PROGRAMARE PROCEDURALA
- lab. 8-

1. Alocarea dinamică a memoriei

- Zona de memorie **heap** poate fi gestionată de către programator prin alocarea dinamică a memoriei.
- Funcţiile standard pentru alocarea dinamică se găsesc în „**alloc.h**” și „**stdlib.h**”.

Nume funcţie	Prototip funcţie/ Exemplu	Descriere funcţie

malloc	void *malloc(unsigned dim);
	
	int* p = (int *)malloc(20*sizeof(int));
	// se alocă 20*4 octeţi
		- alocă în heap o serie de zone de memorie succesive având o dimensiune de dim octeţi;
		- returnează pointer către void la prima locaţie de memorie;
		- pentru a păstra o dată de alt tip de date se va face conversie;
		- dacă nu există o zonă contiguă de memorie (de dimensiunea dată), se va returna NULL => verificare;
		- funcţia nu face iniţializare.

calloc	void *calloc(unsigned nr_elem, unsigned dim);
	
	int* p = (int *) calloc(20, sizeof(int));
	// se alocă 20*4 octeţi
		- alocă în heap nr_elem elemente de câte dim octeţi fiecare;
		- echivalentă cu apelul malloc(nr_elem*dim);
		- dacă nu există o zonă contiguă de memorie (de dimensiunea dată), se va returna NULL => verificare;
		- iniţializează fiecare element cu 0.

realloc	void *realloc(void bloc, unsigned dimNoua);
	
	int* p= (int*) realloc(int* a, 2*n);
		- realocă în heap un bloc(existent în memorie) de dimensiune dimNoua;
		- dacă nu poate fi realocat, se va returna NULL => verificare.

free	void free(void* p);
	
	free(p); // unde p poate fi oricare din exemplele de mai sus
		- eliberează zona de memorie indicată de p.

2. Liste simplu înlănţuite

- Fiecare nod are, pe lânga informaţie, legătură către următorul nod:

```
typedef struct nod{
        int info;
        struct nod* urm;
}t_nod;
```

- Dacă p este primul element din listă, atunci p->info reprezintă informaţia reţinută în nodul p, iar p->urm reprezintă nodul ce urmează după nodul p.
- Alocare: t_nod* p = (t_nod*) malloc(sizeof(t_nod));
- Se pot folosi ca stive(LIFO) sau cozi(FIFO).

3. Probleme

1. Se citește de la tastatură un șir de n numere întregi. Să se memoreze într-o listă liniară simplu înlănţuită și să se afișeze elementele listei.
2. Se citește de la tastatură un șir de n numere întregi și un număr natural k<n. Să se creeze o listă liniară simplu înlănţuită; să se insereze după al k-lea element din listă un nod nou cu valoarea 0; să se afișeze lista înainte și după operaţia de inserare.
3. Se citește de la tastatură un șir de numere întregi. Să se construiască o lista liniară simplu înlănţuită cu elementele citite. Să se găsească elementul maxim. Să se elimine maximul de pe poziţia care se află și să se insereze ca prim element al listei.

Notă: Toate problemele se vor rezolva construind funcţii.