BA - Discussion 13, 2020-12-04

Announcements

- Exam 3 today from 4^{30}-6^{00} pm (EST) with 15 more minutes for uploading
  - → tomorrow at 7^{00} am (EST) for + time zones
  - → Becareful on the exam!
  - → you can use C but we truly don't recommend it... time waste!
  - → programs must follow good programming style but don't have to write comments unless specified by the problem
  - → don't make things harder than they are!!
  - → write general code
  - → Topics: ALL of the C topics covered
- Next Monday: in the morning, time for you to do Course Evals, and there will be Open Hours
  - → Lab attendance is required... it will be optional project presentations
- Final Project: due next Wednesday, 2020-12-09, at 9 am (EST) for EVERYONE!!!
  - → upload one document per group to Gradescope
  - → please make sure all group member names are on it

Review Quiz 8

Review of Material

- Shoutout what you want us to review! Otherwise, we'll go through Sample Exam and/or Study Session solutions

→ Linked Lists

→ pointers

→ don't forget typecasting and integer division!

→ generating random numbers!

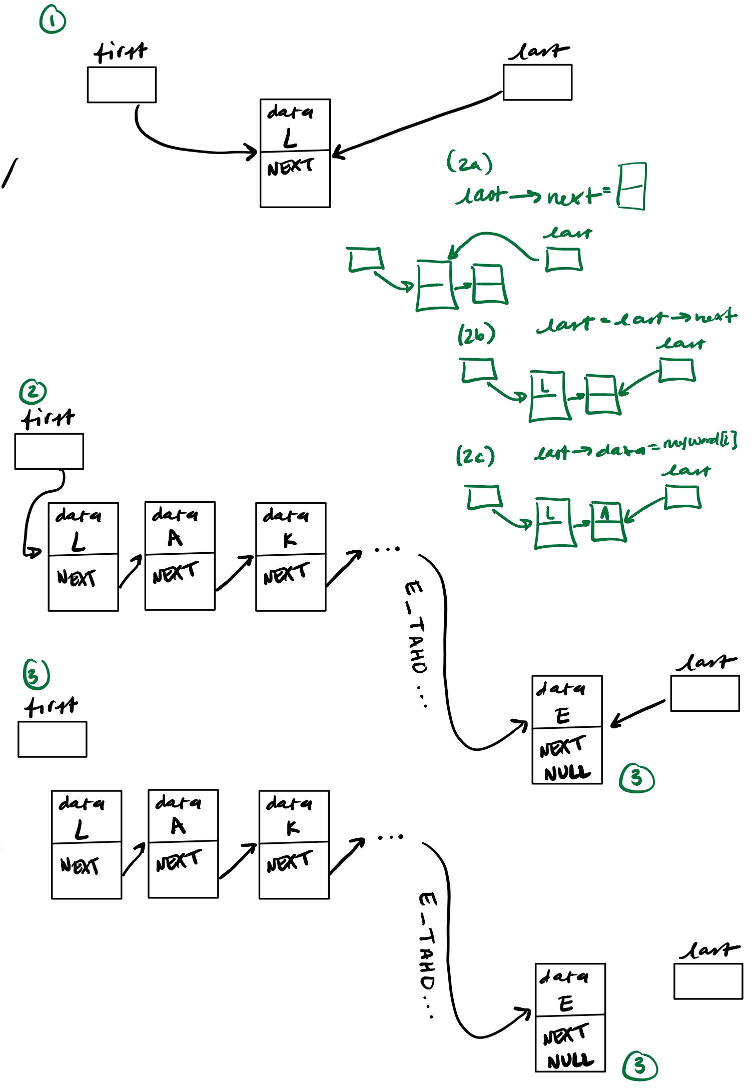
Generating Random Integers General Formula: `rand()%(MAX - MIN + 1) + MIN`

↑ put this on a sticky note right next to your computer!

```
typedef struct linked_list
{
        char data;
        struct linked_list *next;
}       element;
typedef element *elementptr;

void trav_print(elementptr);        ← function prototype

int main()
{
        elementptr first = NULL,
                         last = NULL;    } initializations/
        int i;                             declarations
        char myword[20] = "Lake_Tahoe";

 (1)    first = (elementptr) malloc(sizeof(element));
        last = first;
        first -> data = myword[0];

 (2)    for (i = 1; i < strlen(myword); i++)
        {
     (2a)   last -> next = (elementptr) malloc(sizeof(element));
     (2b)   last = last -> next;
     (2c)   last -> data = myword[i];
        }

 (3)    last -> next = NULL;
        trav_print(first);

        free(first);
        free(last);

        return 0;
}
void trav_print(elementptr fir)     ← function header
{
        elementptr current;

 (4)    current = fir;
        if (current == NULL)
                printf("There is no linked list!\n");
        else
        {
                while (current != NULL)
                {
                        printf("The data value is %c\n", current -> data);
                        current = current -> next;
                }
        }
}
```

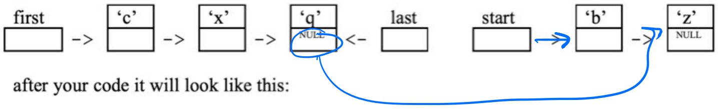
(4)

```
The data value is L
The data value is A
The data value is K
       ⋮          E
                  _
       ⋮          T
                  A
                  H
       ⋮          O
                  E
```

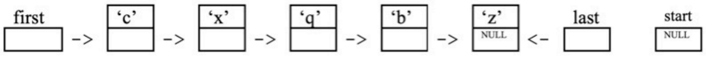


Exam 3 Sample Q#6

6) Two linked lists have been created in a program. The "primary" linked list is pointed to by "first" and "last" pointers. A pointer called "start" points to the beginning of a secondary linked list. You are to write code that will merge the two linked lists by adding the secondary linked list to the end of the primary linked list and then setting *start* to NULL. For example, if this is the scenario to begin with,



after your code it will look like this:



Q: first line of code...
last → next = start → next?
... can't do that because would skip the first struct in the 2nd linked list

Q: while loop? yes!

```
#include <stdio.h>
#include <stdlib.h>

typedef struct linked
{
   char code;
   struct linked *next;
}   elem_t;

typedef elem_t * elemptr;

void trav_and_print(elemptr);

int main()
{
  elemptr first, /* points to beginning of primary list */
          last,  /* points to end of primary list */
          start; /* points to beginning of secondary list */

  /* Assume that the two lists are initialized here, e.g.
       that first, last, and start are all initialized   */

①trav_and_print(first);
②trav_and_print(start);

③/* Add code here that will join the linked lists */


last → next = start;
do
  {
     last = last → next;
  } while (last → next != NULL)
start = NULL;
trav_and_print(first);
//free pointers
return 0;
void trav_and_print(elemptr f)
{
  elemptr current;

  current = f;
  printf("The data is: ");
  do
    {
      printf("%c ", current -> code);
      current = current -> next;
    }   while (current != NULL);

  printf("\n");
}
```

OUTPUT

① The data is: c x q

② The data is: b z

③ The data is: c x q b z