# Instituto Superior de Engenharia do Porto

## Departamento de Engenharia do Porto

### Algoritmia e Programação

#### 3º Exercício Prático

Nome:_____

Número:_____ Turma: _____

Um programador estava a desenvolver uma aplicação para a gestão de clientes de uma discoteca. Acontece que por motivos desconhecidos o programador desapareceu. Portanto alguém tem que terminar a aplicação. O funcionamento da discoteca é o seguinte. A cada cliente é dado, à entrada, um Cartão com um identificador. De cada vez que o cliente consome algo o código do produto e a quantidade é associado e ao seu Cartão. Se o cliente exceder os 10 consumos terá que liquidar a conta e pedir novo Cartão. Em cada consumo só pode consumir um tipo de produto mas pode consumir mais do que uma unidade desse produto. Quando um cliente sai ou sempre que excede os numero de consumos, o programa deverá calcular o preço a pagar e eliminar o seu registo (numero de produtos consumidos a -1). Nunca passam por lá mais de 600 clientes numa noite.
O Programador já tinha definido as estruturas de dados e assim como outras funções. O que se pretende é que desenvolva as que faltam que são as seguintes:

1. Crie uma função para registar a entrada de um cliente.
2. Crie uma função para inserir consumos.
3. Crie uma função para efectuar o pagamento dos consumos.

Considere que existem somente NPROD produtos e que o correspondente vector já está preenchido.

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

#define NCLI        600
#define NPROD       10
#define NCONS       10
#define LSTR        20

typedef struct prod {
        int cod;                        /*codigo do produto*/
        int preco;                      /*preço do produto*/
        char descr[LSTR];               /*descrição do produto*/
}PRODUTO;

typedef struct con {
        int cod;                        /*codigo do produto*/
        int qtd;                        /*quantidade consumida*/
}CONSUMO;


typedef struct cartao {
        int id;                         /*identificador do cartao*/
        int prod_cons;                  /* numero de produtos consumidos */
        CONSUMO consumo[NCONS];
}CARTAO;

/*retorna um valor entre o min e o max*/
int valor_aleatorio(int min, int max)
{
        return (((float)rand()/RAND_MAX)*(max-min)+min);
}
/*inicializa o vector produtos*/
void inicializar_produtos(PRODUTO produtos[]) {
```

```
	int i;
	for(i=0;i<NPROD;i++){
		produtos[i].cod=i;
		sprintf(produtos[i].descr,"produto_%d",i);
		produtos[i].preco=valor_aleatorio(100,200);
	}
}
/*lista o vector produtos*/
void listar_produtos(PRODUTO produtos[]) {
	int i;
	printf("\nCodigo\tDescricao\tPreco unitario\n");
	for(i=0;i<NPROD;i++)
		printf("%d\t%s\t%d\n",produtos[i].cod,produtos[i].descr,produtos[i].preco);

}
```

**1.**

```
void entrada(CARTAO cartoes[],int *id_cartao) {
	if(*id_cartao<NCLI){
		printf("\n\n Cartao : %d\n",*id_cartao);
		cartoes[*id_cartao].prod_cons = 0;
		cartoes[*id_cartao].id= *id_cartao;
		(*id_cartao)++;
	}
}
```

**2.**

```
void consumo(CARTAO cartoes[],int id_cartao) {
	int id=-1,n;
	fflush(stdin);
	printf("\n\n Insira numero do Cartao : ");
	scanf("%d",&id);
	if (id >= id_cartao) {
		printf("\n Esse Cartao ainda nao foi atribuido!!!\n");
		getchar();
		return;
	}
	if(cartoes[id].prod_cons == -1) {
		printf("\n O cliente já saiu!!!\n");
		getchar();
		return;
	}
	if(cartoes[id].prod_cons >= NCONS) {
		printf("\n Cartao preenchido. É favor pagar e requerer novo Cartao!\n");
		getchar();
		return;
	}
	do{
		printf("\n Insira codigo do produto : ");
		scanf("%d",&n);
	}while(n<0 || n>=NPROD);
	cartoes[id].consumo[cartoes[id].prod_cons].cod=n;
	do{
		printf("\n Insira a quantidade: ");
		scanf("%d",&n);
	}while(n<=0);
	cartoes[id].consumo[cartoes[id].prod_cons].qtd=n;
	cartoes[id].prod_cons++;
}
```

**3.**

```
void pagamento(CARTAO cartoes[],int id_cartao, PRODUTO produtos[]) {
        int id=-1,temp,i,total=0;
        fflush(stdin);
        printf("\n\n Insira numero do Cartao : ");
        scanf("%d",&id);
        if (id >= id_cartao) {
                printf("\n Esse Cartao ainda nao foi atribuído!!!\n");
                getchar();
                return;
        }
        if(cartoes[id].prod_cons == -1) {
                printf("\n O cliente já saiu!!!\n");
                getchar();
                return;
        }
        printf("\n      - - - -  D I S C O T E C A  D A N C A  - - - -");
        if(cartoes[id].prod_cons == 0)  {
                printf("\n O cliente nao efectuou consumos! ");
                getchar();
                return;
        }
        printf("\n Cartao n: %d\n",cartoes[id].id);
        printf("\n Codigo\tDescricao\tQtd\tPreco unitario\tPreco");
        for(i=0;i<cartoes[id].prod_cons;i++) {
                temp = (cartoes[id].consumo[i].qtd * produtos[cartoes[id].consumo[i].cod].preco);
                printf("\n d\t%s\t%d\t\t%d\t%d",produtos[cartoes[id].consumo[i].cod].cod, produtos[cartoes[id].consumo[i].cod].descr,
                cartoes[id].consumo[i].qtd, produtos[cartoes[id].consumo[i].cod].preco,temp);
                total += temp;
        }
        printf("\n\n Total a pagar ------------------------------ %d\n\n\n", total);
        cartoes[id].prod_cons = -1;
  }

void main() {
        CARTAO cartoes[NCLI];
        PRODUTO produtos[NPROD];
        int id_cartao=0;
         int opcao;
        inicializar_produtos(produtos);
        do {
                printf("\n      - - - -  D I S C O T E C A  D A N CA  - - - -\n");
                printf(" 1 - Entrada cliente \n");
                printf(" 2 - Consumo \n");
                printf(" 3 - Pagamento \n");
                printf(" 4 - Produtos \n");
                printf(" 0 - Sair \n");
                printf(" Opcao --> ");
                scanf("%d",&opcao);
                switch(opcao) {
                        case 1: entrada(cartoes,&id_cartao);
                break;
                        case 2: consumo(cartoes,id_cartao);
                break;
                        case 3: pagamento(cartoes,id_cartao,produtos);
                break;
                        case 4: listar_produtos(produtos);
                break;
                }
        } while(opcao);
}
```