

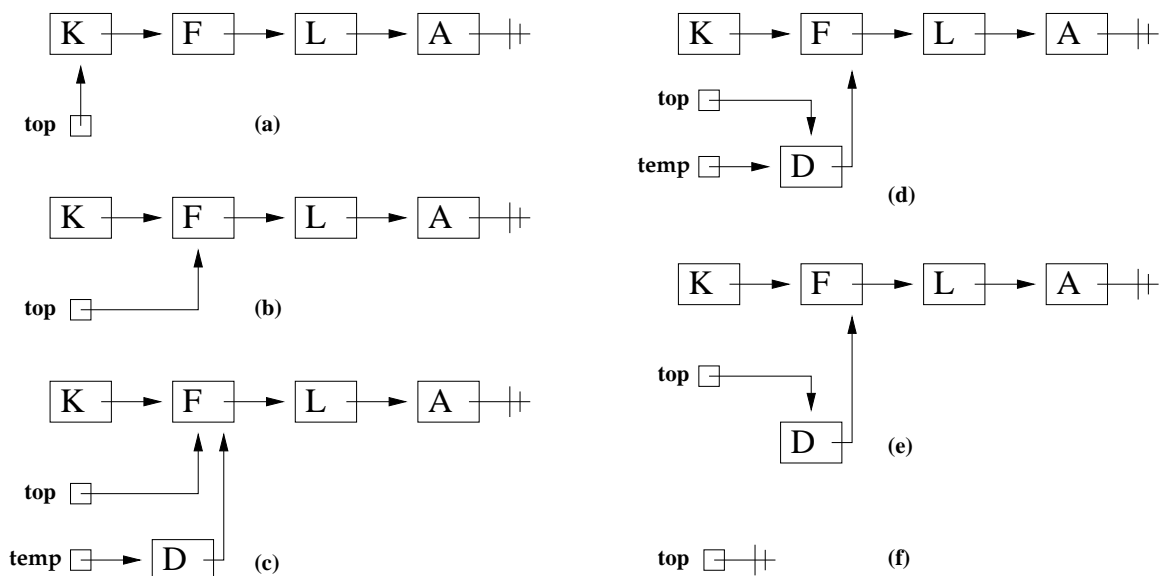
University of Nevada, Las Vegas Computer Science 477/677 Fall 2022

Assignment 5: Due Friday October 21, 2022, Midnight

Name: _____

You are permitted to work in groups, get help from others, read books, and use the internet. Please follow Mr. Wang's instructions on how to submit your completed assignment.

- During the test, there was confusion about the implementation of **stack**. One student told me afterwards that in a prior course, students were given an implementation of stack using a dummy node. That works, but I intended for you to use the other implementation, that is, without a dummy node. The following figure should clarify the issue.



The stack type is defined as follows.

```
struct stack
{
    char item;
    stack*link;
};
```

In the figure above, a stack of characters is accessed by the global variable `stack*top`, a pointer to the top item of the stack, as shown in (a), where the top item is K and the remaining items are F, L, A, in that order. (b) shows deletion of the top item. Garbage collection is not shown. The node holding K should be deallocated if you will need that space later in the program. (c) through (e) show insertion of the item H; the node with H is now the new top. (e) shows an empty stack.

This problem is to redo the question on the last exam, namely write C++ code for the three operators of stack.

```
char pop(stack*&top)
{
```

```
    top = top->link;
```

```
}
```

You need to finish writing the C++ code for **pop**. The ampersand “&” is a C++ operator that permits you to change the value of the argument during execution of the subprogram, as happens in one of the statements of the function.

Writing code for **push** is more involved. Use the figure for a guide.

```
void push(stack*&top,char letter)
```

```
{
```

```
}
```

Finally, write code for the Boolean function **empty**. This function does not “empty” the stack; it doesn’t change the stack at all. Instead, it returns “true” if the stack is empty, and “false” if the stack is not empty. Note the absence of the ampersand in the parameter list.

```
bool empty(stack*top)
{

}
}
```

Test your work by running the following code.

```
stack*s = NULL;
push(s, 'A');
push(s, 'B');
push(s, 'C');
push(s, 'D');
while(not empty(s))
{
    cout << pop(s) << endl;
}
```

2. Write a C++ program to find the Levenstein edit distance between two words. Here is a sample run of your program.

```
Enter a word of length no more than 20: quantumchromodynamics
That word is too long. Try again.
chromodynamics
Enter another word of length no more than 20: crommodynamiks
The Levenstein distance from chromodynamics to crommodynamiks is 3
```