

PONTEIROS:

Questão 4) Este código tem como objetivo calcular a área e o perímetro de um hexágono regular, a partir do comprimento do lado inserido pelo usuário.

ALOCAÇÃO DINÂMICA DE VETORES:

Questão 1) Esse código faz uma pesquisa sobre a opinião de pessoas sobre um produto. Ele lê o número de pessoas entrevistadas e armazena seus sexos e opiniões em arrays dinâmicos, e então conta quantas mulheres gostaram do produto, quantas mulheres não gostaram e quantos homens não gostaram. Em seguida, calcula a porcentagem de mulheres que gostaram do produto e a porcentagem de homens que não gostaram, e imprime essas informações na tela. Por fim, libera a memória alocada para os arrays dinâmicos.

Questão 2) Esse programa tem como objetivo calcular a nota e aprovação de uma turma de 10 alunos em uma prova com um gabarito fornecido. Ele usa uma matriz de caracteres para armazenar as respostas de cada aluno e o gabarito em uma string simples. O programa solicita o número de questões na prova e o gabarito ao usuário. Em seguida, ele usa um loop for para ler as respostas de cada aluno, calcular sua nota e verificar se ele foi aprovado ou reprovado. Por fim, ele calcula a porcentagem de alunos aprovados e a imprime. O programa também gerencia a alocação de memória dinâmica para as matrizes de respostas dos alunos.

MATRIZ DINÂMICA:

Questão 1) Esse código lê um mapa representado por um grid de caracteres, onde "." representa água e "#" representa terra. Em seguida, ele conta o número de "costas", ou seja, o número de "#" que estão adjacentes a um ".", que é a água. O número de linhas e colunas do mapa é especificado pelo usuário, e a alocação de memória é feita dinamicamente usando malloc. Por fim, o mapa é impresso junto com o número de costas encontrado.

ESTRUTURA DE DADOS:

Questão 1) Este código implementa um programa em C que cadastra funcionários, armazenando seus nomes, salários, cargos e identificadores. O programa permite ao usuário imprimir os funcionários cadastrados, buscar o funcionário com o maior e menor salário e alterar o salário de um funcionário específico.

Questão 2) Segue a mesma lógica do programa da questão 1, a única diferença são as variáveis.

TIPOS ESTRUTURADOS:

Questão 1) Este código é uma implementação de um programa para gerenciamento de matrículas de alunos em turmas, incluindo funcionalidades como criar uma nova turma, matricular um aluno, registrar notas e imprimir informações.

O código usa uma estrutura para cada aluno, que inclui seu ID, nome, nota (matriz de 3) e nota média. Outra struct representa cada classe, que possui um ID, o número de vagas disponíveis (inicialmente definido como no máximo 10) e uma matriz de ponteiros para os alunos.

A função principal possui um loop while que apresenta um menu de opções e chama a função apropriada com base na entrada do usuário. As opções disponíveis são:

- 1.criar classe
- 2.Listar classes
- 3.matricular aluno
- 4.Registrar notas
- 5.Listar alunos de uma turma
- 6.Sair do programa

O código também possui algumas funções auxiliares para criar uma nova turma, localizar uma turma específica, matricular um aluno em uma turma, registrar suas notas e imprimir uma lista de alunos de uma determinada turma.

ENUM E UNION:

Questão 1) Este código implementa um programa que cria um struct pessoa que é ligada a uma enum gênero que pode ser masculino ou feminino, então o usuário define quantas pessoas deseja cadastrar, ao terminar de cadastrar o programa imprime todas as informações de cada pessoa, dizendo se o gênero é masculino ou feminino.

Questão 2) Segue a mesma lógica da questão anterior, porém com produtos e com tipos de produtos.

Questão 3) Este código implementa uma enum com todos os meses do ano, sendo que janeiro é igual a 1 para seguir a ordem correta, então dentro da main o usuário digita o dia o mês e o ano, ao fim o programa mostra a data no formato dd/mm/aaaa.

OBSERVAÇÃO: Caso as explicações não sejam suficientes, todos os códigos estão comentados.