AUFGABEN (DEUTSCH)

Aufgabe 1 (2 PUNKTE): Entwerfen und implementieren Sie ein ADT Dynamic Array, das Integerwerte speichert. Die Größe des anfänglichen Arrays muss auf 10 gesetzt warden. Führen Sie Ihren Code aus und zeigen Sie das Ergebnis der folgenden Operationen nacheinander an (HINWEIS: Ich stelle Ihnen ein Beispiel main.cpp zur Verfügung. Sie dürfen aber auch Ihre eigene main.cpp Datei erstellen):

1. Erstellen Sie ein dynamisches Array
2. Geben Sie alle Elemente des Arrays und seine Größe aus
3. Fügen Sie zwei Elementen mit den Werten 100 und 200 dem Array hinzu
4. Geben Sie alle Elemente des Arrays und seine Größe aus
5. Fügen den Wert 500 an der Stelle 5 hinzu
6. Geben Sie alle Elemente des Arrays und seine Größe aus
7. Fügen Sie den Wert 2100 an der Stelle 21 hinzu
8. Geben Sie alle Elemente des Arrays und seine Größe aus

Reichen Sie den C++-Code (cpp-Dateien, h-Dateien und auch ein Makefile) für diese Anwendung zusammen mit einem Bildschirmfoto der Ausgabe der oben genannten Operationen ein.

Aufgabe 2 (3 PUNKTE): Entwerfen und implementieren Sie eine ADT für eine einfach verkettete Liste (Singly Linked List) in C++. Neben dem Einfügen- und Löschen, die Speicher zuweisen oder freigeben, wenn die Liste wächst oder schrumpft, sind andere übliche Operationen, die eine einfach verkettete Liste zu einem ADT machen, z.B. die Rückgabe der Anzahl der Elemente in der Liste, die Entsorgung der gesamten Liste, etc. Führen Sie Ihren Code aus und zeigen Sie das Ergebnis der folgenden Operationen nacheinander an (HINWEIS: Ich stelle Ihnen ein Beispiel main.cpp zur Verfügung. Sie dürfen aber auch Ihre eigene main.cpp Datei erstellen):

1. Erstellen Sie eine einfach verkettete Liste
2. Drucken Sie alle Elemente der Liste und ihre Größe aus
3. Fügen Sie 17 als letztes Element hinzu und dann geben Sie alle Elemente der Liste und ihre Größe aus
4. Fügen Sie 28 als letztes Element hinzu und dann geben Sie alle Elemente der Liste und ihre Größe aus
5. Fügen Sie als erstes Element 1 hinzu und dann geben Sie alle Elemente der Liste und ihre Größe aus
6. Fügen Sie das Element mit dem Wert -100 an Position 1 hinzu (HINWEIS: Position 1 ist das ERSTE Element!), dann geben Sie alle Elemente der Liste und ihre Größe aus
7. Fügen Sie das Element mit dem Wert 100 an Position 100 hinzu, dann geben Sie alle Elemente der Liste und ihre Größe aus
8. Fügen Sie den Wert als letztes Element hinzu, dann geben Sie alle Elemente der Liste und ihre Größe aus
9. Fügen Sie das Element mit dem Wert 2222 an Position 2 hinzu (HINWEIS: Position 1 ist das ERSTE Element!), dann geben Sie alle Elemente der Liste und ihre Größe aus
10. Fügen Sie das Element mit dem Wert 4444 an Position 4 hinzu (HINWEIS: Position 1 ist das ERSTE Element!), dann geben Sie alle Elemente der Liste und ihre Größe aus
11. Entfernen Sie das erste Element und dann geben Sie alle Elemente der Liste und ihre Größe aus
12. Entfernen Sie das erste Element und dann geben Sie alle Elemente der Liste und ihre Größe aus
13. Entfernen Sie das letzte Element und dann geben Sie alle Elemente der Liste und ihre Größe aus
14. Entfernen Sie das letzte Element und dann geben Sie alle Elemente der Liste und ihre Größe aus
15. Entfernen Sie das Element an Position 2 (HINWEIS: Position 1 ist das ERSTE Element!), dann geben Sie alle Elemente der Liste und ihre Größe aus
16. Entfernen Sie das Element an Position 2 (HINWEIS: Position 1 ist das ERSTE Element!), dann geben Sie alle Elemente der Liste und ihre Größe aus

17. Entfernen Sie das Element an Position 3 (HINWEIS: Position 1 ist das ERSTE Element!), dann geben Sie alle Elemente der Liste und ihre Größe aus

TASKS (ENGLISH TEXT)

Task 1 (2 POINTS): Design and implement an ADT Dynamic Array that stores integer values. The size of the initial array must be set to 10. To showcase your implementation run your code and show the result of following operations in sequence (NOTE: I provide you with an example main.cpp but you can create your own):

1. Create a dynamic array
2. Print out all the elements of the array and its size
3. Add elements with values 100 and 200 to the array
4. Print out all the elements of the array and its size
5. Add element 500 at location 5
6. Print out all the elements of the array and its size
7. Add element 2100 at location 21
8. Print out all the elements of the array and its size

Submit the C++ code (cpp files, h files, and also a makefile) for this application along with a screen shot of the output of the above operations.

Task 2 (3 POINTS): Design and implement a Singly Linked List ADT in C++. Beside insertion and deletion that can allocate or free memory as the list grows or shrinks, other common operations that make the singly linked lists an ADT are e.g. returning the number of elements in the list, disposing of the entire list, etc. To showcase your implementation run your code and show the result of following operations in sequence (NOTE: I provide you with an example main.cpp but you can create your own):

1. Create a singly linked list
2. Print out all the elements of the list and its size
3. Add 17 as last element then print out all the elements of the list and its size
4. Add 28 as last element then print out all the elements of the list and its size
5. Add 1 as first element then print out all the elements of the list and its size
6. Add at position 1 the element with value -100 (NOTE: position 1 is the FIRST element!) then print out all the elements of the list and its size
7. Insert element with value 100 at position 100 then print out all the elements of the list and its size
8. Add 3800 as last element then print out all the elements of the list and its size
9. Add at position 2 the element with value 2222 (NOTE: position 1 is the FIRST element!) then print out all the elements of the list and its size
10. Add at position 4 the element with value 4444 (NOTE: position 1 is the FIRST element!) then print out all the elements of the list and its size
11. Remove the first element then print out all the elements of the list and its size
12. Remove the first element then print out all the elements of the list and its size
13. Remove the last element then print out all the elements of the list and its size
14. Remove the last element then print out all the elements of the list and its size
15. Remove the element at location 2 (NOTE: position 1 is the FIRST element!) then print out all the elements of the list and its size

16. Remove the element at location 2 (NOTE: position 1 is the FIRST element!) then print out all the elements of the list and its size
17. Remove the element at location 3 (NOTE: position 1 is the FIRST element!) then print out all the elements of the list and its size

Submit the C++ code (cpp files, h files, and also a makefile) for this application along with a screen shot of the output of the above operations.