```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<iostream>
using namespace std;

//lista simpla de liste sipmle
struct student
{
	int varsta;
	char* nume;
	float medie;

};
struct nodLS
{
	student inf;
	nodLS* next;
};
struct nodLP
{
	nodLS *inf;
	nodLP* next;
};
nodLS* inserareLS(nodLS** capLS, student s)
{
	nodLS* nou = (nodLS*)malloc(sizeof(nodLS));
	nou->inf.varsta = s.varsta;
	nou->inf.nume = (char*)malloc((strlen(s.nume) + 1) *
sizeof(char));
	strcpy(nou->inf.nume, s.nume);
	nou->inf.medie = s.medie;
	nou->next = NULL;
	if (*capLS == NULL)
	{
		*capLS = nou;
	}
	else
	{
		nodLS* temp = *capLS;
		while (temp->next)
			temp = temp->next;
		temp->next = nou;

	}
	return *capLS;
}
```

```
void inserareLP(nodLP** capLP, nodLS* capLS)
{
	nodLP* nou = (nodLP*)malloc(sizeof(nodLP));
	nou->inf = capLS;
	nou->next = NULL;
	if (*capLP == NULL)
		*capLP = nou;
	else
	{
		nodLP* temp = *capLP;
		while(temp->next)
			temp = temp->next;
		temp->next = nou;
	}

}
void traversareLS(nodLS* capLS)
{
	nodLS* temp = capLS;
	while (temp)
	{
		printf("\n Varsta=%d Nume= %s, Media=%5.2f", temp-
>inf.varsta, temp->inf.nume
			, temp->inf.medie);
		temp = temp->next;
	}
}
void traversareLP(nodLP* capLP)
{
	nodLP* temp = capLP;
	while (temp)
	{
		printf("\nSublista:");
		traversareLS(temp->inf);
		temp = temp->next;
	}
}
void dezalocareLS(nodLS* capLS)
{
	nodLS* temp = capLS;
	while (temp)
	{
		nodLS* temp2 = temp->next;
		free(temp->inf.nume);
		free(temp);
		temp = temp2;
```

```
	}
}
void dezalocareLP(nodLP* capLP)
{
	nodLP* temp = capLP;
	while (temp)
	{
		nodLP* temp2 = temp;
		dezalocareLS(temp->inf);
		free(temp);
		temp = temp2;
	}
}
void main()
{
	int n;
	printf("Nr studenti=");
	scanf("%d",& n);
	student s;
	char buffer[20];

	//stundetii care au cod par intr-o lista
	//cod impar in alta lista
	nodLP* capLP = NULL;
	nodLS* capLSpar = NULL;
	nodLS* capLSimpar = NULL;
	for (int i = 0; i < n; i++)
	{
		printf("\n Varsta");
		scanf("%d", &s.varsta);
		printf("\Nume");
		scanf("%s", buffer);
		s.nume = (char*)malloc((strlen(buffer) + 1) * sizeof(char));
		strcpy(s.nume, buffer);
		printf("\Media");
		scanf("%f", &s.medie);
		if (s.varsta % 2 == 0)
			capLSpar =inserareLS(&capLSpar, s);
		else
			capLSimpar = inserareLS(&capLSimpar, s);

	}
	inserareLP(&capLP, capLSpar);
	inserareLP(&capLP, capLSimpar);
	traversareLP(capLP);
```

```
}
```