

Objetivo Geral:

O código cria um sistema simples para determinar a compatibilidade entre diferentes pessoas com base nas respostas a um questionário. Ele permite ao usuário definir o número de perguntas e o número de pessoas. Depois, ele gera respostas aleatórias para cada pessoa, e finalmente compara as pessoas para encontrar aquelas que são compatíveis com outra pessoa específica, usando um limiar de distância entre as respostas.

Estrutura e Componentes:

1. Vetor de Nomes Estático:

- um vetor pré-definido de nomes (vNomAux[]) com 20 opções de nomes.
 - Esse vetor é usado para atribuir nomes aleatórios às pessoas criadas no programa.
- ```
const char *vNomAux[NUM_NOMES] = {"Bola", "Gigi", "Ana", ...};
```

### 2. Tipos Definidos (typedef):

- **Questionario:** Um tipo que representa um questionário. Embora o tipo em si não seja diretamente utilizado no código final (foi simplificado), a ideia original era representar um questionário com um número de perguntas determinado por uma constante no programa.
- **respQuestionario:** Um vetor de respostas associadas ao questionário. Cada resposta é representada por um número inteiro. A quantidade de respostas é determinada por uma constante definida no programa (n)
- **Pessoa:** Uma estrutura que representa uma pessoa, contendo um nome e o vetor de respostas (respQuestionario) ao questionário.
- O nome deve dinamicamente alocado

### 3. Funções Importantes:

- **calcularDistancia():**
  - Calcula a distância entre duas pessoas comparando as respostas delas.
  - A distância é calculada usando a **distância euclidiana**, que é uma forma comum de medir a "distância" entre dois pontos em um espaço.

Da mesma forma, a **distância euclidiana** entre os pontos  $\mathbf{u} = (u_1, u_2, \dots, u_n)$  e  $\mathbf{v} = (v_1, v_2, \dots, v_n)$  do  $R^n$  é definida por

$$d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \dots + (u_n - v_n)^2} \quad (2)$$

- **encontrarCompatibilidade():**
  - Recebe um nome de pessoa, o vetor de pessoas e um limiar de distância.

- ☐ Primeiro, encontra a pessoa pelo nome no vetor de pessoas. Depois, compara essa pessoa com todas as outras para ver quem tem uma distância menor que o limiar.
- ☐ Exibe os nomes das pessoas compatíveis com base na distância calculada.
- ☐ `inicializarPessoas()`:
  - ☐ Inicializa o vetor de pessoas. Atribui a cada pessoa um nome aleatório (usando `vNomAux[]`) e gera respostas aleatórias para o questionário.
  - ☐ As respostas são números inteiros aleatórios entre 0 e 9.

#### 4. Fluxo do Programa (Função `main()`):

- ☐ O programa começa perguntando ao usuário quantas pessoas ele quer criar
- ☐ Em seguida, o vetor de pessoas é inicializado com nomes e respostas gerados aleatoriamente.
- ☐ Depois, o programa pede ao usuário o nome de uma pessoa para comparar e o limiar de compatibilidade.
- ☐ Com essas informações, o programa exibe as pessoas que são compatíveis com a pessoa escolhida, com base na distância entre as respostas delas.
- ☐ Finalmente, o programa libera a memória alocada dinamicamente.

### Exemplo de Execução:

Suponha que o usuário insira:

- ☐ Número de pessoas: 5
- ☐ Nome da pessoa: "Ana"
- ☐ Limiar: 5.0

O programa então exibe as pessoas cujas respostas ao questionário são próximas das respostas da "Ana", com base no limiar fornecido.