

Comandos de Repetição

Disciplina de Programação de Computadores I
Universidade Federal de Ouro Preto

Agenda

- Variável acumuladora
- Variável indicadora
- Variável contadora

Variável Acumuladora

- Quando precisamos ler uma quantidade desconhecida de números e fazer uma operação entre eles, utilizamos uma variável acumuladora.
- A variável acumuladora permite acumular a operação entre os números, a cada passo.
- Resolve o problema de não sabermos quantas variáveis seriam necessárias, já que não sabemos a quantidade de números a serem lidos.

Variável Acumuladora- Exemplo

- Problema: Ler um inteiro positivo n e, sem seguida, ler n números e apresentar a soma destes n números lidos.
- Não podemos criar n variáveis para depois somá-las!
- Solução: Utilizar uma variável acumuladora que, a cada iteração, acumule a soma dos números lidos até o momento.

Variável Acumuladora- Exemplo

Inicio

```
inteiro i, n, temp, soma
soma = 0 (variavel acumuladora)
escreva (Digite a quantidade de números:)
leia (n)
para(i de 1 até n, variando de 1 em 1)
    escreva (Digite um número:)
    leia (temp)
    soma = soma + temp
fim para

escreva (soma)

fim
```

2 4 16 20 9

i	n	temp	soma
1	5	2	0+2
2		4	0+2+4
3		16	0+2+4+16
4		20	0+2+4+16+20
5		9	0+2+4+16+20+9

Variável Acumuladora- Exemplo

```
#include <stdio.h>

int main(){
    int i, n, temp, soma;
    soma = 0;
    printf("Digite a quantidade de números:");
    scanf("%d", &n);
    for (i = 1; i <= n; i++) {
        printf("Digite o %do. número:", i);
        scanf ("%d", &temp);
        soma = soma + temp;
    }
    printf("soma = %d\n", soma);
    return 0;
}
```

Variável Indicadora

- Quando queremos testar se os elementos de um conjunto satisfazem uma propriedade, utilizamos uma variável indicadora (com característica booleana), da seguinte forma:
 - Inicialmente, assume-se que os objetos satisfazem a propriedade (ind = Verdadeiro)
 - Em seguida, utiliza-se um laço para percorrer todos os objetos e verificar se cada um, de fato, satisfaz a propriedade
 - Se algum objeto não satisfizer a propriedade, altera-se a variável indicadora para que reflita a nova situação (ind = Falso)

Variável Indicadora- Exemplo

- Exemplo: Dados n números em sequência, diga se eles estão em ordem crescente.
- Solução:
 - Utilizamos uma variável indicadora que indica se os números estão em ordem crescente
 - Utilizamos duas variáveis para ler os números e verificar se estão em ordem crescente
 - Lemos o primeiro número e utilizamos um laço para ler os demais números e testar se estão em ordem

Variável Indicadora- Exemplo

```
inicio
    inteiro i, n, anterior, atual
    booleano ordenado (indicadora)

    escreva(Digite o tamanho da sequência: )
    leia(n)

    i = 1
    escreva(Digite um número:)
    leia(anterior);
    ordenado = Verdadeiro (Assumimos números ordenados)
    i = i+1

    enquanto ( i <= n )

        escreva("Digite um número:)
        leia(atual)

        se(anterior > atual)
            ordenado = Falso
            pare
        fim se

        anterior = atual (Atualiza para próximo laço)
        i = i+1

    fim enquanto

    se (ordenado = Verdadeiro)
        escreva(Números ordenados);
    senão
        escreva(Números não ordenados)

fim
```

2 3 8 10 11
2 3 1 10 11

i	n	anterior	atual	ordenado
1	5	2		V
2		3	3	
3		8	8	
4		10	10	
5		11	11	
6				

i	n	anterior	atual	ordenado
1	5	2		V
2			3	
3		3	1	F

Variável Indicadora- Exemplo

```
#include <stdio.h>
int main(){

    int i, n, anterior, atual, ordenado;
    printf("Digite o tamanho da sequência: ");
    scanf("%d", &n);

    i = 1;
    printf("Digite o %do. número:", i);
    scanf("%d", &anterior);
    ordenado = 1;    // Assumimos números ordenados
    i++;

    while ( i <= n )
    {
        printf("Digite o %do. número: ", i);
        scanf("%d", &atual);

        if (anterior > atual)
        {
            ordenado = 0;
            break;
        }
        anterior = atual;    // Atualiza para próximo laço
        i++;
    }

    if (ordenado) {
        printf("Números ordenados.\n");
    }else{
        printf("Números não ordenados.\n");
    }
    return 0;
}
```

Variável Contadora

- Quando precisamos contar um número desconhecido de acontecimentos, utilizamos uma variável contadora.
- Esta variável permite que contemos, através de um laço, quantas vezes se deu um acontecimento.
- Acontecimentos: satisfação de uma propriedade, identificação de uma característica, etc.

Variável Contadora- Exemplo

- Problema: Dado um número n , dizer se ele é, ou não, primo.
- Número primo é aquele que só tem 1 e ele mesmo como divisores.
- Solução:
 - utilizar uma variável que conta quantos números, entre 2 e $(n-1)$, dividem n . Se algum número neste intervalo dividir n , então n não é primo.

Variável Contadora- Exemplo

inicio

inteiro n, divisor, divisores;

escreva (Digite o número a ser testado:)

leia(n)

divisor = 2 (Possível divisor atual de n)

divisores = 0 (Quantidade de divisores/Variavel Contadora)

enquanto (divisor <= n – 1) {

 se (resto da divisão de n por divisor = 0) {

 divisores = divisores + 1 (Se divisor divide n, incrementa divisores)

 pare

 fim se

 divisor = divisor + 1 // Obtém o próximo possível divisor de n

fim enquanto

se (divisores > 0)

 escreva(Não é primo!);

senão

 escreva(É Primo!);

fim

5 6

n	divisor	divisores
5	2	0
	3	
	4	
	5	

n	divisor	divisores
6	2	0+1

Variável Contadora- Exemplo

```
#include <stdio.h>

int main(){
    int n, divisor, divisores;

    printf("\nDigite o número a ser testado: ");
    scanf("%d", &n);    // Lê o número a ser testado
    divisor = 2;        // Possível divisor atual de n
    divisores = 0;      // Quantidade de divisores
    while (divisor <= n - 1) {
        if (n % divisor == 0) {
            divisores++;    // Se div divide n, incrementa divisores
            break;
        }
        divisor++;          // Obtém o próximo possível divisor de n
    }
    if (divisores > 0) {
        printf("Não é primo! Tem %d divisores.\n", divisores);
    } else {
        printf("É Primo!\n");
    }
    return 0;
}
```

Exercícios

Exercício 1: Fatorial com for e while

- 1) Faça um programa para ler um número inteiro e calcular o seu fatorial.

Entrada:

4

Saída:

24

Entrada:

6

Saída:

72

Exercício 1: Fatorial utilizando for

```
#include <stdio.h>
int main(int argc, const char * argv[]) {
    int n,i;
    long int fatorial = 1;

    printf("Digite o número:");
    scanf("%d",&n);

    for (i = 2; i<=n; i++) {
        fatorial = fatorial * i;
    }

    printf("\nO fatorial de %d é %ld.", n, fatorial);

    return 0;
}
```

Exercício 1: Fatorial utilizando while

```
#include <stdio.h>
int main(int argc, const char * argv[]) {
    int n,i;
    long int fatorial = 1;

    printf("Digite o número:");
    scanf("%d",&n);

    i=2;
    while(i<=n){
        fatorial = fatorial * i;
        i++;
    }

    printf("\nO fatorial de %d é %ld.", n, fatorial);

    return 0;
}
```

Exercício 2

- Codifique um programa que leia N valores, conte quantos destes valores são negativos e quantos são positivos e imprima estas informações.

Exercício 2 utilizando for

```
int main() {
    int n, i, neg = 0, pos = 0;
    float aux;
    printf("Quantos números devem ser lidos?");
    scanf("%d", &n);
    for(i = 1; i <= n; i++) {
        printf("Digite o %d%c número: ", i, 167);
        scanf("%f", &aux);

        if(aux < 0)
            neg++;
        else if (aux > 0)
            pos++;
    }
    printf("Total: %d negativos e %d positivos.\n", neg, pos);
    return 0;
}
```

Exercício 2 utilizando while

```
int main() {
    int n, i, neg = 0, pos = 0;
    float aux;
    printf("Quantos números devem ser lidos?");
    scanf("%d", &n);
    i=1;
    while(i <= n) {
        printf("Digite o %d%c número: ", i, 167);
        scanf("%f", &aux);

        if(aux < 0)
            neg++;
        else if (aux > 0)
            pos++;
        i++;
    }
    printf("Total: %d negativos e %d positivos.\n", neg, pos);
    return 0;
}
```

Exercício 3

- Escreva um programa que leia N valores e encontre o maior e o menor deles, mostrando o resultado.

Exercício 3

```
#include <stdio.h>
int main(){
    int valor, maior, menor, n, i=1;
    printf("Quantos valores devem ser lidos?");
    scanf("%d", &n);
    printf("Digite o %do valor: ", i);
    scanf("%d", &valor);
    maior = valor;
    menor = valor;
```

Exercício 3

```
for(i=2; i<= n; i++){
    printf("Digite o %do valor: ", i);
    scanf("%d", &valor);
    if(valor > maior)
        maior = valor;
    else if(valor < menor)
        menor = valor;
}
printf("\nMaior: %d e menor: %d\n", maior, menor);
return 0;
}
```


Exercício: do-while

4) Faça um programa que leia uma quantidade não determinada de números positivos, terminando quando o 0 (zero) for lido.

Calcule a quantidade de números pares e ímpares, a média de valores pares e a média geral dos números lidos.

```
#include <stdio.h>
int main(){
    int numero;
    int pares = 0, impares = 0;
    float media_pares = 0.0, media_geral = 0.0;
    int i = 0;
    do{
        printf("Digite um valor positivo ou 0 para terminar: ");
        scanf("%d",&numero);
        if(numero>0){
            if(numero%2 == 0){
                pares++;
                media_pares = media_pares + numero;
            }
            else
                impares++;
            media_geral = media_geral + numero;
        }
        if(numero<0)
            printf("Valor negativo. Digite novamente.\n");
        i++;
    }while(numero!=0);
```

```
printf("\nA quantidade de numeros pares e: %d\n",
    pares);
printf("A quantidade de numeros impares e: %d\n",
    impares);
printf("A media dos valores pares e: %.2f\n",
    media_pares/(i-1));
printf("A media geral dos valores e: %.2f\n\n",
    media_geral/(i-1));
return 0;
}
```

Referências Bibliográficas

- Material de aula do Prof. Ricardo Anido, da UNICAMP:
<http://www.ic.unicamp.br/~ranido/mc102/>
- Material de aula da Profa. Virgínia F. Mota:
<https://sites.google.com/site/viriniaferm/home/disciplinas>
- DEITEL, P; DEITEL, H. C How to Program. 6a Ed. Pearson, 2010.