



**FUNDAÇÃO PRESIDENTE ANTÔNIO CARLOS  
FACULDADE PRESIDENTE ANTÔNIO CARLOS DE UBERLÂNDIA.**

---

## **ROTEIRO DE AULAS PRÁTICAS ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**Prof. Diego Alves da Silva**



**FUNDAÇÃO PRESIDENTE ANTÔNIO CARLOS  
FACULDADE PRESIDENTE ANTÔNIO CARLOS DE UBERLÂNDIA.**

---

**ROTEIRO DE AULA PRÁTICA**

**Curso: Análise e Desenvolvimento de Sistemas**

**Disciplina: Técnicas de Programação 1**

**Professor/a: Diego Alves da Silva**

**Período: 1º período**

**Unidade /SubUnidade: Delta**

**AULA PRÁTICA 5**

**1- Título da aula/Conteúdo: Utilizando conteúdo visto para solucionar exercícios de maratona.**

As maratonas de programação são formas de testar o aprendizado de algoritmos e a capacidade de resolução de problemas.

**2- Objetivos da aula:**

- Conhecer a sintaxe básica da linguagem C;
- Exercitar o pensamento a respeito de resolução de problemas;
- Testar o aprendizado obtido ao longo do semestre;

**3- Material utilizado:**

- Computador com editor de texto e compilador gcc instalados;
- Problemas propostos no site URI;

#### 4- Descrição do Procedimento:

O estudante irá ler e solucionar os problemas propostos de acordo com os exemplos de entrada e saída e as instruções propostas.

Os problemas devem seguir a seguinte estrutura:

```
1  #include <stdio.h>
2
3  int main() {
4
5      /**
6       * Escreva a sua solução aqui
7       * Code your solution here
8       * Escriba su solución aquí
9       */
10
11     return 0;
12 }
```

## 4.1 Problema 1 - Fibonacci, Quantas Chamadas? [1]

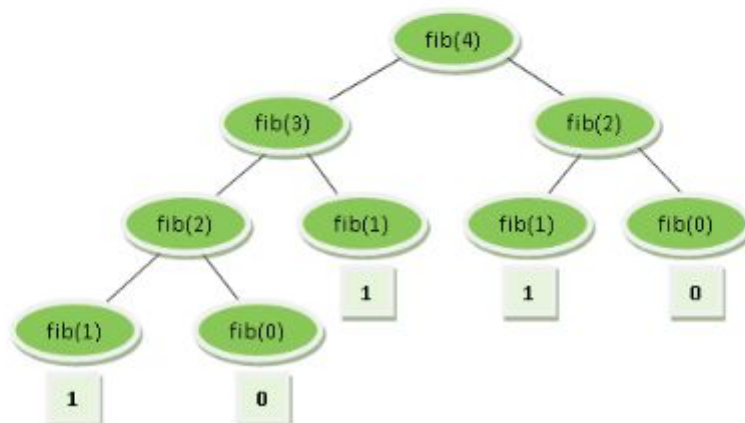
Quase todo estudante de Ciência da Computação recebe em algum momento no início de seu curso de graduação algum problema envolvendo a sequência de Fibonacci. Tal sequência tem como os dois primeiros valores 0 (zero) e 1 (um) e cada próximo valor será sempre a soma dos dois valores imediatamente anteriores. Por definição, podemos apresentar a seguinte fórmula para encontrar qualquer número da sequência de Fibonacci:

$\text{fib}(0) = 0$

$\text{fib}(1) = 1$

$\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2);$

Uma das formas de encontrar o número de Fibonacci é através de chamadas recursivas. Isto é ilustrado a seguir, apresentando a árvore de derivação ao calcularmos o valor  $\text{fib}(4)$ , ou seja o 5º valor desta sequência:



Desta forma,

- $\text{fib}(4) = 1+0+1+1+0 = 3$
- Foram feitas 8 calls, ou seja, 8 chamadas recursivas.

### Entrada

A primeira linha da entrada contém um único inteiro **N**, indicando o número de casos de teste. Cada caso de teste contém um inteiro **X** ( $1 \leq X \leq 39$ ).

### Saída

Para cada caso de teste de entrada deverá ser apresentada uma linha de saída, no seguinte formato:  $\text{fib}(n) = \text{num\_calls}$  calls = **result**, aonde **num\_calls** é o número de chamadas recursivas, tendo sempre um espaço antes e depois do sinal de igualdade, conforme o exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
2	$\text{fib}(5) = 14$ calls = 5
5	$\text{fib}(4) = 8$ calls = 3
4	

Link para problema original: <https://www.urionlinejudge.com.br/judge/pt/problems/view/1029>

## 4.2 Problema 2 - Criptografia [2]

Solicitaram para que você construísse um programa simples de criptografia. Este programa deve possibilitar enviar mensagens codificadas sem que alguém consiga lê-las. O processo é muito simples. São feitas três passadas em todo o texto.

Na primeira passada, somente caracteres que sejam letras minúsculas e maiúsculas devem ser deslocadas 3 posições para a direita, segundo a tabela ASCII: letra 'a' deve virar letra 'd', letra 'y' deve virar caractere 'l' e assim sucessivamente. Na segunda passada, a linha deverá ser invertida. Na terceira e última passada, todo e qualquer caractere a partir da metade em diante (truncada) devem ser deslocados uma posição para a esquerda na tabela ASCII. Neste caso, 'b' vira 'a' e 'a' vira ''.

Por exemplo, se a entrada for "Texto #3", o primeiro processamento sobre esta entrada deverá produzir "Wh{wr #3". O resultado do segundo processamento inverte os caracteres e produz "3# rw{hW". Por último, com o deslocamento dos caracteres da metade em diante, o resultado final deve ser "3# rvzgV".

### Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém um inteiro **N** ( $1 \leq N \leq 1 \cdot 10^4$ ), indicando a quantidade de linhas que o problema deve tratar. As **N** linhas contém cada uma delas **M** ( $1 \leq M \leq 1 \cdot 10^3$ ) caracteres.

### Saída

Para cada entrada, deve-se apresentar a mensagem criptografada.

Exemplo de Entrada	Exemplo de Saída
4 Texto #3 abcABC1 vxpdylY .ph vv.xwfxo.fd	3# rvzgV 1FECedc ks. \n{frzx gi.r{hyz-xx

Link para problema original: <https://www.urionlinejudge.com.br/judge/pt/problems/view/1024>

## **5-Avaliação:**

Toda a atividade será supervisionada pelo professor, que dará o suporte necessário para o aprendizado do aluno.

## **Análise e Discussão**

1. Quais problemas foram solucionados?
2. Submeta o problema no site e verifique a resposta do mesmo.
3. Quais foram as suas principais dúvidas?

## **6-Referências Bibliográficas**

[1] URI Online Judge. Disponível em:

<<https://www.urionlinejudge.com.br/judge/pt/problems/view/1029>>. Acesso em: 14 fev. 2019.

[2] URI Online Judge. Disponível em:

<<https://www.urionlinejudge.com.br/judge/pt/problems/view/1024>>. Acesso em: 14 fev. 2019.