C Course :: Fall 2008, Lab Session - III

21 September 2008

Problem 1

Problem

Implement a stack of integers using pointers in C.

About Stacks

A stack, also called a Last-In-first-Out (LIFO) system, is a linear list in which insertions and deletions can take place only at one end, called the top. This structure operates in much the same way as stack of books. If we want to place another book, it can be placed only at the top. Likewise, if want to remove a book from stack of books, it can only be removed from the top.

The insertion and deletion operations in stack terminology are known as push and pop operations.
See a live demo of stacks: http://www-cse.ucsd.edu/groups/tatami/kumo/exs/stack/

Initial state: There are two pointers (say topPtr and basePtr) and both point to same location in memory. Allocate suitable size (use malloc(STACKSIZE)) to the basePtr, where #define STACKSIZE 10

Your program should have the following functions:
Push(int i): checks whether the stack is full. If not, pushe the integer i onto top of the stack
Pop():checks is stack is empty or not. If not, remove one integer from the top of the stack.
PrintStack(): print the contents of the stack, Call this function after each Push() and Pop().

The program continuously asks the user whether he wants to do the Push() or Pop() operation and after each operation prints the elements currently in stack.

A Sketch of the program:

```
int main() {

        .

        .

        while(1) {
```

```
        printf(“Enter 1: to push, Enter 2: to Pop “);

        scanf(“%d”, &choice);

        if(choice == 1) {

            printf(“\nenter the element ”);

            scanf(“%d”, &element);

            push();

        }

        else if(choice == 2) {

            printf(“Popped integer = %d\n”, pop());

        }

        printStack();

    }

}
```