Recursion
Programming in C (Spring 2008/2009)
Assignment 4
Deadline: Sunday, June 21, 2009

Introduction
The purpose of this assignment is to practice the use of recursion in programming. Therefore, no loops and no pointers can be used in any of the questions. In addition, no library functions can be used (except for input/output where necessary).
Feel free to define additional functions as you see fit (e.g., recursive functions with extra parameters that are called from the functions that you are asked to write).
The required function signatures must exactly match those in the questions. It is recommended that you test the functions you write in the main function, but we won’t check main—the functions will be checked directly after we link our program with yours.
You can assume correct input.

1 Palindrome
Write a recursive function that checks whether a given sentence is a palindrome. A palindrome is a sentence that reads the same backwards, when punctuation and capitalization is ignored. For instance: “A Man, a Plan, a Canal: PANAMA!”.

int isPalindrome(char word[]);
The returned value is 1 or 0 (true or false). Remember that no loops are allowed. You will need to define a helper recursive function with extra parameters, and write isPalindrome as a wrapper function.

int isPalindromeHelper(char word[], int size);

2 Base
Write a recursive function that will compute the value of a representation of a number in given base.

int base(char num[], int radix);
For example, calling the function as
base("1011", 2)
should return 11, but calling it as
base("1011", 10)
should return 1011.
You need to support bases 2–36. “Digits” greater than 9 are encoded as lowercase or uppercase letters. E.g.,
base("Cafe", 16)
returns 51966 = 12 · 16³ + 10 · 16² + 15 · 16¹ + 14 · 16⁰.
Do not use loops or string / math library functions. You will probably need a helper recursive function, like with palindrome.

3 Merge
Write a recursive merge function. The function gets two sorted arrays (together with their sizes), and fills the third array using the contents of the two arrays, so that the third array is also sorted. The function should run in linear time, and no loops are allowed!

void merge(int x[], int xnum, int y[], int ynum, int z[]);

4 Subsets
Write a recursive function that fills the given character array with all possible given-size subsets of characters in given word, separated by commas. You can assume that the word is at most 10 in length.

void subsets(char word[], int sz, char arr[]);
The function then can be used as follows:
char w[] = "abcd";
char r[200];
subsets(w, 2, r);
puts(r);
This should cause the line
ab, ac, ad, bc, bd, cd
to be printed on the console. Pay attention to the order of the words—those need to be in the same order as one produced by the supplied executable!
Here, you can use functions from <string.h>.
Final Remarks

Remember: the output of your program should exactly match the output of the executable file, as published together with this assignment.
All questions regarding the assignment should be sent to Michael Orlov (orlovm@cs.bgu.ac.il) or to Achiya Elyasaf (achiya@bgu.ac.il).

Good luck!