

CGS 3460 Computer Programming Using C, Spring 2008

Homework 6

Due Wednesday, April 16 2008, before 11:59:59pm

Notes

- All submissions must be done electronically via the Courseworx system linked from the website.
- Create a separate C source file for each problem. Problem 1 must be in `p1.c` and Problem 2 in `p2.c`.
- Use `tar` to combine all the files into the file `H6.tar`. To do this, type `tar -cvf H6.tar p1.c p2.c` on the command line. You must only upload `H6.tar` to Courseworx.
- Once you've uploaded the file, download the file and untar it using the command `tar -xvf <filename>`. Display each extracted file to verify your programs are intact.
- Submit C source files *only* (files with `.c` extension). We will compile and run them.
- The first three lines of each C source file must contain your Full Name, UFID and Gatorlink ID as comments.
- Make sure that your code compiles and runs correctly on one of the following CISE machines: `sand.cise.ufl.edu`, `rain.cise.ufl.edu`, `shine.cise.ufl.edu`, `bay.cise.ufl.edu`.
- When obtaining input, be sure to prompt the user appropriately.

1. [Pointers] Write a function called `concat_n` which has the following signature:

```
char* concat_n(const char* s1, const char* s2, int n);
```

The function returns a string which has the first n characters of `s2` concatenated at the end of `s1`. If n is bigger than the length of `s2`, then all of `s2` is concatenated at the end of `s1`. *You are not allowed to use array subscripting; you must access arrays using pointers.* The only library function you are allowed from `<string.h>` is `strlen`. Write a `main()` function that demonstrates the use of this function by getting two strings from the user and a number, and displaying the result returned by this function.

2. [File handling] Consider the `Student` structure we defined in class. Write a program that writes/reads student records to/from file. Use binary files. Your program should first present a prompt with two choices, the first being to write to file and the second being to read from file and the second . If the user enters 1, then the program gives the exact same prompts as `student.c`, gets the student records in an array of structures and after this asks for a filename. The student records are written to this (binary) file. If instead the user enters 2 in the first menu, a filename is obtained from the user, and the file is read and all the student records in it are displayed. Your program should be able to correctly read the files it writes. You may use code from `student.c`, written in class, at will.