INF01202 - Algoritmos e Programação

Semana 08 - Aula Prática

Prof. Vinícius Garcia Pinto

03-05-2019

Instruções

Para cada questão abaixo, elabore um algoritmo adequado para a resolução do problema. Em seguida, implemente seu algoritmo na linguagem C. Responda cada problema em um arquivo `problemaX.c` (troque X pelo nº do problema) incluindo um cabeçalho com o nome completo do aluno e o número do cartão UFRGS.

Exemplo de arquivo a ser enviado:

```c
//  Nome do Aluno: Meu nome completo
//  Cartao UFRGS:  00XXXXXX

/* Breve descrição sobre o problema e sobre o que faz
o código. */

#include<stdio.h>

int main(){

  // Solução do problema

  return 0;
}
```

Forma de envio

Enviar cada resposta separadamente em um arquivo `.c` nomeado `problemaX.c`, onde `X` deve ser substituído pelo número do problema. Os arquivos devem ser enviados pelo Moodle Acadêmico (`http://moodle.ufrgs.br`).

Verificação anti-plágio

A detecção de plágio em qualquer atividade implicará penalidades (nota zero) a todos os envolvidos!

- **todos** os materiais entregues (práticas e trabalho) são submetidos a verificação anti-plágio

Algumas dicas & erros recorrentes

A lista de dicas e erros recorrentes está disponível em: `https://github.com/viniciusvgp/intro-prog-c`

Problema 1

Uma matriz quadrada é dita triangular quando os elementos acima ou abaixo da diagonal principal são zero(s). A matriz é triangular superior (1) se todos os elementos abaixo da diagonal principal são zero ($\forall i > j, a_{ij} = 0$). Se todos os elementos acima da diagonal principal forem zero ($\forall i < j, a_{ij} = 0$), então a matriz é triangular inferior (2). Se ambas as condições forem verdadeiras então a matriz é dita matriz diagonal (3). Faça um programa C que deve **gerar e imprimir** matrizes triangulares de tamanho 10×10 conforme a escolha do usuário. As matrizes devem ser preenchidas de números inteiros aleatórios entre 0 e 499. O programa deve **continuar** a gerar matrizes **até que o usuário decida sair**.

- Dicas:
  - utilize `%03d` para exibir os valores com três dígitos, completando com zeros caso necessário.
  - utilize a função `rand()` da biblioteca `stdlib.h` para gerar números aleatórios.
  - utilize o operador resto (`%`) para restringir a faixa de valores.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \quad (1) \qquad \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} a_{11} & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \quad (3)$$

Exemplo de execução:

```
Programa gerador de matriz triangular!
Opções:
```

```
 1) Triangular Superior
 2) Triangular Inferior
 3) Diagonal
 4) Sair do Programa
Qual a sua escolha: 1

Matriz gerada:
 383  386  277  415  293  335  386  492  149  421
 000  362  027  190  059  263  426  040  426  172
 000  000  236  211  368  067  429  282  030  362
 000  000  000  123  067  135  429  302  022  058
 000  000  000  000  069  167  393  456  011  042
 000  000  000  000  000  229  373  421  419  284
 000  000  000  000  000  000  037  198  324  315
 000  000  000  000  000  000  000  370  413  026
 000  000  000  000  000  000  000  000  091  480
 000  000  000  000  000  000  000  000  000  456

Opções:
 1) Triangular Superior
 2) Triangular Inferior
 3) Diagonal
 4) Sair do Programa
Qual a sua escolha: 2

Matriz gerada:
 373  000  000  000  000  000  000  000  000  000
 362  170  000  000  000  000  000  000  000  000
 496  281  305  000  000  000  000  000  000  000
 425  084  327  336  000  000  000  000  000  000
 005  346  229  313  357  000  000  000  000  000
 124  395  082  045  314  367  000  000  000  000
 434  364  043  250  087  308  276  000  000  000
 178  288  084  403  151  254  399  432  000  000
 060  176  368  239  012  226  086  094  039  000
 295  070  434  378  467  101  097  402  317  492

Opções:
 1) Triangular Superior
 2) Triangular Inferior
 3) Diagonal
 4) Sair do Programa
Qual a sua escolha: 3

Matriz gerada:
 152  000  000  000  000  000  000  000  000  000
```

```
 000  256  000  000  000  000  000  000  000  000
 000  000  301  000  000  000  000  000  000  000
 000  000  000  280  000  000  000  000  000  000
 000  000  000  000  286  000  000  000  000  000
 000  000  000  000  000  441  000  000  000  000
 000  000  000  000  000  000  365  000  000  000
 000  000  000  000  000  000  000  189  000  000
 000  000  000  000  000  000  000  000  444  000
 000  000  000  000  000  000  000  000  000  119

Opções:
 1) Triangular Superior
 2) Triangular Inferior
 3) Diagonal
 4) Sair do Programa
Qual a sua escolha: 8

Opções:
 1) Triangular Superior
 2) Triangular Inferior
 3) Diagonal
 4) Sair do Programa
Qual a sua escolha: 4
Obrigado por utilizar o programa gerador de matriz triangular!
```

Problema 2

O número de inscrição no CPF (Cadastro de Pessoas Físicas) é formado por onze dígitos (000.000.000-00), dos quais os dois últimos (`j` e `k`) são verificadores (controle), ou seja, a partir dos nove primeiros dígitos pode-se determinar os últimos dois. Além dos dígitos `j` e `k`, também é conhecida a função do dígito `i`, o qual indica o estado onde o cidadão fez o registro no CPF, conforme tabela abaixo.

Dígitos do CPF

0	0	0	.	0	0	0	.	0	0	0	-	0	0
a	b	c		d	e	f		g	h	i		j	k

O algoritmo utilizado para cálculo dos dígitos verificadores é conhecido. Dessa forma, assumindo que cada letra no formato `abc.def.ghi-jk` representa um dígito, pode-se:

- calcular o primeiro dígito verificador (`j`), da seguinte forma:
  - somar: 10 * `a` + 9 * `b` + 8 * `c` + 7 * `d` + 6 * `e` + 5 * `f` + 4 * `g` + 3 * `h` + 2 * `i`

Estado de Registro no CPF

Estado(s)	Dígito `i`
DF, GO, MT, MS e TO	1
AC, AP, AM, PA, RO e RR	2
CE, MA e PI	3
AL, PB, PE e RN	4
BA e SE	5
MG	6
ES e RJ	7
SP	8
PR e SC	9
RS	0

  - encontrar o resto da divisão inteira dessa soma por 11
  - se o resto for igual a 0 ou 1:
    - então, o 1º dígito é 0
    - senão o 1º dígito verificador é 11 menos o resto.

- calcular o segundo dígito verificador (`k`):
  - somar: 11 * `a` + 10 * `b` + 9 * `c` + 8 * `d` + 7 * `e` + 6 * `f` + 5 * `g` + 4 * `h` + 3 * `i` + 2 * `1º dígito verificador`
  - encontrar o resto da divisão dessa soma por 11.
  - se o resto for igual a 0 ou 1:
    - então, o 2º dígito verificador é 0
    - senão o 2º dígito verificador é 11 menos o resto

Faça um Programa em C, que leia do teclado um número de CPF completo (11 dígitos sem `.` e `-`) e então, informe o estado de registro e verifique se este é um número de CPF válido. Para isso, calcule os dois dígitos verificadores e compare com o dígitos `j` e `k` fornecidos pelo usuário. Imprima uma mensagem informando se o número é válido ou não.

- Atenção:
  - `'5'` é diferente de 5, `'5'` representa o caractere enquanto 5 representa a quantidade inteira 5 (ver tabela ASCII). Veja o exemplo a seguir para converter um caractere ASCII em seu valor inteiro equivalente:

```c
char x[] = {"2019"};
int a = x[0] - '0';

printf("a vale: %d", a);
```

Exemplos de execução:

```
Digite o CPF: 12312312300
O CPF informado (12312312300) foi registrado em CE, MA ou PI e
é: inválido!

Digite o CPF: 01234567890
O CPF informado (01234567890) foi registrado em SP e é: válido!

Digite o CPF: 09876543210
O CPF informado (09876543210) foi registrado em AC, AP, AM, PA,
RO ou RR e é: inválido!
```