

# Aula 6: Estruturas de Repetição

Adriano Veloso

Algoritmos e Estruturas de Dados I - DCC/UFMG

# Repetição

- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3}$   

```
int main() {  
    printf("S = %f", 1+1/2+1/3);  
    return(0);  
}
```

# Repetição

- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}$
- ```
int main() {  
    printf("S = %f", 1+1/2+1/3+1/4);  
    return(0);  
}
```

# Repetição

- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3}$

# Repetição

- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3}$
- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}$

# Repetição

- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3}$
- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}$
- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}$
- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6}$
- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}$
- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8}$
- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9}$
- Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10}$

# Algoritmo com Repetição

Em vários momentos, torna-se necessário repetir um trecho (ou bloco) de comandos por um certo número de vezes.

# Algoritmo com Repetição

Em vários momentos, torna-se necessário repetir um trecho (ou bloco) de comandos por um certo número de vezes.

- Esses laços são conhecidos como “loops”



# Algoritmo com Repetição

Em vários momentos, torna-se necessário repetir um trecho (ou bloco) de comandos por um certo número de vezes.

- Esses laços são conhecidos como “loops”

Ao invés de se replicar trechos de código, usamos “loops” para repetí-los:

- Grande economia de espaço (programas muito menores)
- Facilidade de programação (ninguém quer replicar um trecho 100000000 de vezes)

# Algoritmo com Repetição

Em vários momentos, torna-se necessário repetir um trecho (ou bloco) de comandos por um certo número de vezes.

- Esses laços são conhecidos como “loops”

Ao invés de se replicar trechos de código, usamos “loops” para repetí-los:

- Grande economia de espaço (programas muito menores)
- Facilidade de programação (ninguém quer replicar um trecho 100000000 de vezes)
- A quantidade de repetições pode ser definida apenas em tempo de execução!

# Tipos de Repetição

- 1 Definidas e garantidas → o trecho de comandos será repetido um número pré-estabelecido de vezes

# Tipos de Repetição

- 1 Definidas e garantidas → o trecho de comandos será repetido um número pré-estabelecido de vezes
- 2 Indefinidas e sem garantia → o trecho de comandos será repetido um número indefinido de vezes, inclusive pode ser repetido para sempre.

# Repetição utilizando **for**

O comando de repetição *for* tem a seguinte sintaxe:

```
for(<valor inicial>; <condição de repetição>; <passo do laço>) {  
    bloco de comandos a ser repetido  
}
```

- Todos os argumentos são opcionais, mas sempre devem ser separados por **;**

# Repetição utilizando **for**

O comando de repetição *for* tem a seguinte sintaxe:

```
for(<valor inicial>; <condição de repetição>; <passo do laço>) {  
    bloco de comandos a ser repetido  
}
```

- Todos os argumentos são opcionais, mas sempre devem ser separados por **;**
- O “loop” **for(;;);** executa indefinidamente

# Exemplo

Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$

# Exemplo

Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$

```
int main(){  
    float S=0;  
    int n;  
    scanf("%d", &n);  
    for(int i=1;i≤n;i=i+1) {  
        S=S +  $\frac{1}{i}$ ;  
    }  
    printf("S = %f", S);  
    return(0);  
}
```



# Repetição utilizando **do-while**

O comando de repetição *do – while* tem a seguinte sintaxe:

```
do {  
    bloco de comandos a ser repetido  
} while(<condição de repetição>)
```

# Repetição utilizando **do-while**

O comando de repetição *do – while* tem a seguinte sintaxe:

```
do {  
    bloco de comandos a ser repetido  
} while(<condição de repetição>)
```

- A primeira execução sempre é realizada

# Repetição utilizando **do-while**

O comando de repetição *do – while* tem a seguinte sintaxe:

```
do {  
    bloco de comandos a ser repetido  
} while(<condição de repetição>)
```

- A primeira execução sempre é realizada
- A condição de repetição é qualquer expressão lógica
- A condição de repetição não é opcional

# Repetição utilizando **do-while**

O comando de repetição *do – while* tem a seguinte sintaxe:

```
do {  
    bloco de comandos a ser repetido  
} while(<condição de repetição>)
```

- A primeira execução sempre é realizada
- A condição de repetição é qualquer expressão lógica
- A condição de repetição não é opcional
- O “loop” **do{ ... }while(1);** executa indefinidamente

# Exemplo

Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$

# Exemplo

Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$

```
int main(){  
    float S=0;  
    int n;  
    scanf( "%d", &n);  
    int i=1;  
    do{  
        S=S +  $\frac{1}{i}$ ;  
        i=i+1;  
    }while(i≤n);  
    printf( "S = %f", S);  
    return(0);  
}
```

# Repetição utilizando **while**

O comando de repetição *while* tem a seguinte sintaxe:

```
while(<condição de repetição>) {  
    bloco de comandos a ser repetido  
}
```

# Repetição utilizando **while**

O comando de repetição *while* tem a seguinte sintaxe:

```
while(<condição de repetição>) {  
    bloco de comandos a ser repetido  
}
```

- A condição de repetição é qualquer expressão lógica
- A condição de repetição não é opcional



# Repetição utilizando **while**

O comando de repetição *while* tem a seguinte sintaxe:

```
while(<condição de repetição>) {  
    bloco de comandos a ser repetido  
}
```

- A condição de repetição é qualquer expressão lógica
- A condição de repetição não é opcional
- O “loop” **while(1){...}** executa indefinidamente

# Exemplo

Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$

# Exemplo

Calcule o valor de  $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$

```
int main(){  
    float S=0;  
    int n;  
    scanf( "%d", &n);  
    int i=1;  
    while(i≤n) {  
        S=S +  $\frac{1}{i}$ ;  
        i=i+1;  
    }  
    printf( "S = %f", S);  
    return(0);  
}
```

# O Comando **break**

O comando **break** interrompe a execução do “loop”, e posiciona o próximo para aquele logo depois do fim do “loop”

# O Comando **break**

O comando **break** interrompe a execução do “loop”, e posiciona o próximo para aquele logo depois do fim do “loop”

```
int main() {  
    while(1) {  
        ...  
        break;  
        ...  
    }  
    próximo comando;  
}
```

Contato

`adrianov@dcc.ufmg.br`