Implement a C application for managing mutual investment funds. For this, you will use a double-linked list that contains pointers to elements of type MutualFund, structure with the following attributes: *mutual fund code* (**char\***), *mutual fund group* (**char\***), *risk level* (**int**), *net asset value* (**double**), *return of investment* (**float – percentage value**).

1. Create a double-linked list by inserting elements based on a selected criterion. The list must have at least 7 entries that are read from a file. ***(1 p)***
2. Write the function for printing all the mutual funds in the structure for both directions. ***(0.5 p)***
3. Write the function that counts how many mutual funds have the risk level greater than a given value, sent as a parameter. ***(1 p)***
4. Write the function that is determines the capital gain / loss for all the mutual funds. The function displays at the console the mutual fund code and the capital gain/loss by applying the positive / negative return of investments to the net asset value attribute. ***(1 p)***
5. Write the function that searches in the double-linked list for the first mutual fund that has the net asset value greater than a threshold specified as a parameter. The mutual fund is returned in **main()** and the values are displayed. ***(1 p)***
6. Write the function for creating an array with all the elements that have the nominal return of investment higher than a limit specified as a parameter and are part of a specified mutual fund group. The array doesn’t share HEAP memory space with the elements found in the double-linked list. The function is called in the **main()** and the result (elements stored in the array) is displayed on the console. ***(1.5 p)***

***The following indications are mandatory:***

*** Functions that are not tested in the main() function are not considered for evaluation; All implementations must be called in the main() function;***

*** Projects with compilation issues are NOT going to be evaluated;***

*** Source code that is commented is NOT going to be evaluated;***

*** The implementation must NOT trigger memory leaks;***