Week 6

Exercise 1: Basic operations of a linked list

Write a simple program that asks the user for numbers and adds them to a linked list, until the user finishes adding by entering 0.

The program then prints the numbers in the list, frees up reserved memory and stops the program from running.

Define an appropriate struct for the program and execute this program without functions. The program must have a self-defined struct and a data type made from it; use the format "%d" to print the list.

Add a newline to the end of each line.

Example run:

```
Enter elements:
Enter an element(Enter 0 to exit):
13
Enter an element(Enter 0 to exit):
8
Enter an element(Enter 0 to exit):
4
Enter an element(Enter 0 to exit):
3
Enter an element(Enter 0 to exit):
0
The following numbers are listed:
13 8 4 3
Clearing…
```

Exercise 2: Linked list with functions

Write a program that reads the car's brand name and model year from a text file and adds them to the linked list as its own fields. After reading the file, print the information on the screen and free the reserved memory.

Enter the text file name as a command line argument. Below is an example of a file (data.txt) from which data is read. Please use this file name in your submission.

In addition to the main program, the program must have 3 functions. In one function, the memory is allocated to a new record and values are set, and the record is added to the list as the last item. The second function takes care of printing the list data and the third takes care of freeing up memory. When processing memory items in functions, it is a good idea to send the address to the item of interest as a parameter and return the address of the appropriate item as a return value.

Add a newline to the end of each line

Example run: ./program data.txt

```
Items in the list are:
Lada 1976
Ferrari 2005
Suzuki 1985
Volvo 1963
Toyota 1993
Honda 2011
Clearing...
```

Exercise 3: Linked list with functions in a menu-driven program

Perform Week 5 Task 4 as a linked list instead of a array. Note that you will need to change the structure of the program somewhat and add the appropriate functions according to the task. If you want to change the length of the list, you can first clear it and then recreate it. Below are the program error messages and the previous week's assignment.

Error messages:

1. "The list is empty."

2. "List size can't be negative."

3. "Unknown selection, please try again."

Week 5 Task 4
Write a menu-based program that prints the numbers of an integer array and that can resize the current integer array. These operations should be repeated until the user wants to stop running the program. At the beginning of the program, you need to create a pointer to an integer array and a variable that maintains the size N of the array. The size has a default value of zero. When the user resizes the array, dynamically allocate the required amount of memory and fill it with the numbers 0 to N-1. When printing, the program prints all the values in the array on the same line, separated by a space. Memory allocation and printing should be performed in their own functions. Always make sure that the memory allocation is successful and free the allocated memory at the end of the program. Use "%d" format to print the integers.

Example run:

```
1) Print the items in the list
2) Resize the list
0) Stop
Select an Item:
1
The list is empty.
1) Print the items in the list
2) Resize the list
0) Stop
Select an Item:
2
Enter the size for the list:
4
1) Print the items in the list
2) Resize the list
0) Stop
Select an Item:
1
Items in the list are:0 1 2 3
1) Print the items in the list
2) Resize the list
```

```
0) Stop
Select an Item:
2
Enter the size for the list:
6
1) Print the items in the list
2) Resize the list
0) Stop
Select an Item:
1
Items in the list are:0 1 2 3 4 5
1) Print the items in the list
2) Resize the list
0) Stop
Select an Item:
2
Enter the size for the list:
0
1) Print the items in the list
2) Resize the list
0) Stop
Select an Item:
1
The list is empty.
1) Print the items in the list
2) Resize the list
0) Stop
Select an Item:
0
```

Exercise 4 Menu-based program for list management

Write a menu-based program for managing your list. The list must have the following functionalities (see example below):

- adding an item to the end of the list
- clearing the list, printing the list
- removing a node contatining a given elements
- printing the list
- exiting the program

Note that you already have implemented the other tasks but the removal. **Hint:** do not try to use functions in removal, it will get rather complicated that way. You need to update the pointers to the first and last element – there are different cases whether the removed element is the first one, the last one, or both!, or somewhere in the middle.

The node in the list must contain data as an **integer**. Print the list in the format "%d ".

Example run:

```
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
1
Enter the data:
13
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
1
Enter the data:
14
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
1
Enter the data:
15
```

```
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
4
Items in the list are:13 14 15
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
3
Enter the element you want to remove:
16
Item not found.
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
3
Enter the element you want to remove:
14
Items in the list are:13 15
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
4
Items in the list are:13 15
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
3
Enter the element you want to remove:
13
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
4
Items in the list are:15
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
```

```
0)Exit
1
Enter the data:
16
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
4
Items in the list are:15 16
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
2
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
4
The list is empty.
1)Add item to end of list
2)Clear list
3)Remove item from list
4)Print list
0)Exit
0
```