```
/*
Um posto de saúde deseja cadastrar os seguintes dados sobre as pessoas atendidas: nome, idade,
peso e altura. Defina uma estrutura (registro) conveniente para armazenar estes dados.

Considere que o cadastro será armazenado em um vetor. O tamanho do vetor deve ser definido
dinamicamente. Solicite ao usuário o número de pessoas que serão cadastradas.

Crie uma função que leia os dados de uma pessoa e retorne um registro com os dados lidos. Leia os
dados das pessoas e armazene-os em um vetor.

Crie uma função que imprima os dados das pessoas com idade  abaixo da média dos cadastrados.
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Definição da estrutura pessoa

typedef struct pessoa
{
    char nome[50];
    int idade;
    float peso;
    float altura;
} pessoa;

// Função que lê os dados de uma pessoa, armazena em um registro e retorna o registro pessoa.

pessoa nova_pessoa()
{
    pessoa p;
    getchar();
    printf("\nNome: ");
    fgets(p.nome, 50, stdin);
    printf("Idade: ");
    scanf ("%d", &p.idade);
    printf("Peso: ");
    scanf ("%f", &p.peso);
    printf("Altura: ");
    scanf ("%f", &p.altura);
    return p;
}

/*
Função que imprime os dados das pessoas com idade menor que a média do grupo
v: ponteiro para a primeira posição de um vetor de registros pessoa
n: número de posições do vetor (número de pessoas)
media: média das idades das pessoas do grupo
*/
void imprimeAbaixoMedia(pessoa* v, int n, float media)
{
    int i;
    printf("\nPessoas com idade menor que a média do grupo:\n");
    for (i = 0; i < n; i++)
    {
        if (v[i].idade < media)
        {
            printf("\nNome = %s", v[i].nome);
            printf("Idade = %d\n", v[i].idade);
            printf("Peso = %.2f\n", v[i].peso);
            printf("Altura = %.2f\n", v[i].altura);
        }
    }
}

int main(void)
{
```

```
    int nroPessoas, i, j;
    pessoa* vetP;
    float media_idade;
    printf("\nInforme o número de pessoas atendidas: ");
    scanf("%d",&nroPessoas);

    // Alocação de memória para o vetor de registros pessoa
    vetP = (pessoa *) malloc (nroPessoas * sizeof (pessoa));

    // Verifica se a alocação foi realizada com sucesso.
    if (vetP == NULL)
    {
        printf("Erro durante a alocação de memória");
        exit(1);
    }

    for (i = 0; i < nroPessoas; i++)
    {
        vetP[i] = nova_pessoa();
        media_idade += vetP[i].idade;
    }

    media_idade = media_idade / nroPessoas;
    imprimeAbaixoMedia(vetP, nroPessoas, media_idade);

    // Liberação de memória
    free(vetP);
    return 0;
}
```