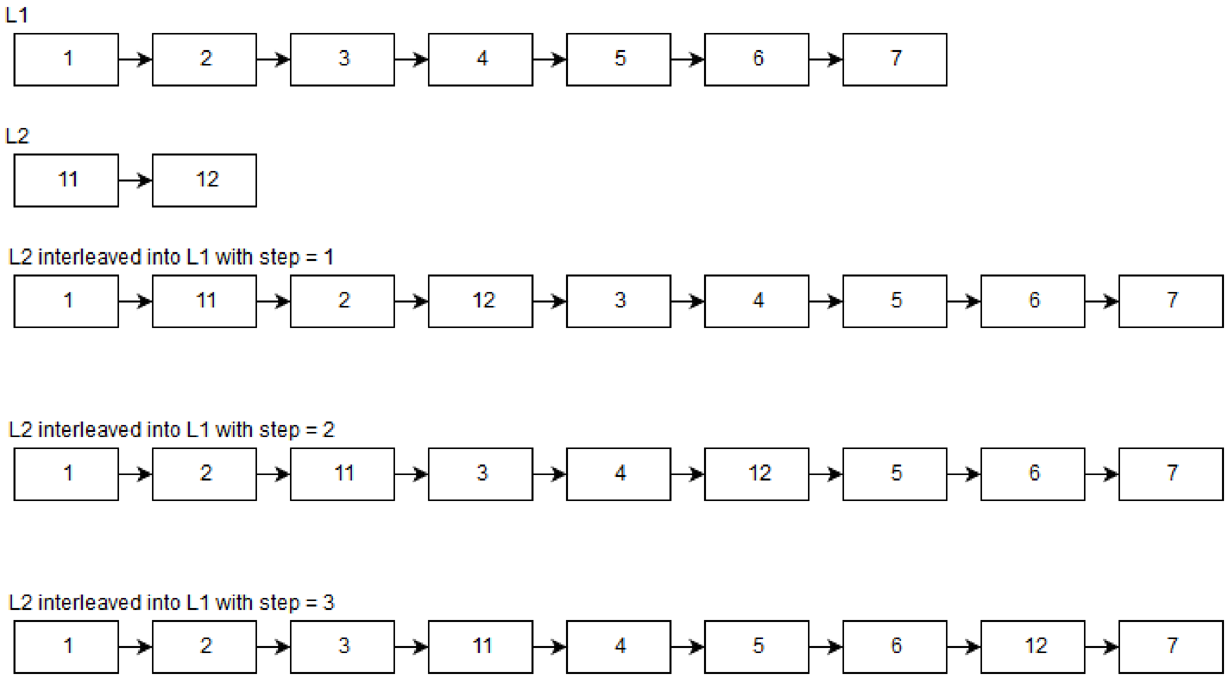
COP3530

Homework #2

Linked List.

In this assignment you have to implement the Linked List data structure and perform a number of interleave operations on different lists.

Interleave Operation: Consider L1 and L2 to be two Linked Lists. Interleaving L2 into L1 with step equal to n means putting every element of L2 after each n elements of L1. You can see the following examples to find out how the interleave operation exactly works:


L1
1 2 3 4 5 6 7
L2
11 12
L2 interleaved into L1 with step = 1
1 11 2 12 3 4 5 6 7
L2 interleaved into L1 with step = 2
1 2 11 3 4 12 5 6 7
L2 interleaved into L1 with step = 3
1 2 3 11 4 5 6 12 7

In this assignment, your program will first read an initial list from input then a number of lists follow which your program should interleave into the initial list with the given steps.

INPUT:

On the first line you can read the number of elements in the initial list, N.

On the second line you can read N numbers separated by space, which are the elements in the initial list.

On the third line you can read the number of following lists, M, which are going to be interleaved into the initial list.

On the following 3*M lines you can read the information about each list as follows: On each first line, a number S, showing the step of the interleave operation. On each second line, a number K, showing the number of elements in the list which is going to be interleaved into the initial list. And on each third line, K numbers separated by space showing the elements inside the aforementioned list.

Pay attention to the samples below to better understand how the input is going to be.

OUTPUT:

Your program should output the final list after all of the interleave operations, by printing the elements in one line, separated by spaces.

Sample input 1 (the blue statements are just for your information, they are not part of the input)**:**

```
5                 |     N, the number of elements in the initial list
1 2 3 4 5         |     the elements in the initial list
1                 |     number of additional lists
1                 |     step                 \
2                 |     number of elements   | Additional List #1
11 12             |     the elements         /
```

Sample output 1:

```
1 11 2 12 3 4 5
```

Sample input 2 (the blue statements are just for your information, they are not part of the input)**:**

```
5
1 2 3 4 5
2                 |     number of additional lists
1                 |     step                 \
2                 |     number of elements   | Additional List #1
11 12             |     the elements         /
3                 |     step                 \
2                 |     number of elements   | Additional List #2
21 22             |     the elements         /
```

Sample output 2:

```
1 11 2 21 12 3 4 22 5
```

Sample input 3 (the blue statements show the state of the initial list after each interleave operation, they are not a part of the input)**:**

```
3
1 2 3          |      initial list is: 1->2->3
5
1
2
1 2            |      now initial list is: 1->1->2->2->3
2
2
1 2            |      now initial list is: 1->1->1->2->2->2->3
3
2
1 2            |      now initial list is: 1->1->1->1->2->2->2->2->3
4
2
1 2            |      now initial list is: 1->1->1->1->1->2->2->2->2->2->3
5
2
1 2            |      now initial list is: 1->1->1->1->1->1->2->2->2->2->2->2->3
```

Sample output 4:

```
1 1 1 1 1 1 2 2 2 2 2 2 3
```

Notes

- The lengths of the lists will always be in a way that the second list would fit in the first list when doing the interleave operation. In other words the length of the first list is always bigger than the length of the second list multiplied by the step of the operation.
- **For any questions, please use the discussion section of Canvas.** Do not send emails for questions about the assignment.
- Your code is going to be compiled with **GCC**. It's your responsibility to make sure there are no compile errors. If your code generates compile errors you will receive no credit.
- You should submit a single cpp file. Do not archive (i.e. zip/rar/tar) your source file.