

# BIL 108 - Computer Programming

## Final Examination

14.06.2012

1. (5 Pts) Write a “print” function which accepts an integer and prints it in an independent line as in the following examples:

Function call	Terminal Output
print(52)	0052
print(385)	0385
print(59821)	59821
print(7)	0007

2. (10 Pts) Write a function “atoi” which accepts a string and returns the integer represented by the string. For example, atoi(“123”) should return the integer 123.
3. (20 Pts) Write a **recursive** function “numOf Occ” which accepts a string, a key character and a limit character and returns the number of occurrences of key character in the string until the limit character is met. It should have the following prototype with no more input arguments:

```
int numOf Occ(char* str, char key, char limit)
```

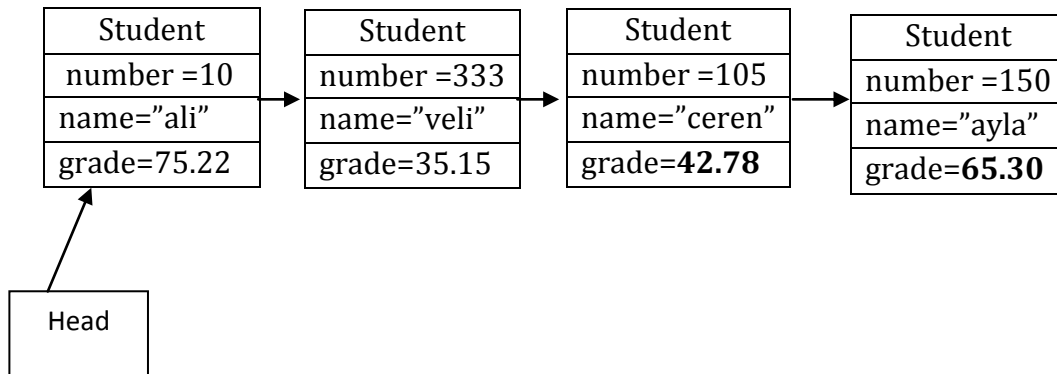
You are not allowed to use any helper functions. Some example runs are given below:

Function call	Return Value
numOf Occ(“a sample string for demonstration”, ‘s’, ‘d’)	2
numOf Occ(“a sample string for demonstration”, ‘s’, ‘z’)	3
numOf Occ(“a sample string for demonstration”, ‘w’, ‘z’)	0

4. (30 Pts) Write a function binToText which reads all the “Student” structures in a binary file and writes them to a text file, such that in the text file each record is hold in a separate line and some whitespace characters exist between variables. Your function will accept 2 file pointers; one for the binary source file and the other for the target text file as the input arguments. File names will be given to your main function as command line arguments (if you cannot do this, determine the names of the files in main function, which will cause you to lose 10 points). Student structure has the following members (**the order of the members is important**):

Student
int number
char name[20]
float grade

5. (40 Pts) Write a function "buildList" which reads all the Students objects in a binary file (alternatively creates at least 10 Student objects to be stored into the list with unique names and numbers – 5 points penalty), builds and returns a linked list holding the Student objects in the given order. Also, write a function "checkOrdering" which accepts a linked list, each node of which represents a Student object, as the input argument and finds the number of increasingly ordered successive elements in the list with respect to the grades. For example if the following list is given to your function, it should return 2.



General:

- You will submit only one source file with the name of "F\_<student number>.c" for all the questions below.
- **Write your name and number as the comment box in your program.**
- Write one function for each question and a main function which calls & tests all the other functions.