

- 1.) Write a program to display all comments in a C program. The program is typed in at the keyboard, and the output displayed on the screen. It can also be tested on files by redirecting the standard input. Don't forget to handle quoted strings properly.

Furthermore, only comments that start with `/*` and end with `*/` need to be accounted for.

Implement the program in C, name the file `comment.c` (3.0%)

- 2.) There are many different ways to sort data. For some applications, e.g., making movies, it is better to sort pointers to the data, rather than sort the data itself. You must write a program to demonstrate this kind of activity.

The program must

- define data types for ... (0.5%)
  - (i) A pointer to an integer,
  - (ii) an array of five integers,
  - (iii) an array of five pointers to integers.
- Furthermore, the program, needs an initialization function that (0.5%)
  - (i) Initializes the elements of an array of five integers to random integers,
  - (ii) Use `rand` to generate the random integers, using `srand` with the return from `getpid` to seed the random number generation.
  - (iii) Initializes the elements of an array of five pointers to integers to point to the corresponding elements of the array of integers. (The structure is analogous to the one in [this example](#) that we discussed in class.)
- Have a function that prints an array of five integers. (0.5%)
- Have a function that prints the integers pointed to by an array of five pointers to integers. (0.5%)
- Have a function that uses a bubble-sort to sort an array of five integers, in ascending order of the integers. (0.5%)

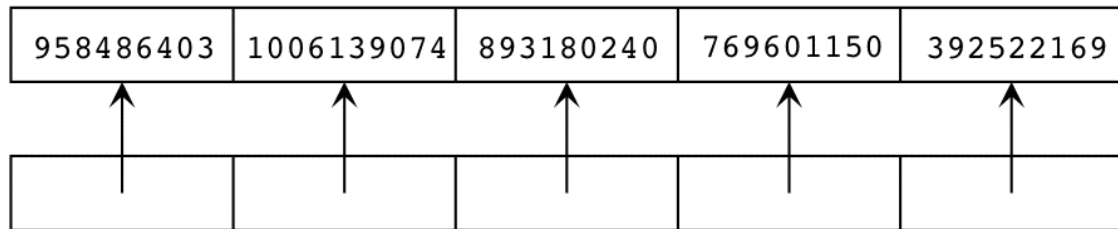
*And to make it fun, this function may not contain any [ or ] characters. (You may want to implement it with []s first, and then work out how to remove them later :-).*

- Have a function that uses a bubble-sort to sort an array of five pointers to integers, in ascending order of the integers pointed to by the pointers (i.e., do not sort the integer

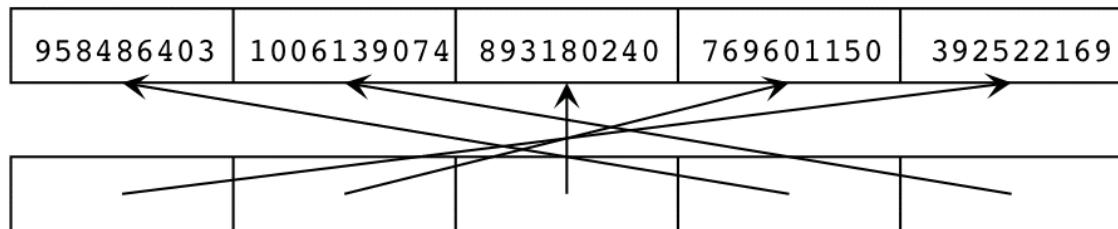
array - sort the array of pointers so that when you print it, the integers come out in ascending order). (0.5%)

- Have a main function that ...
  - (i) Declares an array of five integers and an array of five pointers to integers. Initializes the arrays (as described above).
  - (ii) Prints the array of integers.
  - (iii) Sorts the array of pointers (as described above). Prints the integers pointed to by the array of pointers.
  - (iv) Sorts the array of integers.
  - (v) Prints the array of integers.
  - (vi) Prints the integers pointed to by the array of pointers.
  - (vii) The picture shows the array of integers and array of pointers after initialization, sorting the array of pointers, and sorting the array of integers.

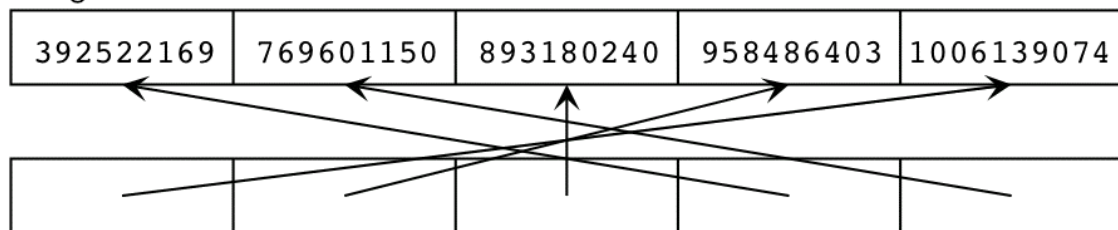
Initialised



Pointers sorted



Integers sorted



The output from a sample run should look like this.

---- Initialized array of integers ----

```
0 : 958486403
1 : 1006139074
2 : 893180240
3 : 769601150
4 : 392522169
```

---- Sorted array of pointers ----

```
0 : 392522169
1 : 769601150
2 : 893180240
```

```
3 : 958486403
4 : 1006139074
---- Sorted array of integers ----
0 : 392522169
1 : 769601150
2 : 893180240
3 : 958486403
4 : 1006139074
---- Array of pointers ----
0 : 1006139074
1 : 958486403
2 : 893180240
3 : 392522169
4 : 769601150
```

Implement the program in C, name the file `pointers.c` (3.0%)

Remarks:

- The assignment is due next Friday, Apr. 2nd, 11:59PM.
- You need to submit 2 `.c` files (`comments.c` and `pointers.c` with the corresponding C code using `submit2` (Please DO NOT submit compiled executables)! Furthermore, the C code needs to compile without any warnings (otherwise the TAs will not consider it)!