6.31 ***(Palindromes)*** A palindrome is a string that's spelled the same way forward and backward. Some examples of palindromes are: "radar," "able was i ere i saw elba," and, if you ignore blanks, "a man a plan a canal panama." Write a recursive function testPalindrome that returns 1 if the string stored in the array is a palindrome and 0 otherwise. The function should ignore spaces and punctuation in the string.

Source code.

```c
#include <stdio.h>
#include <ctype.h>
#include <string.h>

// This function will recursively test whatever text that has been put in and get rid of any spaces or punctuation.
int testPalindrome(char *str, int start, int end) {
    // if the beginning of the strings pointer is after the ends we know that we have covered the entire string and it can stop.
    if (start >= end) {
        return 1;
    }

    // Checks the current character if it is either a space or punctuation it will skip it.
    if (!isalnum(str[start])) {
        return testPalindrome(str, start + 1, end);
    }

    if (!isalnum(str[end])) {
        return testPalindrome(str, start, end - 1);
    }

    //checks if equal, if they are it will go on. if they arent we can end as they ar enot equal to each other.
    if (str[start] != str[end]) {
        return 0;
    }

    return testPalindrome(str, start + 1, end - 1);
}


int main() {
    char str[100];

    while(1){
        //gets the string, and then makes it lower case.
```

```
        fgets(str, sizeof(str), stdin);
        int length = strlen(str);
        strlwr(str);

        printf("%d\n", testPalindrome(str, 0, length - 1));
    }

    return 0;
}
```

Examples of this code.

```
1
racecar
1
Racecar
1
Rac e car!
1
Ra#CE CAR!
1
```

This shows the palindrome “racecar” being detected as a palindrome even when it is separated by spaces and when punctuation is added into it.

```
Palindrome
0
A Dog! Panic in a Pagoda
0
A dog A panic in a pagoda!
1
Do geese see god?
1
UFO Tofu
1
NotAPalindro me
0
```

When the phrase input to the program is not a palindrome it will return a 0 rather than a one.

7.31 ***(Calculator Using Function Pointers)*** Using the techniques you learned in Fig. 7.28, create a text-based, menu-driven program that allows the user to choose whether to add, subtract, multiply or divide two numbers. The program should then input two double values from the user, perform the appropriate calculation and display the result. Use an array of function pointers in which each pointer represents a function that returns void and receives two double parameters. The corresponding functions should each display messages indicating which calculation was performed, the values of the parameters and the result of the calculation.

Source code

```
#include <stdio.h>
```

```c
// Initializations of the functions.
void add(double a, double b);
void subtract(double a, double b);
void multiply(double a, double b);
void divide(double a, double b);

int main() {
    // using pointers for eahc of the
    void (*operations[4])(double, double) = {add, subtract, multiply, divide};

    int choice;
    double num1, num2;

    while(1){
        // scans information in for the calculator.
        printf("Select an operation:\n");
        printf("1. Add, 2. Subtract, 3. Multiply, 4. Divide.\n");
        printf("Enter your choice (1-4): ");
        scanf("%d", &choice);

        printf("First number: ");
        scanf("%lf", &num1);
        printf("Second number: ");
        scanf("%lf", &num2);

        //does the calculation if the input choice is valid.
        if (choice >= 1 && choice <= 4) {
            operations[choice - 1](num1, num2);
        } else {
            printf("Invalid choice\n");
        }
        printf("\n\n\n");
    }
    return 0;
}

// functions for each of the mathematical operations.
void add(double num1, double num2) {
    double result = num1 + num2;
    printf("Addition: %.2lf + %.2lf = %.2lf\n", num1, num2, result);
}

void subtract(double num1, double num2) {
    double result = num1 - num2;
    printf("Subtraction: %.2lf - %.2lf = %.2lf\n", num1, num2, result);
```

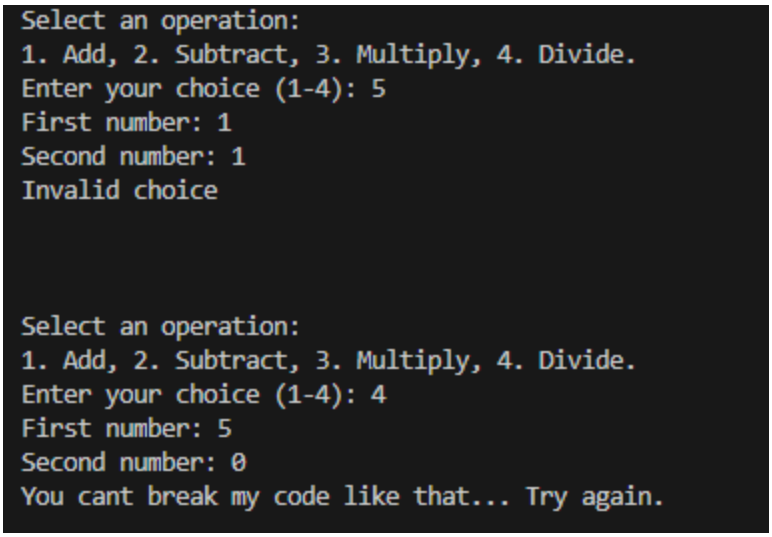
```c
}

void multiply(double num1, double num2) {
    double result = num1 * num2;
    printf("Multiplication: %.2lf * %.2lf = %.2lf\n", num1, num2, result);
}

void divide(double num1, double num2) {
    if (num2 != 0) {
        double result = num1 / num2;
        printf("Division: %.2lf / %.2lf = %.2lf\n", num1, num2, result);
    } else {
        printf("You cant break my code like that... Try again.\n");
    }
}
```

Code examples

```
Select an operation:
1. Add, 2. Subtract, 3. Multiply, 4. Divide.
Enter your choice (1-4): 1
First number: 1
Second number: 3
Addition: 1.00 + 3.00 = 4.00



Select an operation:
1. Add, 2. Subtract, 3. Multiply, 4. Divide.
Enter your choice (1-4): 2
First number: 6
Second number: 3
Subtraction: 6.00 - 3.00 = 3.00
```

```
Select an operation:
1. Add, 2. Subtract, 3. Multiply, 4. Divide.
Enter your choice (1-4): 3
First number: 6
Second number: 7
Multiplication: 6.00 * 7.00 = 42.00



Select an operation:
1. Add, 2. Subtract, 3. Multiply, 4. Divide.
Enter your choice (1-4): 4
First number: 18
Second number: 4
Division: 18.00 / 4.00 = 4.50
```

These are examples of each of the operations with a successful run for each of them.



This is some error checking for when the user puts in a selection for an operation that is outside of the 4 choices seen here. Also error checking for dividing by zero.

Write unix commands below.

1. Enter the command to get a brief description of the **pwd** command.

   Command: whatis pwd

2. What is your home directory (the directory you are in when you first log on)?

   Output: users/LukeErlewein

3. Without changing directories, enter the command to view a long listing of ALL the files in the **/ (root)** directory.
   a. Command: ls -la

   b. What is the name of a hidden file in that directory?

      .profile

4. Create a new subdirectory (under your home directory) called **public_html**. This directory will hold your web pages.

Command:_ mkdir ~/public_html

5. What are all of the permissions on the new directory?
   Drwxrwxr -x 2 lukeerlewein lukeerlewein 4096 May 20 18:07 /home/lukeerlewein/public_html

6. Using the octal method, change the permissions for the **public_html** directory to: read, write, and execute for yourself, execute for group and others. This makes your public_html directory available for browsing to your group and the world.

   Command: chmod 711 ~/public_html

7. Verify that the changes to the permissions are correct (in this case, the command may have more keystrokes, but we just want to see the permissions for the public_html directory, not all the rest of files and directories) .

   Command: ls -ld ~/public_html

8. Change directories to go to the parent of your home directory.

   Command: cd ..

9. What is the name of this directory (include entire path)?
   lukeerlewein@lukeerlewein-VirtualBox:/$

10. Using the option method (e.g. o=r), change the permissions for your own home directory to: read and execute for yourself, execute for group and others. The execute permissions allows your directory to also become web available.

    Command: chmod u=rx,g=x,o=x ~

11. Using the option method, change the permissions for your own home directory to: add write permissions for you; remove execute for the group.

    Command: chmod u+w,g-x ~

12. Create a new subdirectory of your home directory named FilterFiles.

    Command: mkdir ~/FilterFiles

13. Go to the FilterFiles directory.

Command: cd FilterFiles/

14. Create a shuffled file in your FilterFiles directory. Contents of the file are displayed in step 15.

    Command: I made a text file.

15. Verify that the contents of the file are:

```
1   Ali, Muhammad              Jan 17, 1942
4   Gandhi, Mohandas           Oct 2, 1869
33  Biden, Joseph              Nov 20, 1942
3   Lincoln, Abraham           Feb 12, 1809
11  Meir, Golda                May 3, 1898
```

    Please note that the numbers (1, 4, 33, 3, and 11) begin in column 1, the names begin in column 5, and the date of birth begins in column 32.

    Command: cat shuffled_file.txt

16. Sort the data based on the numbers (1,4,33,3,11) descending making sure they come out in the proper order (33,11,4,3,1).

    Command: sort -k1,1nr shuffled_file.txt

17. Sort the data based on the names in ascending order.

    Command: sort -k2 shuffled_file.txt

18. Sort the data based on the months.

    Command: I want to say that it should be sort -k4,3 shuffled_file.txt But it is not working for some reason I have no clue why

19. Print only the years from the shuffled file and redirect output to YearOnly file using operator >.

    Command: awk '{Print $6}' shuffled_file.txt > YearOnly

20. Sort the YearOnly file in descending order, placing the results in the YearSorted file and display the results on the screen (using one line command, rather than two separate commands).

    Command: Sort -nr YearOnly >YearSorted && YearSorted

21. Show only the lines that do not contain duplicate years from the YearSorted file.

Command: uniq yearSorted

22. Show only the years that contain duplicates from the YearSorted file.

    Command: uniq -d yearSorted

23. How many lines are in the /etc/passwd file?

    Answer: 49

    Command: wc -l /etc/passwd

24. Display the lines in the /etc/passwd file a page at a time (show both commands that will do this).

    a. Command: less /etc/passwd

    b. Command: more /etc/passwd

25. Print lines the first 10 lines from the /etc/passwd file.

    Command: head -n 10 /etc/passwd

26. Print the last 5 lines from the /etc/passwd file.

    Command: tail -n 5 /etc/passwd

27. Print lines 30-35 with the line numbers from /etc/passwd file.

    Command: head -35 /etc/passwd | tail -5 | nl