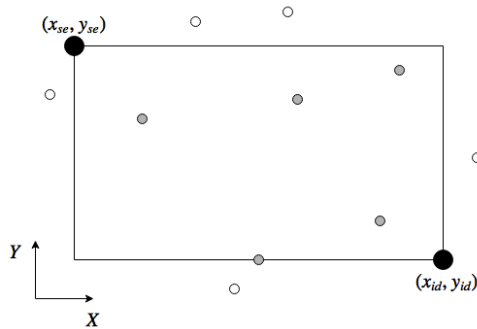




Exercícios – Tipos Estruturados e Ordenação

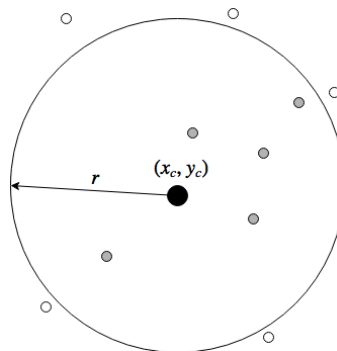
1. Defina um tipo estruturado **Aluno**, contendo os campos **nome**, **matricula**, e **nota**. Em seguida:
 - (a) Implemente uma função que receba como parâmetro de entrada um vetor com **n** alunos. A função deve computar a menor e maior notas dentre todos os alunos, armazenando o resultado em dois parâmetros de saída. A função **main** deve ler do usuário o número **n** de alunos, os dados de cada um deles e utilizar o resultado da função para exibir os dados de todos os alunos que obtiveram a menor nota seguidos pelos dados de todos os alunos que obtiveram a maior nota (considerando que mais de um aluno obteve a menor/maior nota).
 - (b) Implemente uma função que receba como parâmetro de entrada um vetor com **n** alunos. A função deve retornar a posição no vetor do aluno com o maior nome. A função **main** deve ler do usuário o número **n** de alunos, os dados de cada um deles e utilizar o resultado da função para exibir os dados do aluno com maior nome na tela.
 - (c) Implemente uma função que receba como parâmetros de entrada um vetor com **n** alunos e uma string contendo um nome a ser buscado. A função deve retornar a posição no vetor do aluno cujo nome seja igual à string passada como parâmetro ou -1 caso um aluno com o nome passado como parâmetro não exista. Considere que todos os nomes serão compostos por letras minúsculas e apenas um espaço separando cada palavra. A função **main** deve ler do usuário o número **n** de alunos, os dados de cada um deles e um nome para busca. Por fim, ela deve exibir na tela todos os dados do aluno a ser buscado ou uma mensagem informando caso ele não tenha sido cadastrado de acordo com o resultado da chamada à função implementada.
2. Implemente uma função que receba como parâmetro de entrada duas variáveis do tipo estruturado **Data**, o qual deve ser composto pelos campos **dia**, **mes** e **ano**. A função a ser implementada deve retornar 1 caso a primeira data seja a mais antiga, 0 se as datas forem iguais e -1 caso a segunda data seja a mais antiga. A função **main** deve ler o dia, mês e ano de ambas as datas e exibir na tela uma mensagem informando qual das duas datas é mais antiga, de acordo com o valor retornado pela função.
3. Defina um tipo estruturado **Ponto2D** para representar pontos em um plano. Em seguida:
 - (a) Implemente uma função que receba como parâmetro de entrada um vetor com **n** pontos 2D. A função a ser implementada deve retornar o ponto 2D mais próximo da origem (ponto com coordenadas iguais a zero). A função **main** deve ler o número **n** de pontos, as coordenadas de cada ponto e exibir na tela as coordenadas do ponto mais próximo da origem de acordo com o resultado da função.
 - (b) Implemente uma função que receba como parâmetro de entrada um vetor com **n** pontos 2D. A função a ser implementada deve retornar o centróide dos pontos, isto é, um ponto que possui as coordenadas iguais à média de todos os pontos. A função **main** deve ler o número **n** de pontos, as coordenadas de cada ponto e exibir na tela as coordenadas do centróide de acordo com o resultado da função.

4. Um retângulo pode ser representado computacionalmente por um tipo estruturado contendo como campos as coordenadas (x_{se}, y_{se}) do seu vértice superior esquerdo e as coordenadas (x_{id}, y_{id}) do seu vértice inferior direito. Utilizando esta representação, implemente uma função que, recebendo como parâmetros um vetor com n pontos 2D e um retângulo, retorne a quantidade de pontos que estão dentro do retângulo. Considere a ilustração a seguir para desenvolver o seu raciocínio.



A função `main` deve ler a quantidade n de pontos, cada uma das n coordenadas dos pontos e as coordenadas dos vértices superior esquerdo e inferior direito do retângulo. Por fim, ela deve exibir na tela a quantidade de pontos dentro do retângulo utilizando uma chamada à função implementada.

5. Um círculo pode ser representado computacionalmente por um tipo estruturado contendo como campos as coordenadas do seu centro (x_c, y_c) e o seu raio r . Utilizando esta representação, implemente uma função que, recebendo como parâmetros de entrada um vetor com n pontos 2D e um círculo, retorne a quantidade de pontos que estão dentro do círculo. Considere a ilustração a seguir para desenvolver o seu raciocínio.



A função `main` deve ler a quantidade n de pontos, cada uma das n coordenadas dos pontos, as coordenadas do centro do círculo e o seu raio. Por fim, ela deve exibir na tela a quantidade de pontos que estão dentro do círculo utilizando uma chamada à função implementada.

6. Defina um tipo estruturado `Matriz` para representar uma matriz de tamanho qualquer de números inteiros. O tipo estruturado deve conter como campos o número de linhas, o número de colunas e os dados da matriz. Em seguida, implemente uma função que receba como parâmetros de entrada duas variáveis do tipo `Matriz` e que armazene em um parâmetro de saída, também do tipo `Matriz`, o resultado da multiplicação entre as duas primeiras matrizes. A função deve verificar se a multiplicação é possível e caso não seja, deve emitir uma mensagem informando e encerrar o programa. A função `main` deve ler as dimensões e os dados de duas matrizes e exibir na tela o resultado da multiplicação entre as duas matrizes, de acordo com o resultado da função.

7. Defina um tipo estruturado `Funcionario`, contendo os campos `nome`, `matricula`, e `salario`. Em seguida:
- (a) Implemente uma função que receba como parâmetro de entrada e saída um vetor com `n` funcionários. A função deve realizar a ordenação do vetor de funcionários em ordem decrescente de salário. A função `main` deve ler o número `n` de funcionários, os dados de cada um deles e exibir o vetor ordenado conforme o resultado da função.
 - (b) Implemente uma função que receba como parâmetro de entrada e saída um vetor com `n` funcionários. A função deve realizar a ordenação do vetor de funcionários em ordem alfabética do nome, utilizando a função `strcmp`. A função `main` deve ler o número `n` de funcionários, os dados de cada um deles e exibir o vetor ordenado conforme o resultado da função.