Laboratory 2

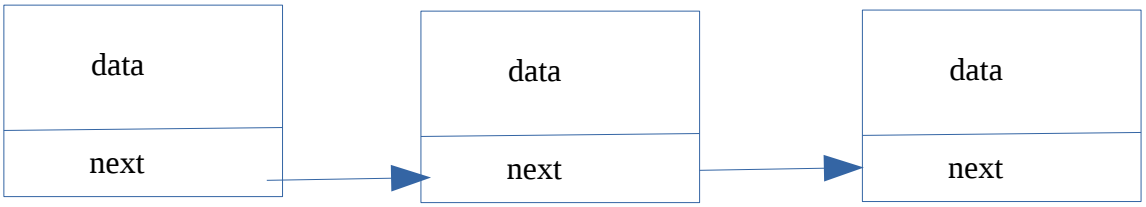
List data structure

based on instructions for “Programowanie w języku C 2” by mgr inż. Leszek Ciopiński

---

(Directional) List data structure

One of the most common used dynamic data structure is a list. A list is a structure which is used to store data for many elements (similarly as in a table, but in a different way). Each element of a list consists of the stored data and pointer to the next list element:



The advantages of list data structure:

- uses only as much memory, as is needed to store data elements (empty table uses the same memory as full table),
- it is expanding during adding new elements, we do not need to know the final number of elements.

Disadvantages:

- to access the appropriate element, we have to iterate through all previous elements.

The simplest implementation of a list which contains integers is as follows:

```
typedef struct strListElement {
   int data;
   struct strListElement *next;
 } ListElement;

 ListElement *first; /* pointer to the first element of a list */
```

The **typedef** keyword is used to create new types based on others, e.g.:

```
typedef old_name  new_name;
typedef int* PtrInt;
```

For dynamic data structures it is very helpful because it shortens the text which has to be written in order to create a variable of our structure.

In order to create the first element of a list, following code could be used:

```
first = malloc (sizeof(el_listy));
first->data = 2;
first->next = NULL;
```

The `NULL` value is used to store the information in a pointer that it does not points to any memory address. Also, that value may be used as an information that the element is the last one in our list.

Bidirectional list

The bidirectional list allows to iterate elements in both ways: forwards and backwards. The implementation is very similar, but pointer to the previous element should be added to the element structure.

Tasks:

1. Create implementation of a directional list which stores integer numbers. Implement also functions to add, delete and iterate through elements. Each operation should write text log into the console.

2. To the list from Task 1 add function to search for a specific number. The application should asks for a number and then show through which list elements the search function iterates. Also the search result should be written to the console.