Lab 6: Pointers and Arrays

This lab will explore the usage of pointers and arrays to develop a generic menu function and to explore sorting algorithms.

6.1 Objectives

The objectives of this lab are to:

- Pass strings into a function.
- Use arrays and indexes to simplify patterns.
- Create a more interactive menu.
- Explore sorting algorithms.

6.2 Task 1: Menu Functions

This part of the lab will create different menu styles that will accept the same inputs and return the same outputs. The program will allow users to switch between the menu styles.

```
char titleText[80] = "Lab 6: Pointers and Arrays";
const int optionNum[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
const char *options[10] = {"Answers to ELEX 2205",
                           "Answers to ELEX 2220",
                           "Answers to MATH 2342",
                           "Answers to MECH 1210",
                           "Answers to PHYS 2164",
                           "Answers to ROBT 1270",
                           "Classic Menu Style..",
                           "Cursor Menu Style...",
                           "Scrolling Menu Style",
                           "Condensed Menu Style"};
```

The “system” function will be particularly useful for modifying the color or clearing the screen. This may add OS dependency

```
#include <stdlib.h>      // system
```

System allows commands available in command prompt or terminals to be accessible in C. Consider using the following command:

```
    system("COLOR 0A");
    system("CLS");
```

To avoid having the user hit enter each time scanf receives a character, use the kbhit (keyboard hit) function along with the getch (get character) function. These functions require:

```
#include <conio.h>       // kbhit, getch
```

The following lines will poll the user for a keypress and save that character.

```
        // Get user input.
        while(!kbhit());
        userInput = getch();
```

Write a function called generateMenu which will print out an array of menu items and allow the user to select a menu option by entering a numeric value. The prototype should be:

```
int generateMenu(char *, char*[], int);
```

Inputs:

- title – a string containing the title for the menu.
- items – a list of strings containing each menu item.
- nItems – the number of items.

Outputs:

- select – the value of the item selected by the user.

This menu should print out the title in a pleasant format. The menu items should be printed with the number the user must type to select an option.

```
Lab 6: Pointers and Arrays
--------------------------
0 - Answers to ELEX 2205
1 - Answers to ELEX 2220
2 - Answers to MATH 2342
3 - Answers to MECH 1210
4 - Answers to PHYS 2164
5 - Answers to ROBT 1270
6 - Classic Menu Style..
7 - Cursor Menu Style...
8 - Scrolling Menu Style
9 - Condensed Menu Style
```

Write a function called cursorMenu which will print out an array of menu items and allow the user to select a menu option moving a cursor up and down. The prototype should be:

```
int cursorMenu(char *, char*[], int);
```

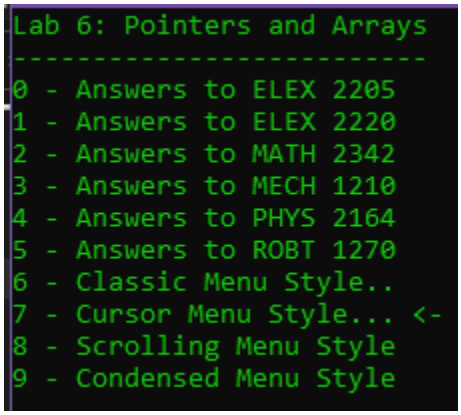
Inputs:

- title – a string containing the title for the menu.
- items – a list of strings containing each menu item.
- nItems – the number of items.

Outputs:

- select – the value of the item selected by the user.

This menu should print out the title in a pleasant format. A cursor can be moved up (by pressing ‘w’) or moved down (by pressing ‘s’). The can make a selection (by pressing ‘d’).



6.3 TASK 2: SORTING

Write a function called sortCoordinates that accepts an array containing x-coordinates and a separate array of equal length containing the y-coordinates.

Sort the coordinates by the sum of the two values in ascending order. Compare the sums of two coordinates and swap their positions if they are found to be out of order.

Use a couple integers to count and print how many comparisons and how many swaps occur.

x	y	x+y
23	8	31
-3	-5	-8
19	18	37
-16	12	-4
18	2	20
-27	6	-21
5	-23	-18
9	0	9
28	28	56
-10	3	-7

Sorted:

x	y	x+y
-27	6	-21
5	-23	-18
-3	-5	-8
-10	3	-7
-16	12	-4
9	0	9
18	2	20
23	8	31
19	18	37
28	28	56

6.4 DELIVERABLES

Please submit all code.

Deliverable	Description	Marks
6.1	Classic Menu	[5]
6.2	Cursor Menu	[5]
6.3	Sorted Coordinates	[5]
6.4	Sorted Coordinates Improved (Less comparisons and swaps).	[5]

There is no tolerance for plagiarism for any lab deliverables.