UCL COMPUTER SCIENCE

COMP0002 Programming Principles

Exercises 4

Purpose: Writing programs that use pointers and dynamic memory.

Goal: Complete as many of the exercise questions as you can. If you are keeping up, you need to do all the questions. Some questions are more challenging and are designed to stretch the more confident programmers. Don't worry if you can't do them all now, but be prepared to come back and try them later on.

Feedback: It is important that you get feedback on your exercise answers so that you know they are correct, that you are not making common mistakes, that the program code is properly presented and that you are confident you have solved the problem properly. To do this, get your answers reviewed by a lab demonstrator or programming mentor during a lab session.

NOTE: Keep a copy of all your exercise answers.

Q4.1 Write a version of the string copy function that takes a single string argument and creates a new copy in dynamic (heap) memory. The function signature should be:

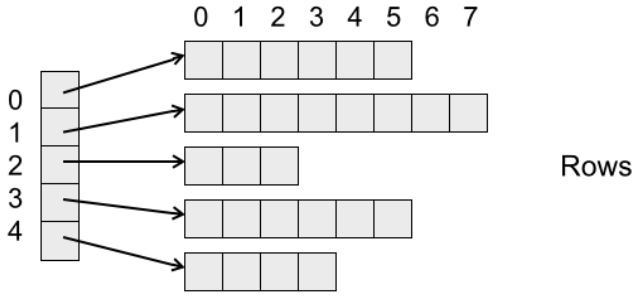
```
char* stringCopy(char *s)
```

Q4.2 Write a function with the signature `int strend(char *s, char *t)` that returns true (1) if the string pointed to by t occurs at the end of the string pointed to by s. Otherwise the function returns false (0).

Q4.3 Write a function with the signature `int* sort(int* n)` that takes a pointer to an array of integers, and returns a pointer to a new array containing the integers in sorted order.

Note that a sort function with the signature above is not given the size of the array to be sorted, and can only work with a fixed size array where the size is specified elsewhere in the program. A better signature would be `int* sort(int* n, int size)` that adds a second parameter giving the size of the array to be sorted. Modify a copy of your function, edit it to have this signature with a new name, and update the function body to use the size parameter.

Q4.4 a) Write a function to create and return a ragged 2D array data structure. A ragged 2D array has rows of different lengths. The basic data structure has an array of pointers that point to each row, each of which is an array of integers:



The function should take as a parameter the number of rows and and array that specifies the length of each row.

b) It would be better if the ragged array data structure also stored information about the number of rows and the length of each row. Modify your data structure so that it can store the size information, and also provide functions to create and free a ragged array, along with helper functions to get and set values in a ragged array and print out a ragged array.

Q4.5 Write a program that takes a block of text and formats it into two columns, so that the text can be printed on the screen in the two columns. Each column should be forty characters wide and text should be adjusted so that only complete words show on each line. Make use of pointers, strings (character arrays), with all memory allocated dynamically on the heap.

Additional challenges:

i) Read the original text from a file.

ii) Write the formatted text out to a file.

iii) Allow the number and width of columns to be specified.

iv) Add support for hyphenation, allowing words to be split across two lines.