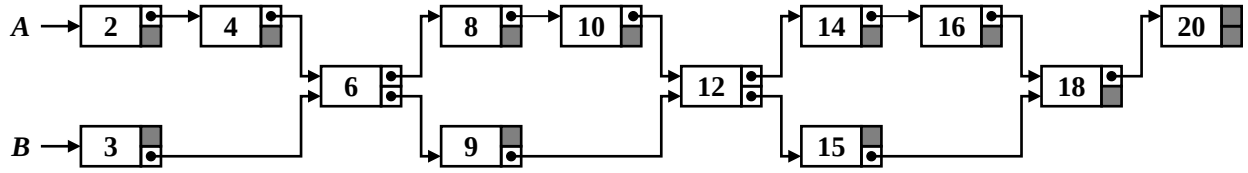
CS29003 Algorithms Laboratory

Warm-Up Assignment

Date: 08–January–2019 (Not for submission)

Chained Linked Lists

Let n, a, b be three positive integers, where a, b are small constant values. Your task is to create two sorted linked lists, the first consisting of all the positive multiples of a less than or equal to n, and the second consisting of all the positive multiples of b less than or equal to n. The common multiples of a and b are shared in the two lists. The following figure shows the lists for $n = 20$, $a = 2$, and $b = 3$. Your program should work for all values of a and b. That is, $a < b$, $a > b$ and $a = b$ should all be allowed. Moreover, a and b may or may not be coprime. Even a or b may be a multiple of the other.



In order to implement this structure, you need nodes, each containing a data field and two pointers (call them `next1` and `next2`). The first list (multiples of a) is to be linked using `next1`, whereas the second list (multiples of b) is to be linked using `next2`. Declare a user-defined data type to store such a node. The two linked lists are to be specified by two pointers A and B to the first nodes of the lists.

Write a function `createlist1()` that creates the first list on the integers $a, 2a, 3a, \ldots, \lfloor n/a \rfloor a$ using freshly allocated nodes. The multiples of a are to be inserted in the increasing order (that is, appended to the end of the list). The function should return the header pointer A. Its running time must be $O(n)$.

Write a function `createlist2()` to create the list on the multiples of b and to return the header pointer B. If a multiple of b is also a multiple of a, then the node storing this multiple in the first list is to be included. Otherwise, a new node is to be allocated to store the multiple. This function should also run in $O(n)$ time.

Write a function `prnlist()` to print each individual list. Supply the list header and optionally an indicator (first or second) to the function. An ambiguity arises only when both the two pointers `next1` and `next2` of the node pointed to by the pointer passed as the first argument are non-NULL.

Write a function `prnboth(A,B)` to print, in the sorted order, the union of the keys in the two lists. The function should run in $O(n)$ time. It must not destroy the links in any of the lists. Only the printing is to be done in the sorted order (without duplicate printing of the keys belonging to both the lists).

Write a `main()` function that reads n, a, b from the user, and creates the two lists by calling `createlist1()` and `createlist2()` in that sequence. After both the functions return, print the two lists individually, and then the union of the keys in the lists.

Sample output

```
n = 250
a = 21
b = 15

21 42 63 84 105 126 147 168 189 210 231

15 30 45 60 75 90 105 120 135 150 165 180 195 210 225 240

15 21 30 42 45 60 63 75 84 90 105 120 126 135 147 150 165 168 180 189 195 210 225 231 240
```

Submit a single C/C++ source file. Do not use global/static variables.