

C de Cornos - Projeto 1

Você será responsável pela implementação de um sistema de cadastro de usuários. Esse sistema deve conter algumas operações essenciais, como incluir, remover e listar os usuários cadastrados.

Especificações Funcionais

O sistema deve possuir 4 funcionalidades: inclusão de usuário, remoção de usuário, busca de usuário por nome e listagem de todos os usuários. O sistema comporta o cadastro de no máximo 100 usuários.

Lógica principal

O sistema deve possuir um laço de execução principal, onde será solicitado o input do operador do sistema. Deve ser imprimido um menu descrevendo os comandos disponíveis, que deverão ser:

1. Inclusão de novo usuário
2. Buscar um usuário
3. Remover cadastro de usuário
4. Listar todos os usuários cadastrados

Faz parte do menu a opção de sair do programa, que deve ser feita com o input de 0. Outros inputs serão considerados inválidos, e deverá ser imprimida a mensagem `Opcao invalida!`

Informações do usuário

O cadastro dos usuários deve conter as seguintes informações:

```
char *nome
char *email
unsigned short idade
```

Essa definição deve ser feita de maneira coesa, através de uma `struct` de nome `Usuario`. Considere que todos os campos de texto têm tamanho máximo de 100.

Inclusão de usuário

Quando selecionada a opção 1, o operador deve fornecer as informações para o cadastro. Ao iniciar o cadastro, imprimir a mensagem `Cadastrando novo usuario`. Ao solicitar o input, imprimir o nome do campo solicitado.

Primeiro campo - Nome: Segundo campo - Email: Terceiro campo - Data de nascimento:

A data de nascimento sempre será fornecida no formato ISO (YYYY-MM-DD). A partir dessa data, calcular a idade do usuário e salvar esta informação.

No caso de sucesso, imprimir a mensagem `%s cadastrado com sucesso!`, com duas quebras de linha, onde `%s` é o nome do usuário.

Busca de usuário

Quando selecionada a opção 2, imprimir a mensagem `Buscando usuario`. Em seguida, a mensagem `Digite o nome:` deve ser imprimida, onde o operador deverá fornecer o nome do usuário que será buscado.

Se o usuário não existir no cadastro, imprimir `%s nao encontrado no cadastro!` com duas quebras de linha, onde `%s` é o nome de busca.

Quando encontrado, imprimir os dados do usuário conforme a regra de listagem.

Remoção de usuário

Quando selecionada a opção 3, imprimir a mensagem `Remoção de usuário`. Em seguida, a mensagem `Digite o nome:` deve ser imprimida, onde o operador deverá fornecer o nome do usuário que será removido.

Se o usuário não existir no cadastro, imprimir `%s nao encontrado no cadastro!` com duas quebras de linha, onde `%s` é o nome de busca.

Quando encontrado, remover o cadastro do usuário da lista de usuários. Isso significa que é necessário mover todos os membros da lista após o usuário removido para "preencher" o espaço vago na lista.

Listagem de usuários

Quando selecionada a opção 4, a informação de todos os usuários deve ser imprimida na tela. A impressão de cada usuário deve seguir este formato de exemplo:

```
Nome: Joao Silva
Email: joaoss@gmail.com
Idade: 25 anos
```

Incluir duas quebras de linha após a idade para que haja distinção visual entre os diferentes usuários.

Recomendações Técnicas

Lógica principal

Na `main()`, realize o loop de input. Use uma função auxiliar para imprimir o menu principal. Se será imprimido uma vez ou toda vez que passar pelo loop, fica ao critério de desenvolvimento.

Exemplo de função para imprimir o menu:

```

void print_menu()
{
    printf("=====\n");
    printf("=====\n");
    printf("===== MENU PRINCIPAL - CADASTRO DE USUARIOS =====\n");
    printf("=====\n");
    printf("=====\n");
    printf("==== 1 - Cadastrar novo usuario          ====\n");
    printf("==== 2 - Buscar por um usuario          ====\n");
    printf("==== 3 - Remover cadastro de usuario      ====\n");
    printf("==== 4 - Listar usuarios cadastrados      ====\n");
    printf("==== 0 - Sair                             ====\n");
    printf("=====\n");
    printf("=====\n");
}

```

Ainda na `main()`, você deve declarar a lista de usuários. Para manter o controle do tamanho atual da lista, vale declarar uma variável (ex: `int tamanho`) e passar ambas as variáveis para as funções mais específicas.

O tamanho máximo da lista é de 100, o mesmo tamanho de todos os outros inputs variáveis, como as strings, então é possível utilizar um `#define MAX_SIZE 100`.

Ao realizar o cadastro e remoção de usuários, será necessário ajustar o tamanho da lista. Lembre-se de atualizar a variável `int tamanho` de acordo. No entanto, caso haja algum erro no cadastro ou remoção, o tamanho não deve ser afetado. Isso pode ser feito de acordo com o retorno das funções.

Inclusão de usuário

Para realizar a inclusão de um usuário, você precisará ler 3 inputs do operador. Tome cuidado com inputs onde houver espaço -- por exemplo, ao digitar o nome `Joao Silva`, a leitura desse valor através de um simples `scanf("%s", nome)`; não é suficiente, pois o `scanf` ignora espaços, truncando parte do input.

Você não receberá a idade do usuário diretamente, mas sim a sua data de nascimento. Crie uma função para extrair a idade. Para simplificar, você pode considerar apenas o ano da data de nascimento e fazer a diferença com 2024.

Considere usar a função `atoi()`.

Para controlar o tamanho da lista, essa função pode retornar um código específico para indicar que o cadastro foi feito corretamente. Por exemplo, `return 1;`. Na inclusão, não será necessário tratar nenhum cenário de falha.

Buscar por um usuário

Para buscar um usuário, é necessário ter um critério de busca. Esse critério será o nome digitado pelo operador.

Recomendação de organização: a função de busca pode retornar o índice do usuário específico, e outra função imprimir seus dados.

Caso o usuário não seja encontrado, um aviso deve ser imprimido. Nesse caso, retornar o código -1 é uma alternativa, pois os retornos a partir de 0 são válidos.

Remoção de usuário

Para remover um usuário, é necessário encontra-lo antes com base no nome fornecido pelo operador. É possível receber um pedido de deleção de um usuário que não exista, onde não deve ser feita nenhuma operação.

Com isso, faz sentido que a deleção utilize a função de busca para encontrar o usuário. Quando encontrado, a lista de usuários deve ser deslocada para cobrir aquele espaço.

Lista com 4 usuários: | A | B | C | D | |---|---|---|---|

Remover B: | A | ☐ B | C | D | |---|---|---|---|

Atualizar lista: | A | C | D | | |---|---|---|---|