

Problemas da semana 11

Recursão

Problema 1 - Somando os dígitos

Crie uma função recursiva com a seguinte assinatura para calcular a soma dos dígitos de um número inteiro:

```
int somaDigitos(int n);
```

Essa função deve resolver a soma dos dígitos com chamadas recursivas, não vale usar estruturas de repetição.

A função main deve ler um número inteiro e escrever na tela o resultado da soma dos dígitos utilizando a função somaDigitos. Por exemplo, somaDigitos(1573) deve retornar 16, somaDigitos(-157) deve retornar 13 e somaDigitos(-15) deve retornar 6.

Exemplos

Input	Output
1573	16
-157	13
-15	6

Problema 2 - Função recursiva para calcular potência é uma potência

Escreva uma função recursiva para calcular uma exponenciação com a seguinte assinatura. Sua função não deve tratar o caso em que o expoente é negativo.

```
int potencia(int base, int expoente);
```

Problema 3 - Razão áurea

A razão áurea é uma constante matemática que representa uma relação específica entre duas variáveis a e b: a está para a + b, assim como b está para a. Ou seja:

$$\frac{a+b}{a} = \frac{a}{b}$$

cujos valor é: 1.6180339887...

Essa proporção é bastante utilizada em artes, no design e na arquitetura como a proporção ideal entre medidas. Essa razão também pode ser obtida através da seguinte fórmula:

$$\phi = 1 + \frac{1}{1 + \frac{1}{1 + \dots}}$$

Escreva a seguinte função recursiva para aproximar o valor da proporção áurea, onde numFracoes é o número de frações da fórmula acima:

```
double phi(int numFracoes);
```

A função main deve escrever na tela o valor aproximado de ϕ com 10 casas de precisão. Não há dados de entrada. Use **double**.

Problema 4 - Número de ocorrências no vetor (versão recursiva)

Escreva uma função recursiva que receba como parâmetros um ponteiro para inteiro contendo o endereço base de um vetor de **n** inteiros, o inteiro **n**, um inteiro **x** e retorne o número de ocorrências de **x** no vetor.

```
int numeroOcorrenciasVetor(int *v, int n, int x);
```

A função main deve ler da entrada padrão o valor de **n**, **n** números inteiros e, em seguida o número **x**. Depois o programa deve escrever na tela o número de ocorrência de **x** nos **n** números inteiros.

Os problemas nas páginas a seguir são bônus

Problema 5 - Ataque de aliens (bônus, pesquisa sobre flood fill)

Uma nave tripulada por humanos foi invadida por aliens. Para desespero dos tripulantes os aliens são bons em se esconder e atacar no momento inesperado. Felizmente a nave é equipada com sensores capazes de identificar criaturas, seja humano ou alien. O seu programa vai ler o mapa da nave e escrever na tela qual dos tripulantes (identificados pelos números de 1 a 3) estão em uma mesma sala de um alien (identificados por @). Caso nenhum esteja na mesma sala de um alien, o programa deve escrever 'salvos'.

```
#####  
#  #  #  #  #  
#      #2 #3#  
#1 #  #  #  #  
#### #  #  #  
#@      #@ #  #  
#####
```

Figure 1: Exemplo 1: o programa deve escrever 1 2

- Um # identifica uma parede ou uma porta
- Cada linha contém no máximo 30 caracteres
- Todos os caracteres da borda do mapa são necessariamente #
- Não há limites para a quantidade de aliens, mas há sempre 3 tripulantes
- O mapa é sempre encerrado com uma linha contendo FIM
- Você pode começar programando pelo q5inicial.c disponibilizado pelo professor e que já faz a leitura do mapa
- Não há mobilidade na diagonal (vide figura abaixo)

```
#####  
##    #  
#  #  #  
#  #  #  
#    ##  
#####
```

Figure 2: Neste mapa há duas salas e não apenas uma, pois não é possível atravessar paredes através da diagonal.

Problema 6 - Experiência bizarra (bônus, estude sobre backtracking no material do Prof. André)

Você está com um programa (mágico) que visualiza criaturas a partir do código genético. Esse programa recebe como entrada uma string de até 100 caracteres representando o DNA da criatura. Cada setor do DNA é responsável por uma característica da criatura. Querendo saber qual o efeito de determinado setor na criatura, você resolveu criar um programa que escreva na tela todas as combinações possíveis para aquele setor.

O seu programa deve ler uma string com as letras do DNA (C, G, T ou A) e dois inteiros que representam o índice do começo do setor e o índice do fim do setor (considere que o índice começa em 0). O programa deve então escrever na tela todas as combinações mudando as letras desse setor.

Por exemplo, na string abaixo, identificaríamos o setor em destaque com dois inteiros: 2 e 6.



Figure 3: Exemplo de sequência de DNA



Figure 4: Resultado depois de misturar uma abelha, um morcego e um elefante

Exemplos

Input	Output
ATCGGTCTCAAGC	ATAAAAATCAAGC
	ATAAAACTCAAGC
	ATAAAAGTCAAGC
	...
	ATTTTTTTCAAGC