

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

// Estrutura
typedef struct _pessoa{
    char cod[5];
    int idade;
}Pessoa;

// Protótipos
char * concatena (char * s , char * t );
char * adicionaCodigo(char * codigos, Pessoa p);

int main(void) // Main dada no enunciado.
{
    Pessoa p1 = {"250C",35};
    Pessoa p2 = {"700D",40};
    Pessoa p3 = {"300B",30};
    char * s0 = "10A;20B";
    char * r = (char *)malloc((strlen(s0)+1)*sizeof(char));
    strcpy(r,s0); // inicializa a cadeia
    r = adicionaCodigo(r,p1); // adiciona ou nao o codigo de p1
    printf("Atualizacao: %s\n",r); // imprime -> Atualizacao: 10A;20B;250C
    r = adicionaCodigo(r,p2); // adiciona ou nao o codigo de p2
    printf("Atualizacao: %s\n",r); // imprime -> Atualizacao: 10A;20B;250C
    r = adicionaCodigo(r,p3); // adiciona ou nao o codigo de p3
    printf("Atualizacao: %s\n",r); // imprime -> Atualizacao: 10A;20B;250C;300B
    free(r);
    return 0;
}

char * adicionaCodigo(char * codigos, Pessoa p)
{
    if(p.idade < 40) // se a idade for menor que 40, inserimos ela nos codigos.
    {
        char * codsComPontoVirgula = concatena(codigos,";") ; // concatena ; com os
                                                                // codigos recebidos .

        return concatena(codsComPontoVirgula ,p.cod); // concatena os codigos (com ;)
                                                       // com o código da pessoa e
                                                       // retorna o resultado.
    }

    return codigos; // caso a idade >= 40, então retorna codigos inalterado.
}

char * concatena(char * s , char * t)
{
    int tamS = strlen(s) ;
    int tamT = strlen(t) ;
    int tamNovaStr = tamS + tamT + 1 ; // tamanho da string t + string s + 1 para o \0
    char * novaStr = (char*) malloc(tamNovaStr) ; // não é necessário colocar sizeof(char)
                                                  // pois char tem 1 byte, logo sizeof(char) é 1 .

    int i , j;
    // copia a string s para a nova string.
    strcpy(novaStr,s);

    for(i = tamS, j = 0 ; j <= tamT ; i++ , j++ ) // i recebe o tamS pois ele deve
        novaStr[i] = t[j];                       // começar a colocar os caracteres
                                                  // a partir do \0 da string s (q foi copiado no trecho acima)
                                                  // j é o índice que percorrera a string t.
                                                  // j deve ser <= a tamT a fim de copiar também o \0.

    free(s);
    return novaStr;
}
```