

1. Define and implement a structure representing a vulgar fraction with the following public methods:
 - the constructor with two integers given as the parameters,
 - read – reading the contents of the fraction,
 - print – printing the fraction,
 - isGreater – checking if the fraction is greater than the second one given as the parameter,
 - sum – calculating a fraction being the sum of this one and the second one given as the parameter,
 - product – calculating a fraction being the product of this one and the second one given as the parameter,

The program should create two fractions given by the user and test all the methods implemented. If you find helpful to write some additional methods, write them as well.

2. Define and implement a structure of an employee with the following fields: a forename, a surname, a position (being strings) and a salary being a real number. Implement the following methods of the structure:
 - the constructor with parameters for the all necessary fields of the employee (some of them may have the default values),
 - the destructor,
 - set & get methods for all the fields of the employee (i.e., getSalary, setSalary, etc.),
 - incSalary – increasing the salary of the employee by the value given as a parameter,
 - print – printing all the data of the employee.

Test all the structure capabilities – create an array of the employees and let the user manage them.

3. Write a program which implements a stack (a Last-In-First-Out structure) of unlimited size, aimed at storing real numbers, and a set of methods operating on this structure:
 - push – adding a real number to the stack (with memory allocation),
 - pop – removing from the stack a real number pushed there most recently (with memory deallocation) or raising an error / empty stack exception,
 - print – printing the values which are on the stack currently, starting from the one pushed recently,
 - isEmpty – informing whether the stack is empty,
 - height – returning the number of elements on the stack,
 - swap – swapping the content of the stack,
 - erase – removing all the items from the stack.

The program should use all these methods to make the operations as follows:

- 1) create a stack containing the numbers given by the user,
- 2) print the contents of the stack,
- 3) remove two values from the stack and print them,
- 4) print the number of values on the stack if it is nonempty, or an appropriate message otherwise,
- 5) swap the stack and print its content,
- 6) erase the stack.