

Universidade Federal do Ceará - Campus Quixadá
QXD0010 – Estruturas de dados – Turma 02A
Curso de Ciência da Computação
Prof. Atílio Gomes

ATIVIDADE SOBRE RECURSIVIDADE

Entregue os exercícios de programação descritos neste documento até o dia **28/08/2019** (quarta-feira) pelo SIPPA¹. Leia com cuidado as instruções abaixo.

Instruções:

- Esta atividade é **individual** e deve ser implementado usando a linguagem de programação C++
- Essa atividade possui seis questões. Cada questão deve estar em um arquivo .cpp e deve compilar sem erros (**não aceitarei programas que não compilam**)
- Coloque todos os seus arquivos em uma única pasta e compacte-a com algum compactador (.zip, .rar, .gz, etc.) antes de enviar pelo SIPPA.
- Não se esqueça de colocar o seu **nome** e **matrícula** como comentário no seu código.
- Lembre-se de indentar corretamente o seu código para facilitar o entendimento.
- Os códigos-fonte devem estar devidamente organizados e documentados.
- Obs.: Se você usar alocação dinâmica de memória, lembre-se de desalocar os endereços de memória alocados quando os mesmos não forem mais ser usados.
- Antes de enviar um programa, teste-o. Cuidado com falhas de segmentação ou com chamadas recursivas sem caso base, ou com casos bases ausentes.
- Obs.: Qualquer indício de plágio resultará em nota **ZERO** para todos os envolvidos.

DICA: COMECE A ATIVIDADE O QUANTO ANTES.

¹SIPPA: <https://academico.quixada.ufc.br/sippa>

1. Escreva uma função recursiva chamada `soma_positivos` que calcule a soma dos elementos positivos de um vetor de inteiros $A[0..n - 1]$ com n elementos. O problema faz sentido quando $n = 0$? Quanto deve valer a soma neste caso?

Escreva uma função `main` que use a função `soma_positivos`.

Entrada: Um inteiro positivo n (que é o tamanho do vetor A), seguido de n inteiros (que são os valores do vetor). Cada inteiro da entrada deve ser lido em um linha separada.

Saída: um inteiro que é a soma dos elementos positivos do vetor A .

Exemplo de entrada e saída:

```
<<
5
-1
2
-4
5
-5
>>
7
```

2. Escreva uma função recursiva que calcule a soma dos dígitos decimais de um inteiro positivo. Por exemplo, a soma dos dígitos de 132 é 6. A sua função deve obedecer o seguinte protótipo: `int soma_digitos(int num)`

Entrada: Um inteiro positivo n .

Saída: a soma dos dígitos de n .

Exemplo de entrada e saída:

<<

4352

>>

14

3. Dado um vetor de inteiros A , imprima um triângulo de números tal que:
- na base do triângulo estejam todos os elementos do vetor original;
 - o número de elementos em cada nível acima da base é um a menos que no nível inferior;
 - e cada elemento no i -ésimo nível é a soma de dois elementos consecutivos do nível inferior.

Por exemplo, se fornecermos como entrada o vetor $A = \{1, 2, 3, 4, 5\}$, a saída do seu programa deve ser o seguinte triângulo:

```
48
20, 28
8, 12, 16
3, 5, 7, 9
1, 2, 3, 4, 5
```

Entrada: Um inteiro positivo n (que é o tamanho do vetor A), seguido de n inteiros (que são os valores do vetor). Cada inteiro da entrada deve ser lido em uma linha separada.

Saída: o triângulo de somas como definido acima.

Exemplo de entrada e saída:

```
<<
4
2
3
4
5
>>
28
12, 16
5, 7, 9
2, 3, 4, 5
```

4. O *coeficiente binomial* é uma relação estabelecida entre dois números naturais n e k , $n \geq k \geq 0$, definida do seguinte modo:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Escreva uma função recursiva que calcule o coeficiente binomial de dois números inteiros não negativos n e k , $n \geq k$.

Dica: Use a *relação de Stifel*:

$$\binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k}$$

Entrada: Dois inteiros n e k , tal que $n \geq k \geq 0$. Cada inteiro da entrada deve ser lido em um linha separada.

Saída: o valor $\binom{n}{k}$.

Dois exemplos de entrada e saída:

<<

12

3

>>

220

<<

15

8

>>

6435

5. Escreva uma função recursiva que calcule a diferença entre o valor de um elemento máximo e o valor de um elemento mínimo de um vetor de inteiros A com $n \geq 1$ elementos.

Entrada: Um inteiro positivo n (que é o tamanho do vetor A), seguido de n inteiros (que são os valores do vetor). Cada inteiro da entrada deve ser lido em um linha separada.

Saída: A diferença entre o valor de um elemento máximo e o valor de um elemento mínimo do vetor A

Exemplo de entrada e saída:

```
<<
5
-1
2
4
6
5
>>
7
```

6. Dados dois inteiros positivos n e k , imprima todos os números com k dígitos em ordem crescente de tal forma que os dígitos de cada número estejam entre os primeiros n números naturais e não sejam dígitos repetidos. Use recursividade para resolver essa questão.

Entrada: Dois inteiros positivos n e k , nesta ordem. Cada inteiro da entrada deve ser lido em um linha separada.

Saída: os números com k dígitos, como foram descritos acima, impressos em ordem crescente.

Exemplos de entrada e saída:

<<

3

2

>>

1 2

1 3

2 3

<<

5

5

>>

1 2 3 4 5

<<

5

3

>>

1 2 3

1 2 4

1 2 5

1 3 4

1 3 5

1 4 5

2 3 4

2 3 5

2 4 5

3 4 5