



# Universidade Estadual do Norte do Paraná

## Bacharelado em Ciência da Computação

### Centro de Ciências Tecnológicas

Tempo de duração: 4hr00

Avaliação I

Programação em C

**Disciplina: Programação I**

**Prof. Maurício M. Arimoto**

**Aluno:** Gabriel Lima de Moraes

**Matrícula:** 202011113030011

Data: 05.12.2020

#### Instruções:

1. A avaliação deve ser entregue em um único arquivo compactado com os códigos-fonte das soluções.
2. A avaliação entregue após o período estabelecido para a sua realização será rejeitada.
3. Cada questão tem um valor/peso; selecione as questões que quiser resolver, porém a soma total dos pesos deve ser obrigatoriamente igual a 10 pontos.
4. Código-fonte executando corretamente recebe nota integral; não executando, mas com código próximo da solução, recebe até 50% do valor da questão.
5. Código-fonte com solução idêntica ao do colega será automaticamente rejeitada, e ambos recebem nota zero (0).
6. A interpretação dos enunciados dos exercícios faz parte da avaliação.

1. [1,5] Suponha que você deseja implementar um programa de controle de frequência diária para a turma de ProgI. O programa deve receber uma lista de frequência para cada aluno (P – Presença e F – Falta) em 30 dias de aula e armazenar em uma matriz de caracteres chamada listaFrequencia. Considere uma turma composta por 27 alunos, sendo que cada dia de aula equivale a duas frequências. Em seguida, o programa deve calcular e mostrar:
  1. Total de faltas de cada aluno individualmente, e se o mesmo foi reprovado por falta. Considere que o aluno deve ter pelo menos 75% de presença na disciplina;
  2. Total de alunos reprovados por faltas na disciplina;
  3. Dia de aula com o maior número de faltas;
  4. Dia de aula com o maior número de presenças.

2. [2,5] Faça um programa para controlar as provas feitas pelos alunos nas disciplinas do curso de Ciência da Computação. O programa deve receber a idade de oito alunos e armazenar em um vetor; em outro vetor deve armazenar o código de cinco disciplinas; e em uma matriz armazenar a quantidade de provas que cada aluno fez em cada disciplina. As seguintes funções devem ser implementadas:
  - (a) Função para calcular e mostrar a quantidade de alunos com idade entre 18 e 25 anos que fizeram mais de duas provas em uma disciplina com código digitado pelo usuário. O usuário pode digitar um código que não está cadastrado; nesse caso, o programa deve mostrar mensagem informando sobre isso;
  - (b) Função para mostrar uma listagem com os números de alunos e os respectivos códigos das disciplinas em que fizeram menos de três provas. Analisar cada disciplina;
  - (c) Função para calcular e mostrar a média de idade dos alunos que não fizeram prova em alguma disciplina. Cuidado para não contar duas vezes o mesmo aluno.
3. [1,0] Suponha que você necessita fazer uma comparação de preços entre produtos de informática de diferentes lojas da sua região. Faça um programa que receba os preços de 20 produtos de informática de 5 lojas diferentes e os armazene em uma matriz 5 x 20. Após a matriz ser preenchida, mostre o número da loja e o número do produto com o menor preço.
4. [2,5] Faça um programa que efetue reservas de passagens aéreas de uma companhia. O programa deverá ler e informar sobre voos (número, código origem e código destino) juntamente com o número de lugares disponíveis para 12 aviões. Deve-se utilizar um vetor para cada um desses dados. O programa deverá fornecer as seguintes opções:
  1. Consultar
  2. Efetuar Reserva
  3. Sair

Quando for escolhida a opção 1, deverá ser disponibilizado mais um menu com as seguintes opções:

1. Por número de voo
2. Por código origem
3. Por código destino

Independentemente da opção escolhida, todos os dados do voo devem ser exibidos: número, código origem e código destino.

Quando for escolhida a opção 2 deverá ser perguntado o número do voo no qual a pessoa deseja viajar. Nesse caso, o programa pode emitir uma das seguintes mensagens:

- Voo inexistente – caso o código do voo não exista.
- Reserva confirmada – caso exista o voo e lugar disponível, dando baixa nos lugares disponíveis.
- Voo lotado – caso o voo exista, mas não há lugares disponíveis.

5. [2,0] Faça um programa para controlar um orçamento doméstico. O programa deve apresentar opções para:

1. Atualizar Orçamento
2. Visualizar Orçamento Atual
3. Encerrar Execução

Ao entrar em atualizar orçamento (opção 1), o programa deve mostrar as categorias de orçamento, numeradas de 1 a 5: 1 – comida, 2 – aluguel, 3 – roupa, 4 – luz/água, 5 – divertimento. O usuário deve fazer referência à categoria pelo número e não pelo nome. O usuário deve indicar qualquer uma das categorias pelo número e então digitar um valor. O programa registra a quantidade gasta e mantém o total de gastos para cada categoria.

Quando o usuário escolher a opção 2, o programa deve mostrar o total gasto em cada categoria e também o total (total de todas as categorias). Além disso, deve ser impresso o código da categoria que resultou no maior gasto no orçamento.

O usuário pode continuar entrando valores após ter visto a totalização e mais tarde pedir para ver o total novamente. Ao escolher a opção 3 a execução do programa é encerrada.

6. [2,0] Faça um programa para cadastrar restaurante (máximo de 50). Cada restaurante deve ter os seguintes dados armazenados: código, preço médio e código do tipo de comida. Deve-se utilizar um vetor para cada um desses dados. O programa deve implementar as seguintes funcionalidades:

1. Cadastrar Restaurante (não podem ser inseridos restaurantes com o mesmo código).
2. Listar todos os restaurantes (código, preço médio e código do tipo de comida) que tenham o preço médio entre dois valores informados pelo usuário.
3. Listar todos os restaurantes (código, preço médio e código do tipo de comida) que sirvam um determinado tipo de comida cujo o código é informado pelo usuário.
4. Sair.

**OBS:** O usuário deverá decidir quantos restaurantes quer cadastrar sem ser necessário cadastrar todos em seguida. Também não deve ser permitido listar restaurante sem ter nenhum restaurante cadastrado (deverá ser enviada ao usuário uma mensagem de erro caso isso aconteça).

7. [1,0] Implemente um programa que leia uma string de tamanho qualquer e imprima:
- (a) O total de caracteres da string;
  - (b) A quantidade de vogais na string; e
  - (c) A porcentagem das vogais em relação ao total de caracteres da string.

8. [1,0] Implemente uma função, chamada *contaNumero*, que receba como parâmetro uma matriz quadrada de números inteiros *matriz*, sua ordem *N*, um valor *X* e retorne quantas vezes o elemento *X* está presente na matriz. Caso o elemento não apareça nenhuma vez na matriz, a função deve retornar o valor  $-1$ . Em seguida, implemente o programa principal que:

[1]

- Leia um número inteiro *K* que corresponde à ordem de uma matriz ( $k \leq 20$ );
- Leia a matriz quadrada *mat* de ordem *K*;
- Leia um número interior qualquer *I*; e
- Chame a função *contaNumero* e imprima uma mensagem indicando o número de vezes que o número *I* está presente na matriz *mat*, ou a mensagem "Número Inexistente" caso *I* não seja encontrado na matriz *mat*.

9. [2,0] Faça um programa que leia dois vetores X e Y de números inteiros com N elementos cada ( $N \leq 20$ ). Em seguida, o programa deve gerar e imprimir 5 novos vetores (C, D, E, F e G) conforme mostrado abaixo.

O vetor C deve corresponder à união dos vetores X e Y, ou seja, todos os elementos de X e os elementos de Y que não estejam em X.

$$\begin{array}{l}
 X = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 3 & 5 & 4 & 2 & 1 & 6 & 8 & 7 & 11 & 9 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \qquad Y = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 2 & 1 & 5 & 12 & 3 & 0 & -1 & 4 & 7 & 6 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \\
 C = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 3 & 5 & 4 & 2 & 1 & 6 & 8 & 7 & 11 & 9 & 12 & 0 & -1 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ \hline \end{array}
 \end{array}$$

O vetor D deve corresponder à diferença entre os vetores X e Y, ou seja, todos os elementos de X que não existam em Y.

$$\begin{array}{l}
 X = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 3 & 5 & 4 & 2 & 1 & 6 & 8 & 7 & 11 & 9 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \qquad Y = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 2 & 1 & 5 & 12 & 3 & 0 & -1 & 4 & 7 & 6 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \\
 D = \begin{array}{|c|c|c|} \hline 8 & 11 & 9 \\ \hline 0 & 1 & 2 \\ \hline \end{array}
 \end{array}$$

O vetor E deve corresponder à soma entre X e Y, ou seja, a soma de cada elemento de X com o elemento da mesma posição em Y.

$$\begin{array}{l}
 X = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 3 & 5 & 4 & 2 & 1 & 6 & 8 & 7 & 11 & 9 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \qquad Y = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 2 & 1 & 5 & 12 & 3 & 0 & -1 & 4 & 7 & 6 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \\
 E = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 5 & 6 & 9 & 14 & 4 & 6 & 7 & 11 & 18 & 15 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array}
 \end{array}$$

O vetor F deve corresponder ao produto entre X e Y, ou seja, a multiplicação de cada elemento de X com o elemento da mesma posição em Y.

$$\begin{array}{l}
 X = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 3 & 5 & 4 & 2 & 1 & 6 & 8 & 7 & 11 & 9 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \qquad Y = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 2 & 1 & 5 & 12 & 3 & 0 & -1 & 4 & 7 & 6 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \\
 F = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 6 & 5 & 20 & 24 & 3 & 0 & -8 & 28 & 77 & 54 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array}
 \end{array}$$

O vetor G deve corresponder à intersecção entre X e Y, ou seja, apenas os elementos que aparecem nos dois vetores.

$$\begin{array}{l}
 X = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 3 & 5 & 4 & 2 & 1 & 6 & 8 & 7 & 11 & 9 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \qquad Y = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 2 & 1 & 5 & 12 & 3 & 0 & -1 & 4 & 7 & 6 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \\
 G = \begin{array}{|c|c|c|c|c|c|} \hline 3 & 5 & 4 & 2 & 1 & 6 & 7 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline \end{array}
 \end{array}$$

10. [2,5] Para estudar o transporte na região metropolitana de Marília - SP, a Empresa de Desenvolvimento Urbano e Habitacional de Marília - EMDURB cobriu a cidade com uma rede de quadrículas, conforme a Figura 1. Para cada uma das 120 quadrículas foi usada uma identificação e foram levantados dois números denominados **Índice de Transporte Urbano – ITU** e **Índice de Densidade Populacional – IDP**.

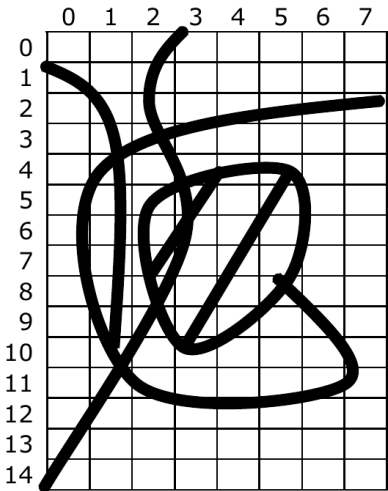


Figura 1: Cidade coberta por uma rede de quadrículas

Faça um programa capaz de fornecer aos técnicos da EMDURB os três mapas seguintes: [3]

- 1. Mapa *ITU*;
- 2. Mapa *IDP*; e
- 3. Mapa com números 1, 2 ou 0, dependendo dos valores de *ITU* e *IDP* de cada quadrícula.

Os mapas 1 e 2 estão esboçados na Tabela 1, conforme apresentada a seguir.

Mapa ITU					Mapa IDP				
32,	20,	10,	...	90	10,	0,	5,	...	30
35,	0,	51,	...	20	32,	58,	91,	...	80
...	...	...	...	...	...	...	...	...	...
25,	5,	13,	...	20	43,	50,	61,	...	0
8 colunas					8 colunas				

15 linhas

Tabela 1: Mapa *ITU* e Mapa *IDP*

O mapa 3 deve conter:

- (a) Número 1 nas quadrículas em que *ITU* é menor que 20 e *IDP* é maior que 55;
- (b) Número 2 nas quadrículas em que *ITU* é maior que 0 e *IDP* é igual a 0; e
- (c) Número 0 nas quadrículas que não satisfazem nem o item *a* nem o item *b*.

O exemplo correspondente do mapa 3 é apresentado na Tabela 2.

0,	2,	0,	...,	0
0,	1,	0,	...,	0
...,	...,	...,	...,	...
0,	0,	1,	...,	2

Tabela 2: Mapa Resultante

11. [1,5] Faça um programa que leia um par de números ( $R$ ,  $S$ ) e imprima esse par se ele satisfaz a uma das seguintes condições:
1. A metade do *maior* valor dos dois números do par é igual ao menor  $\pm 0.1$ ;
  2. O maior entre os números  $a$  e  $b$ , sendo  $a$  obtido pela soma dos dois números, e  $b$  obtido pela subtração do segundo pelo primeiro, seja igual a 10.
- OBS:** Deve-se implementar uma função chamada *imprimePar* que receba como parâmetro os números  $R$  e  $S$  e mostre os respectivos números obedecendo às condições acima.
12. [1,5] Suponha que você ficou incumbido de desenvolver um programa para corrigir provas de múltipla escolha da disciplina de Engenharia de Software. Considere que cada prova tem 20 questões e cada questão vale 0,5 ponto. O programa deve receber o gabarito da prova, o número de matrícula de 30 alunos e suas respectivas respostas. Em seguida, o programa deve calcular e mostrar:
1. Número de matrícula e nota de cada aluno;
  2. Nota média da turma;
  3. Percentual de aprovação, sabendo-se que a nota mínima é 7.0.