



Exercício – Controlando um LED

Tarefa

Escreva um programa para controlar o estado de um LED. O programa deve ser capaz de executar as seguintes ações:

- * Configurar o pino em que o LED será conectado e o seu estado inicial;
- * Ligar o LED;
- * Desligar o LED;
- * Inverter o estado do LED;
- * Exibir uma mensagem indicando o estado do LED.

Divida a implementação nos seguintes arquivos:

mainLED.c: Contém o programa principal que será responsável pela execução da operação desejada de acordo com um código digitado pelo usuário.

LED.h: Contém a definição do tipo **LED** e a declaração dos protótipos das funções.

LED.c: Contém a implementação de cada função.

Metodologia

Você deve realizar a implementação em etapas alterando cada arquivo conforme solicitado. Execute o programa ao final de cada etapa para confirmar se o enunciado foi plenamente atendido.

Detalhamento das etapas

Segue a descrição de cada etapa a ser implementada.

Etapa 1

(a) Digite o programa abaixo no arquivo **mainLED.c**. Nele está contida a função **main()** que será responsável pelo gerenciamento das operações. Faça-o funcionar (criando um projeto) testando cada um dos 4 itens do cardápio. Analise o código para compreender o seu funcionamento.

```
#include <stdio.h>

int main() {
    int opcao;

    printf("LED inicializado\n");
    printf("1.Liga LED\n");
    printf("2.Desliga LED\n");
    printf("3.Inverte estado do LED\n");
    printf("4.Fim\n");
    do {
        printf("Escolha: ");
        scanf("%d",&opcao);

        switch (opcao) {
            case 1:
                printf("Item 1\n");
                break;
            case 2:
                printf("Item 2\n");
```

```

        break;
    case 3:
        printf("Item 3\n");
        break;
    }
} while (opcao != 4);
return 0;
}

```

Etapa 2

(a) Crie um arquivo chamado **LED.h** com a definição do tipo **LED** (typedef). Ele deve conter dois campos:

```

int pino; // Pino no qual o LED será conectado
int estado; // Armazena um inteiro indicando se o LED está ligado (1)
// ou desligado (0).

```

(b) Declare uma variável do tipo **LED** na função **main()** do arquivo **mainLED.c**.

(c) Adicione o arquivo **LED.h** no projeto e execute a compilação.

Etapa 3

(a) Crie um arquivo chamado **LED.c** com a implementação da função chamada **inicializaLED**. Ela deve receber como argumentos uma variável do tipo **LED** (passada por referência) e um inteiro indicando o pino em que o LED será conectado. A função deve atribuir ao campo **pino** o valor recebido como parâmetro. O LED deve ser configurado com o estado **desligado**.

(b) Inclua no arquivo **LED.c** a implementação da função chamada **exibeEstadoLED**. Ela deve receber como argumento uma variável do tipo **LED**. A função deve exibir uma das seguintes frases conforme o estado do LED:

```

LED do pino n : LIGADO
ou
LED do pino n : DESLIGADO

```

OBS: No lugar de **n** deve ser impresso o número do pino em que o LED está conectado.

(c) Inclua o protótipo das duas funções no arquivo **LED.h**.

(d) Inclua a chamada às duas funções conforme indicado no trecho de código fonte abaixo:

```

...
// Chamar aqui a função inicializaLED.
// Escolha um valor de pino para passar como argumento
// Chamar aqui a função exibeEstadoLED
printf("LED inicializado\n");
printf("1.Liga LED\n");
printf("2.Desliga LED\n");
printf("3.Inverte estado do LED\n");
printf("4.Fim\n");
do {
    printf("Escolha: ");
    scanf("%d",&opcao);
    ...
}

```

(e) Adicione o arquivo **LED.c** no projeto e execute a compilação.

Etapa 4

(a) Implemente a função **ligaLED**. Ela deve receber como argumento uma variável do tipo **LED** passada por referência. A função deve alterar o estado do LED para **ligado**. A definição da função e o seu respectivo protótipo devem ser inseridos nos arquivos adequados.

(b) Substitua o `printf("Item 1\n");` do `case 1:` (localizado na função `main()`) pelas chamadas das funções `ligaLED` e `exibeEstadoLED` conforme o trecho de código abaixo.

```
...
switch (opcao) {
    case 1:
        // Chamar aqui a função ligaLED
        // Chamar aqui a função exibeEstadoLED
        break;
    ...
}
```

Etapa 5

(a) Implemente a função `desligaLED`. Ela deve receber como argumento uma variável do tipo `LED` passada por referência. A função deve alterar o estado do LED para **desligado**. A definição da função e o seu respectivo protótipo devem ser inseridos nos arquivos adequados.

(b) Substitua o `printf("Item 2\n");` do `case 2:` (localizado na função `main()`) pelas chamadas das funções `desligaLED` e `exibeEstadoLED` conforme o trecho de código abaixo.

```
...
case 2:
    // Chamar aqui a função desligaLED
    // Chamar aqui a função exibeEstadoLED
    break;
...
```

(c) Altere a função `inicializaLED` de forma que a função `desligaLED` seja chamada para estabelecer o estado inicial do LED como **desligado**.

Etapa 6

(a) Implemente a função `inverteEstadoLED`. Ela deve receber como argumento uma variável do tipo `LED` passada por referência. A função deve inverter o estado do LED. A definição da função e o seu respectivo protótipo devem ser inseridos nos arquivos adequados.

(b) Substitua o `printf("Item 3\n");` do `case 3:` (localizado na função `main()`) pelas chamadas das funções `inverteEstadoLED` e `exibeEstadoLED` conforme o trecho de código abaixo.

```
...
case 3:
    // Chamar aqui a função inverteEstadoLED
    // Chamar aqui a função exibeEstadoLED
    break;
...
```