

Fundamentos da Programação

Prova 1 - segunda chamada

1. (3,0 pontos) Conversão para a base 3

A conversão de um número para a base 3 é semelhante ao procedimento para a base 2

$$\begin{array}{r} 404 \quad | \quad 3 \\ \hline 2 \quad 134 \quad | \quad 3 \\ \hline \quad 2 \quad 44 \quad | \quad 3 \\ \hline \quad \quad 2 \quad 14 \quad | \quad 3 \\ \hline \quad \quad \quad 2 \quad 4 \quad | \quad 3 \\ \hline \quad \quad \quad \quad 1 \quad 1 \quad | \quad 3 \\ \hline \quad \quad \quad \quad \quad 1 \quad 0 \end{array}$$

$\Rightarrow 112222$

Quer dizer, nós dividimos o número sucessivamente por 3, até que o quociente seja igual a zero. E daí, nós escrevemos os restos na ordem contrária em que eles foram obtidos.

Para implementar esse procedimento como um programa em C, nós podemos

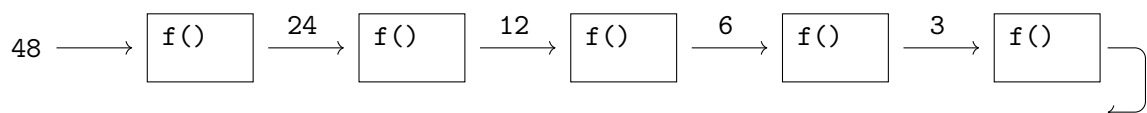
1. utilizar o comando `while` para repetir a operação de divisão, enquanto o quociente é maior do que 0.
2. armazenar os restos em uma lista auxiliar, que depois é impressa de trás para frente

Faça um programa que implementa a conversão para a base 3 utilizando essas ideias.

2. (2,0 ponto) Fatoração recursiva

- a) Faça uma função recursiva que divide um número n (fornecido como parâmetro) repetidamente, até que seja obtido um número ímpar.

Por exemplo,



Sua função deve retornar o número de vezes que a divisão foi realizada.

Nota: Evite utilizar variáveis globais; faça toda comunicação por meio da passagem de parâmetros e valores de retorno da função.

- b) Modifique a função para que ela faça a divisão por 3 repetidamente, até que o número não seja mais divisível por 3.

Depois modifique a função para que ela faça a divisão por um número k qualquer, que é passado como parâmetro para a função.

Nota: Você só precisa entregar essa última versão.

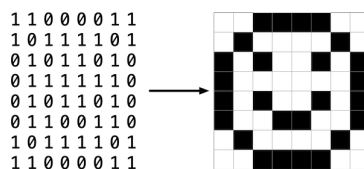
- c) Utilize a função do item (b) para produzir (e imprimir) a fatoração completa de um número n qualquer.

Nota: Você pode escrever um laço que realiza chamadas sucessivas à função recursiva, variando o valor de k .

3. (3,0 pontos) Mapas de bits

Um *mapa de bits* é uma representação de uma imagem (em preto e branco) na forma de uma matriz de 0's e 1's.

Por exemplo,



Suponha que

```
int Im[N][N];
```

é um mapa de bits que representa uma imagem qualquer.

E suponha que

```
int p[L][L];
```

é um mapa de bits que representa um padrão que pode aparecer na imagem — (*por exemplo, a carinha ilustrada acima*)

Faça um programa que encontra todas as ocorrências do padrão p na imagem Im .

Nota: Cada ocorrência deve ser indicada pela localização (x,y) onde o padrão aparece, que corresponde ao canto superior esquerdo do padrão.

4. (3,0 pontos) Banco de dados

Imagine que M é uma tabela de médicos, com o nome e o código de cada um deles

M	mara	juca	lia	ivo	teca
	72	77	41	39	81

Imagine que P é uma tabela de pacientes, com o nome e o código de cada um deles

P

hugo	lana	neco	fafá	toni	leia	dudu	ana
10	11	12	15	18	19	21	23

E imagine que **C** é uma tabela de consultas, com o código do médico e o código do paciente que fizeram essa consulta

T

72	77	77	39	41	72	77	81	81	39
18	23	21	12	15	18	23	10	11	12

Faça um programa que, dado o código de um médico, lista os nomes de todos os pacientes que tiveram consulta com ele.

Nota: Observe que um paciente pode ter feito diversas consultas com o mesmo médico, mas o seu nome deve ser listado apenas uma vez.