# LISTE SIMPLE

```
struct produs
{
	int* cod;
	char* denumire;
	float pret;
	float cantitate;
};

struct nodls
{
	produs* inf;
	nodls* next;
};

nodls* inserare(nodls* cap, produs* p)
{
	nodls* nou = (nodls*)malloc(sizeof(nodls));
	nou->inf = (produs*)malloc(sizeof(produs));
	nou->inf->cod = (int*)malloc(sizeof(int));

	*(nou->inf->cod) = *(p->cod);
	nou->inf->denumire = (char*)malloc((strlen(p->denumire) + 1) * sizeof(char));
	strcpy(nou->inf->denumire, p->denumire);
	nou->inf->pret = p->pret;
	nou->inf->cantitate = p->cantitate;
	nou->next = NULL;
	if (cap == NULL)
		cap = nou;
	else
	{
		nodls* temp = cap;
		while (temp->next != NULL)
			temp = temp->next;
		temp->next = nou;
	}
	return cap;
}

void traversare(nodls* cap)
{
	nodls* temp = cap;
	while (temp != NULL)
	{
		printf("\nCod=%d, Denumire=%s, Pret=%5.2f, Cantitate=%5.2f",
			*(temp->inf->cod), temp->inf->denumire, temp->inf->pret, temp->inf-
>cantitate);
		temp = temp->next;
	}
}

void dezalocare(nodls* cap)
{
	nodls* temp = cap;
	while (temp != NULL)
	{
```

```
		nodls* temp2 = temp->next;
		free(temp->inf->denumire);
		free(temp->inf->cod);
		free(temp->inf);
		free(temp);
		temp = temp2;
	}
}

void main()
{
	int n;
	printf("Nr. produse=");
	scanf("%d", &n);
	nodls* cap = NULL;
	produs* p;
	char buffer[20];
	for (int i = 0; i < n; i++)
	{
		p = (produs*)malloc(sizeof(produs));
		p->cod = (int*)malloc(sizeof(int));
		printf("Cod=");
		scanf("%d", p->cod);
		printf("Denumire=");
		scanf("%s", buffer);
		p->denumire = (char*)malloc((strlen(buffer) + 1) * sizeof(char));
		strcpy(p->denumire, buffer);
		printf("Pret=");
		scanf("%f", &p->pret);
		printf("Cantitate=");
		scanf("%f", &p->cantitate);
		cap = inserare(cap, p);
		free(p->denumire);
		free(p->cod);
		free(p);
	}
	traversare(cap);
	dezalocare(cap);
}
```