

Estrutura de repetição (Capítulo 4)

- 1) Faça um programa que leia 10 números digitados pelo usuário e exiba a soma e a média desses valores.
- 2) Altere o programa anterior, de tal maneira que o usuário informe a quantidade de números que serão digitados (ou seja, o valor “10” não deve ser fixo no programa).
- 3) Faça um programa que calcule a multiplicação de 2 números inteiros sem utilizar o operador “*”. Em vez disso, utilize apenas o operador de adição “+”.
- 4) Faça um programa que calcule o fatorial de um número. Se o número for menor do que zero, então o programa deverá informar ao usuário que o valor é inválido.
- 5) Faça um programa que solicite ao usuário dois números inteiros diferentes “n1” e “n2” e calcule a soma de todos os números ímpares dentro do intervalo definidor por [n1,...,n2]. Considere que n1 é sempre menor do que n2.
- 6) Altere o programa anterior de tal maneira que quando o usuário digitar um intervalo inválido (n1>n2), o programa irá solicitar novos valores para n1 e n2.
- 7) Faça um programa que leia um número inteiro positivo e determine se este é primo ou não. Por definição, um número é primo quando é divisível somente por si próprio e por 1.
- 8) Otimize o programa anterior com base nas seguintes considerações:
 - Números pares (com exceção do 2) não podem ser primos, visto que são divisíveis por 2. Se um número não for divisível por 2, não será divisível por nenhum outro número par. Portanto, com exceção do número 2, somente é necessário testar números ímpares.
 - É mais fácil que um número seja divisível por um número pequeno do que por um número maior. Portanto, se iniciarmos a procura do divisor de baixo para cima, ao invés de cima para baixo teremos chance de encontrar o número muito antes.
 - Nenhum número pode ser divisível por outro número maior que a metade dele. Portanto, não precisamos testar a divisibilidade dos números na faixa entre a metade e o próprio número.
- 9) Dado o valor de E, calcular $S=1 + 1/2 + 1/4 + 1/6 + \dots$ até que um termo da série seja menor do que E. O valor de E deverá ser digitado pelo usuário.

Resposta. Para E=0.00001 S=6.698

- 10) Faça um programa para calcular o valor de S, dado por:

$$S= 1/1 + 3/2 + 5/3 + 7/4 + \dots + 99/50$$

Resposta: 95.50

- 11) Faça um programa para calcular o valor de S, dado por:

$$S= 1/1 - 2/4 + 3/9 - 4/16 + \dots - 10/100$$

Resposta: 0.645635

- 12) Faça um programa para calcular e mostrar o valor de PI, usando a série:

$PI=4- 4/3 + 4/5 - 4/7 + 4/9 - \dots$ até que um termo seja menor do que 0.0001, em valor absoluto. Use a função “abs(numero)” para determinar o valor absoluto de um número.

13) Faça um programa usando o comando “for” para calcular o seguinte somatório:

$$\sum_{i=3}^n (5*i+2)$$

em que “n” é definido pelo usuário.

Resposta: Para N=10, somatório = 276

14) Os números naturais menores do que 10 e múltiplos de 3 ou 5 são: 3, 5, 6 e 9. A soma destes múltiplos é 23. Faça um programa que encontre a soma de todos os múltiplos de 3 ou 5 menores do que 1000.

Resposta: 233168

15) Cada novo termo da sequência de Fibonacci é gerado pela adição dos 2 termos anteriores. Ao iniciar a sequência com 1 e 2, os dez primeiros termos são: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... Faça um programa que encontre a soma dos números pares da sequência de Fibonacci cujo termo não exceda 4 milhões.

Resposta: 4613732

16) O número 2520 é divisível (resto zero) por todos números inteiros de 1 a 10. Faça um programa, em linguagem C, que encontre o menor número inteiro positivo divisível por todos os inteiros de 1 a 20.

Resposta: 232792560

17) Escreva um programa que solicita ao usuário um valor entre 3 e 18. Este valor representa a **soma** dos valores do lançamento de 3 dados. Em seguida, o programa deve imprimir todas as possibilidades de que no lançamento dos 3 dados a soma de seus valores seja igual ao valor informado pelo usuário. Por exemplo, caso o usuário digite o valor 10, o programa deverá exibir: 6,3,1; 1,6,3; 6,2,2; 5,3,2 etc. Você pode exibir sequências iguais em dados diferentes, por exemplo, as sequências 6,3,1 e 1,6,3 possuem os mesmos números, mas em dados diferentes.

18) Complete o código do programa abaixo. O programa deve apresentar o maior de 10 números digitados pelo usuário.

```
main(){
    int numero;
    int contador=0;
    do{
        contador=contador+1;
        printf("Digite o %do numero: ",contador);
        scanf("%d",&numero);
        /*****COMPLETE O PROGRAMA NESTE PONTO*****/
    }while(contador<10);
}
```

Se necessário adicione novas variáveis.

Sugestão: Digite o programa ao invés de copiar e colar usando o documento PDF.

19) Altere o programa anterior de tal maneira que seja exibido o maior e o menor número da sequência digitada pelo usuário.

20) Qual o menor número inteiro entre 1 e 10000 que possui a maior quantidade de divisores? Quantos e quais são os divisores? Implemente um programa para encontrar a resposta.

Resposta: menor número 7560 com 64 divisores. Exiba os divisores.

- 21) Implemente um programa para calcular o mínimo múltiplo comum de dois número inteiros utilizando o método da decomposição simultânea dos números em fatores primos.

$$\begin{array}{r|l}
 4, 15 & 2 \\
 2, 15 & 2 \\
 1, 15 & 3 \\
 1, 5 & 5 \\
 1, 1 &
 \end{array}$$

$$\text{MMC} = 2 * 2 * 3 * 5 = 60.$$

- 22) Implemente um programa solicita ao usuário a hora de início e a hora de término de um jogo, ambas subdivididas em 2 valores distintos: horas e minutos. O programa deve calcular e exibir a duração do jogo em minutos. Considere que o tempo máximo de duração de um jogo é de 24 horas e que o jogo pode começar em um dia e terminar no outro.

Teste seu programa com os seguintes valores:

| Início | Fim |
|--------|-------|
| 00:00 | 23:59 |
| 23:50 | 00:50 |
| 10:50 | 10:51 |
| 10:40 | 10:39 |

- 23) Implemente um programa que solicita ao usuário um valor inteiro correspondente à idade de uma pessoa em dias e informe-a em anos, meses e dias.

Obs.: apenas para facilitar o cálculo, considere todo ano com 365 dias e todo mês com 30 dias. Nos casos de teste nunca haverá uma situação que permite 12 meses e alguns dias, como 360, 363 ou 364. Este é apenas um exercício com objetivo de testar raciocínio matemático simples.

Exemplos para teste:

400 dias → 1 ano(s) 1 mes(es) 5 dia(s)

800 dias → 2 ano(s) 2 mes(es) 10 dia(s)

30 dias → 0 ano(s) 1 mes(es) 0 dia(s)

- 24) Implemente um programa que solicita ao usuário um valor real com duas casas decimais. Este valor representa um valor monetário. A seguir, calcule o menor número de notas e moedas possíveis no qual o valor pode ser decomposto. As notas consideradas são de 100, 50, 20, 10, 5, 2. As moedas possíveis são de 1, 0.50, 0.25, 0.10, 0.05 e 0.01. A seguir mostre a relação de notas necessárias.

Exemplos para teste:

=====

Valor: 576.73

NOTAS:

5 nota(s) de R\$ 100.00

1 nota(s) de R\$ 50.00

1 nota(s) de R\$ 20.00
0 nota(s) de R\$ 10.00
1 nota(s) de R\$ 5.00
0 nota(s) de R\$ 2.00
MOEDAS:
1 moeda(s) de R\$ 1.00
1 moeda(s) de R\$ 0.50
0 moeda(s) de R\$ 0.25
2 moeda(s) de R\$ 0.10
0 moeda(s) de R\$ 0.05
3 moeda(s) de R\$ 0.01
=====

Valor: 4.00

NOTAS:

0 nota(s) de R\$ 100.00
0 nota(s) de R\$ 50.00
0 nota(s) de R\$ 20.00
0 nota(s) de R\$ 10.00
0 nota(s) de R\$ 5.00
2 nota(s) de R\$ 2.00

MOEDAS:

0 moeda(s) de R\$ 1.00
0 moeda(s) de R\$ 0.50
0 moeda(s) de R\$ 0.25
0 moeda(s) de R\$ 0.10
0 moeda(s) de R\$ 0.05
0 moeda(s) de R\$ 0.01
=====

Valor: 91.01

NOTAS:

0 nota(s) de R\$ 100.00
1 nota(s) de R\$ 50.00
2 nota(s) de R\$ 20.00
0 nota(s) de R\$ 10.00
0 nota(s) de R\$ 5.00
0 nota(s) de R\$ 2.00

MOEDAS:

1 moeda(s) de R\$ 1.00
0 moeda(s) de R\$ 0.50
0 moeda(s) de R\$ 0.25
0 moeda(s) de R\$ 0.10
0 moeda(s) de R\$ 0.05
1 moeda(s) de R\$ 0.01

- 25) Faça um programa que leia dois inteiros: X e Z (devem ser lidos tantos valores para Z quantos necessários, até que seja digitado um valor maior do que X para ele). Conte quantos números inteiros devem ser somados em sequência (considerando o X nesta soma) para que a soma ultrapasse a Z o mínimo possível. Escreva o valor final da contagem.

A entrada pode conter, por exemplo, os valores 21 21 15 30. Neste caso, é então assumido o valor 21 para X enquanto os valores 21 e 15 devem ser desconsiderados pois são menores ou iguais a X. Como o valor 30 está dentro da especificação (maior do que X) ele será válido e então deve-se processar os cálculos para apresentar na saída o valor 2, pois é a quantidade de valores somados para se produzir um valor maior do que 30 (21 + 22).

Exemplo para teste:

3 1 20 → 5

26) Faça um programa que solicita ao usuário um valor A e um valor N . Imprimir a soma dos N números a partir de A (inclusive). Enquanto N for negativo ou ZERO, um novo N (apenas N) deve ser lido.

27) Na matemática, um número perfeito é um número inteiro para o qual a soma de todos os seus divisores positivos próprios (excluindo ele mesmo) é igual ao próprio número. Por exemplo o número 6 é perfeito, pois $1+2+3$ é igual a 6. Sua tarefa é escrever um programa que imprima se um determinado número é perfeito ou não.

Exemplos:

6 → perfeito

5 → não é perfeito

28 → é perfeito

28) Leia um conjunto não determinado de pares de valores M e N (parar quando algum dos valores for menor ou igual a zero). Para cada par lido, mostre a sequência do menor até o maior e a soma dos inteiros consecutivos entre eles (incluindo o N e M).

Exemplos:

5 2 -> 2 3 4 5 Soma=14

6 3 -> 3 4 5 6 Soma=18

5 0 -> Terminar o programa

29) Implemente um programa que solicita ao usuário 10 valores inteiros. Apresente o maior valor lido e a posição dentre os 10 valores lidos.