MC-102 — Aula 07
Comandos Repetitivos:Exemplos

Instituto de Computação – Unicamp

22 de Março de 2012

Roteiro

1. Maior Número
2. Soma de n Números
3. Fatorial
4. Números Primos
5. Números de Fibonacci
6. Opcional: Decimal - Binário

Introdução

- Vimos quais são os comandos de repetição em C.
- Veremos agora alguns exemplos de utilização.

Maior Número

- Vamos fazer um programa que lê n números do teclado e informa qual foi o maior número impresso.
- O programa deve ter os seguintes passos:
  1. Lê um número n.
  2. Repete n vezes a leitura de um número determinando o maior.
- Como determinar o maior??

Maior Número

- A idéia e criarmos uma variável "maior"que sempre armazena o maior número lido até então.

```
Ler um número n
maior = recebe primeiro número
Repetir n-1 vezes
    aux = recebe número
    Se aux > maior então
        Atualiza o maior
```

Maior Número

```
int main(){
  int cont, n;
  double maior, aux;

  printf("\n Digite a quantidade de numeros:");
  scanf("%d",&n);

  printf("\n Digite numero:");
  scanf("%lf",&maior);
  cont = 2;
  while(cont<=n){
    printf("\n Digite numero:");
    scanf("%lf",&aux);
    if(aux>maior)
       maior = aux;
    cont++;
  }
  printf("\nO maior e:%.2lf\n",maior);
}
```

Soma de *n* Números

- Vamos fazer um programa que lê *n* números do teclado e informa a soma destes.
- Uma variável *soma* irá armazenar a soma dos números até um determinado ponto de execução.
- Ao ler um próximo número, como atualizar soma?
  - Basta fazer: *soma = soma + numero;*

Soma de *n* Números

```
int main(){
    int n, soma=0, aux;

    printf("Quantidade de numeros:");
    scanf("%d",&n);

    while(n>0){
        printf("Digite proximo numero:");
        scanf("%d",&aux);
        soma = soma + aux;
        n--;
    }

    printf("A soma dos numeros e: %d\n", soma);
}
```

Fatorial

- Vamos fazer um programa que lê um número inteiro positivo n do teclado e informa qual o seu fatorial.
- Lembrando: n! = n*(n-1)! = n*(n-1)*(n-2)! = n*(n-1)* ... *1!
- Por definição 0! e 1! são iguais a 1.
- O programa deve ter os seguintes passos:
  1. Lê um número n.
  2. Calcula $n * (n-1) * \ldots * 2 * 1$
- Como fazer este cálculo??
- Note que n não é fixo portanto temos que usar comandos repetitivos.

Fatorial - exemplo 5!

$$= (2 * 1) \quad (1)$$

$$\Downarrow$$

$$= (3 * (2 * 1)) \quad (2)$$

$$\Downarrow$$

$$= (4 * (3 * (2 * 1))) \quad (3)$$

$$\Downarrow$$

$$= (5 * (4 * (3 * (2 * 1)))) \quad (4)$$

Fatorial

- A idéia e criarmos uma variável "fat" que na n-ésima iteração do laço vale $fat = n!$.
- Assim, para calcular $(n+1)!$ podemos fazer $fat = (n+1) * n!$.
- Mas antes da atribuição $fat = n!$. Portanto faremos $fat = (n+1) * fat$

```
Ler um número n
cont = 1
fat = 1
Repetir n vezes
   fat = fat * cont
   cont = cont + 1
```

Fatorial

```
int main(){
  int cont, n;
  int fat;

  printf("\n Digite numero:");
  scanf("%d",&n);

  fat = 1;
  for(cont=1; cont<=n; cont++){
    fat = fat*cont;
  }
  printf("\nO fatorial e:%d\n",fat);
}
```

Fatorial

- No exemplo anterior o fatorial é calculado corretamente para $n \leq 16$.
- Se $n = 17$ o fatorial fornece um valor negativo!!!
- Por que??

Solução

- Um inteiro usa 32 bits para ser representado.
- Podemos trocar o tipo de *fat* para *unsigned int*
- Ou trocar para *double*!!

```
int main(){
  int cont, n;
  double fat;

  printf("\n Digite numero:");
  scanf("%d",&n);

  fat = 1;
  for(cont=1; cont<=n; cont++){
    fat = fat*cont;
  }
  printf("\nO fatorial e:%lf\n",fat);
}
```

Números Primos

- Um número é primo se seus únicos divisores são 1 e ele mesmo.
- Números primos são importantes para minha vida?
- http://www.numaboa.com.br/criptografia/chaves/350-rsa

Números Primos

- Um número é primo se seus únicos divisores são 1 e ele mesmo.
- Dado um número n como detectar se este é ou não primo??
  1. Lê um número n.
  2. Testa se nenhum dos números entre 2 e $\lfloor n/2 \rfloor$ divide n.
- Lembre-se que o operador % retorna o resto da divisão.
- Portanto $(a\%b)$ é zero se e somente se b divide a.

Números Primos

```
Ler um número n
cont = 2
Enquanto cont <= n/2 faça
    Se n%cont for igual a zero Então
        N não é primo
    cont = cont +1
```

Números Primos

```
int main(){
  int cont, n, eprimo;

  printf("\n Digite um numero:");
  scanf("%d",&n);


  cont = 2;
  eprimo=1;
  while( (cont<=n/2) && (eprimo) ){
    if(n%cont == 0)
      eprimo=0;
    cont++;
  }
  if(eprimo)
    printf("\nE primo!!\n");
  else
    printf("\nNao e primo!!\n");
}
```

Números Primos

```
int main(){
  int cont, n, eprimo;

  printf("\n Digite um numero:");
  scanf("%d",&n);
  if (n == 0 || n == 1)
     printf("\nNao e primo!!\n");
  else{


    código do slide anterior

}
```

Números de Fibonacci

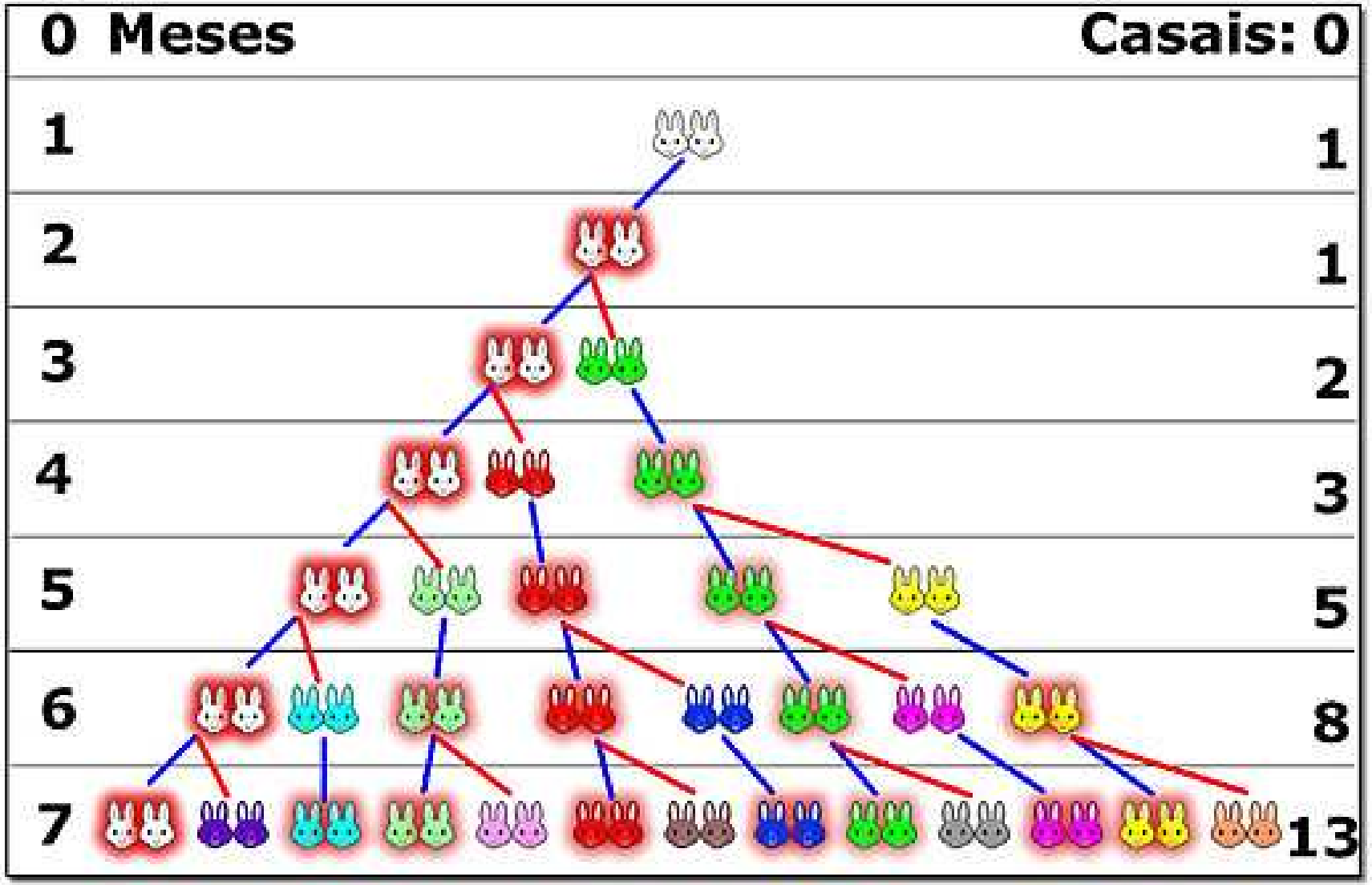
- A série de Fibonacci é: $1, 1, 2, 3, 5, 8, 13, \ldots$
- Ou seja o n-ésimo termo é a soma dos dois anteriores

$$F(n) = F(n-1) + F(n-2)$$

  onde $F(1) = 1$ e $F(2) = 1$.
- Vamos fazer um programa que imprime os primeiros n números da série.

Fibonacci e sua criação de coelhos

- http://www.bpiropo.com.br/fpc20070108.htm



Números de Fibonacci

- A série de Fibonacci é: $1, 1, 2, 3, 5, 8, 13, \ldots$
- Ou seja o n-ésimo termo é a soma dos dois anteriores

$$F(n) = F(n-1) + F(n-2)$$

  onde $F(1) = 1$ e $F(2) = 1$.
- Aplicações:
- http://pessoal.sercomtel.com.br/matematica/alegria/fibonacci/seqfib2.htm
- Vamos fazer um programa que imprime os primeiros n números da série.

Números de Fibonacci

```
Ler um número inteiro n (assume que é positivo)
contador = 1
f_atual = 1, f_ant = 0
Enquanto contador <= n faça
        Imprime f_atual
        aux = f_atual
        f_atual = f_atual + f_ant
        f_ant = aux
        contador = contador +1
```

Números de Fibonacci

```
int main(){
  int n, f_ant, f_atual, f_aux, cont;

  printf("\n Digite um numero:");
  scanf("%d",&n);

  cont = 1;
  f_ant=0; f_atual=1;
  while( cont<=n ){
    printf(" %d, ",f_atual);
    f_aux = f_atual;
    f_atual = f_atual + f_ant;
    f_ant = f_aux;
    cont++;
  }
  printf("\n");
}
```

Opcional: Representação Decimal-Binário

- Já sabemos que um computador armazena todas as informações na forma binária.
- É útil saber como converter valores decimais em binário.
- Dado um número em binário $b_n b_{n-1} \ldots b_2 b_1 b_0$, este corresponde na forma decimal à:

$$\sum_{i=0}^{n} b_i \cdot 2^i$$

- Exemplos:

$$101 = 2^2 + 2^0 = 5$$

$$1001110100 = 2^9 + 2^6 + 2^5 + 2^4 + 2^2 = 512 + 64 + 32 + 16 + 4 = 628$$

Opcional: Representação Decimal-Binário

- Vamos supor que lemos do teclado um inteiro binário.
- Ou seja, ao lermos $n = 111$ assumimos que este é o número binário (e não cento e onze).
- Como transformar este número no correspondente valor decimal (7 neste caso)??
- Basta usarmos a expressão:

$$\sum_{i=0}^{n} b_i \cdot 2^i$$

  Para isso porém devemos conseguir recuperar os uns e zeros individualmente.
- Fácil: $n\%10$ recupera cada um dos dígitos de n.
- Além disso $n = n/10$ remove o último dígito de n.

Opcional: Representação Decimal-Binário

```
int main(){
  int n, dec, pot, digito;

  printf("\n Digite um numero:");
  scanf("%d",&n);
  pot = 1; dec = 0;
  while( n != 0 ){
    digito = n%10;  //captura o dígito (a direita)
    dec = dec + (digito*pot);  //somatório
    n = n/10; //retira dígito previamente considerado
    pot = pot*2;
  }
  printf("\n %d\n",dec);
}
```

Opcional: Representação Decimal-Binário

- Agora dado um número em decimal como obter o correspondente em binário.
- Qualquer número pode ser escrito como uma soma de potências de 2:

$$5 = 2^2 + 2^0$$

$$13 = 2^3 + 2^2 + 2^0$$

- O que acontece se fizermos sucessivas divisões por 2 de um número decimal?

$$13/2 = 6 \text{ com resto } 1$$

$$6/2 = 3 \text{ com resto } 0$$

$$3/2 = 1 \text{ com resto } 1$$

$$1/2 = 1 \text{ com resto } 1$$

Opcional: Representação Decimal-Binário

```
int main(){
  int n, bin, aux, pot;

  printf("\n Digite um numero:");
  scanf("%d",&n);
  bin=0;
  pot=1;
  while( n > 0 ){
    aux = n%2;
    bin = bin + (aux*pot);
    n = n/2;
    pot = pot*10;
  }
  printf("\n %d\n",bin);
}
```

Exercício

- Faça um programa em C que calcule o máximo divisor comum de dois números m, n. Você deve utilizar a seguinte regra do cálculo do mdc com $m \geq n$

$$mdc(m, n) = m \text{ se } n = 0$$

$$mdc(m, n) = mdc(n, m\%n) \text{ se } n > 0$$