro do laço
  - Falta do incremento (ou decremento) da variável de controle
  - Condição de saída incompatível com o valor inicial e incremento da variável de controle, o que fará com que nunca termine a execução do laço (o famoso "**entrar em loop**")

Exemplo 3

- Desenvolva um programa que leia 10 números e escreva os que forem pares.

```
i=1;
while (i<=10){
    scanf("%d",&N);
    if (N%2==0)
        printf("%d ",N);
    i=i+1;
}
```

Exercícios envolvendo contagem

- Desenvolva um programa que leia 30 valores e conte quantos estão no intervalo [10,20], escrevendo ao final essa informação
    - A estrutura desse algoritmo é semelhante ao anterior, sendo necessária uma variável para controlar a contagem de 30 vezes e uma variável para armazenar o valor lido
    - Além delas é necessário uma variável para controlar a contagem
    - Essa variável (contadora) deve iniciar com 0 e, a cada vez em que for necessário contar, deve ser incrementada de 1
    - Lembrando que o colchete representa um intervalo fechado (isto é, que inclui o limite do intervalo) e os parênteses representam um intervalo aberto (que não incluem o limite do intervalo)

Exercícios envolvendo contagem

```
cont=0;
i=1;
while (i<=30){
      scanf("%d",&N);
      if (N>=10 && N<=20)
            cont=cont+1;
      i=i+1;
}
printf("cont=%d",cont);
```

Exemplo 4

- Desenvolva um programa que leia números até que seja digitado um número negativo, e escreva todos que forem pares

```
scanf("%d",&N); // lê o primeiro valor
while (N>0){
    if (N%2==0)
        printf("%d ",N);
    scanf("%d",&N); // lê o próximo valor de N
}
```

Comando Break

- O comando break encerra a execução de um comando de repetição fazendo com que a execução continue a partir do comando imediatamente após o fim do comando de repetição
- Normalmente é usado dentro de um comando condicional
- Ex: Ler números até que sejam digitados 10 números pares ou um número igual a zero:

```
while (quant<10){
    scanf("%d",&n);
    if (n==0) break;
    quant=quant+1;
}
```