# Assignment 2
Computer Science Tutor

A stack is an important data-structure used in many applications such as compilers, communication protocols, interpreters, etc. A stack S stores data items and is defined by the following operations:

**Isempty(S)**
- equal to true, if S is empty
- equal to false, if S contains at least one element

**Push(S, data)**
- stores a given data element at the top of S.

**Pop(S)**
- if S is empty it will return a special null-data element
- if S is not empty it returns the data element at the top of S, while deleting the data element from the top of S.

**Top(S)**
- if S is empty, it will return a special null-data element
- if S is not empty, it returns the data element at the top of S.

In this assignment you are asked to implement a stack S that stores strictly positive integers. A user should be able to issue commands at the command line that have the following forms and results:

| | |
|---|---|
| **'e'** | the program will respond with 'S is empty', or 'S is not empty', if the stack S is empty, not empty, respectively. |
| **'P <number>'** | where <number> is a strictly positive integer, resulting in the <number> being pushed on the stack S. |
| **'p'** | if the stack is not empty, the program pops the top element from the stack S and reports its value, otherwise it reports 'Error S is empty'. |
| **'t'** | if the stack is not empty, the program reports the value of the element at the top of the stack S, otherwise it reports 'Error S is empty'. |
| **'q'** | the program stops. |