Ethan

1. Read input file
   Convert letter to numbers
   For loop (i <= 25) {
   Assign all positions to 0
   }
   Matrix[first letter][second letter] = 1;
   // if -u, then also include matrix[second letter][first letter] = 1;
2. AB, BA, BC, CB, CF, FC, CZ
3. 5, 4, 2, 1

1. Stack create:
   Create new stack of minimum length
   Use malloc or calloc

   Stack destroy:
   free(s);

   Stack empty:
   For loop goes through stack
   If element is 0, mark one
   If length of stack = marked zeros, it is empty

   Stack size:
   For loop that goes through the stack
   i++
   If stack element is 0, stop loop

   Stack push:
   If stack is full (top of stack == stack size), then reallocate memory *2;
   Entries[top] = item;
   Move top up one

   Stack pop:
   Move top down one
   Pointer to item = entries[top]

Stack print:
For loop that goes until end is reached (using stack length)
Print stack[i]
I++

PSEUDOCODE:
Start by creating the matrix
2 functions (valid and recursive function)
Valid tests if there is a 1 in the col and row
The recursive function:
For loop that asks if the curr_node and i are valid
If so, stack_push
Recurse with curr_node = i
Else:
stack_pop

EDIT: WOWZA I did not know what I was talking about. Here is the new answers:

1. while(fscanf("%c%c\n", &a, &b) != EOF) {
Assign res[a - 65][b - 65] = 1;
If u
Assign res[b - 65][a - 65]
}

3. 1, 3, 1, 2, 4, 2, 5