Semana 05 – Aula Prática
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

```
#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 = 1)
  - 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 = 0)

Variável Indicadora - Exemplo

- Exemplo: Dados n números em sequência, diga se eles estão em ordem crescente.
- Solução:
  - Utilizamos uma variável contadora 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

```
#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);
	i++;
	ordenado = 1;                    // Assumimos números ordenados
	while ( i <= n )
	{
		printf("Digite o %do. número: ", i);
		scanf("%d", &atual);
		i++;
		if (anterior > atual)
		{
			ordenado = 0;
		}
		anterior = atual; // Atualiza para próximo laço
	}
	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

```
#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
                }
                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("Qiuantos 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.

```c
#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;
}
```

Exercício 2 : Cálculo de Série

2) Seja a seguinte série:

1, 4, 4, 2, 5, 5, 3, 6, 6, 4, 7, 7, ...

Escreva um algoritmo em C que seja capaz de gerar N termos dessa série.
Esse número N deve ser lido do teclado.

```c
#include <stdio.h>
int main(int argc, const char * argv[]) {
    int n, r1 = 1, r2 = 4, r0=3;
    printf("Digite o valor de N:");
    scanf("%d", &n);

    for (int i = 1; i<=n; i++) {
        int valor;
        if(i%3==1){
            valor = r1 + i/3;
        } else if(i%3==2){
            valor = r2 + i/3;
        } else {
            valor = r0 + i/3;
        }

        if(i==n){
            printf("%d.\n",  valor);
        }else{
            printf("%d, ", valor);
        }
    }
    return 0;
}
```

Exercício: Sequência de Fibonacci.

3) Faça um programa para imprimir N números da sequência de Fibonacci.
O valor de N deve ser lido do teclado.

Exemplo de sequência de Fibonacci:

Entrada
8

Saída
1 1 2 3 5 8 13 21

```c
#include <stdio.h>
int main(int argc, const char * argv[]) {
    int n, anterior = 0, proximo = 1;

    printf("Digite o valor de N:");
    scanf("%d", &n);

    for (int k = 1; k <= n; k++)
    {
        int temp;

        printf("%d ", proximo);

        temp = anterior + proximo;
        anterior = proximo;
        proximo = temp;
    }

    return 0;
}
```

Exercício: Cálculo do valor de Pi

4) O valor aproximado do número π pode ser calculado usando-se a série:

S = 1 – 1/33 + 1/53- 1/93 ...
π = (S * 32)1/3

Faça um algoritmo em C que calcule e imprima o valor de π usando os 51 primeiros termos da séria acima.

```
#include <stdio.h>
int main(int argc, const char * argv[]) {

    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/virginiaferm/home/disciplinas
- DEITEL, P; DEITEL, H. C How to Program. 6a Ed. Pearson, 2010.