Designs 7 (Files)

Graded course work

Important Notes:

- **Write all the programs, from the designs in this handout**
- **Use the same program names, and variable names that I have specified**
- **Compile, run, and test your programs**
- **Submit a copy of these programs to me for grading**
- **If you don't know how to do any particular part, ask me -- I will show you how**

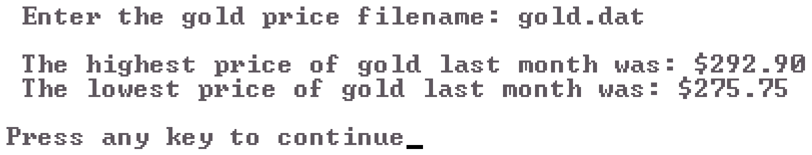
Program (gold3.c)

Based on the program temps3.c, write a program based on the variable declarations:

```
char file_name[15];  // variable for storing the file name

FILE *pointer_to_file;   // file pointer

int element_number = 0;   // loop counter

float highest_price = -1000,
      lowest_price = 1000,
      london_gold_closing[40];   // enough elements for a month
```

that will read data from the file gold.dat, and produce the ouput below.

Program output:



```
 Enter the gold price filename: gold.dat

 The highest price of gold last month was: $292.90
 The lowest price of gold last month was: $275.75

Press any key to continue_
```

Program (`phone1.c`)

Write this program in stages, hand in only the completed program.

Step 1

Write a telephone book program that uses a 5 element (line) array of structures with 3 fields: *name*, *address*, and *phone number*. Assume local numbers. (See program `student5.c`)

The program will ask the user to enter the name, address, and phone number for each person, and put them into the array.

The program will then print out the stored information.

Step 2

Create a data file (with 5 entries) named `phone2.dat`, using Notepad, in the format:

```
<person’s name>
<person’s address>
<person’s phone number>
       (etc)
```

Then change the program, to read the data from the file, into the array.

NOTE: When you read a line of text (string) from this file, you will also be reading the newline character (`‘\n’`) that terminates each line of the file. You will need to change this character to the null character (`‘\0’`), before you put the string into the field of the array. To do this, use the line of code:

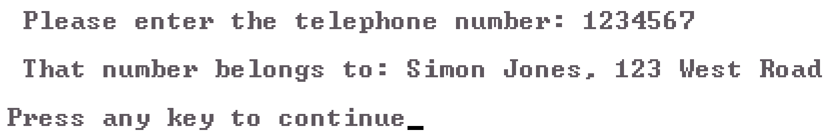
```
<the string>[strlen(<the string>) - 1] = '\0';
```

The program will then print out the stored information.

Step 3

Delete the lines of code that print out the stored information, and replace them with code that will ask the user for a telephone number. The program will then search through the array for that telephone number, printing out the name, and address of the person it belongs too if found, a not found message otherwise. (See program `states3.c`)

Program output:



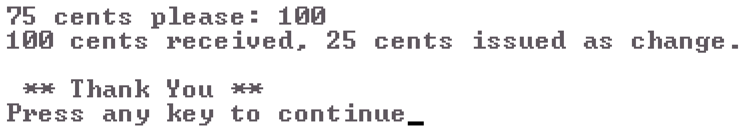
```
 Please enter the telephone number: 1234567

 That number belongs to: Simon Jones, 123 West Road

Press any key to continue_
```

Program (`soda1.c`)

Write a program that ask the user to pay for a 75¢ can of soda. The first prompt will read:

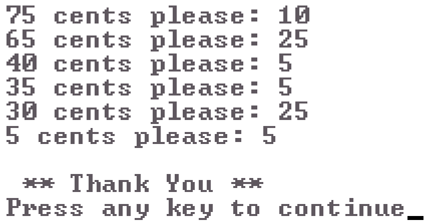
```
75 cents please:
```

If the user entered 100 ($1), the program will show:



```
75 cents please: 100
100 cents received, 25 cents issued as change.

 ** Thank You **
Press any key to continue_
```

If the user enters coins, the program might show:



```
75 cents please: 10
65 cents please: 25
40 cents please: 5
35 cents please: 5
30 cents please: 25
5 cents please: 5

 ** Thank You **
Press any key to continue_
```

NOTE: Keep this program simple. Use whole coins, do not enter larger coins than you are asked for.

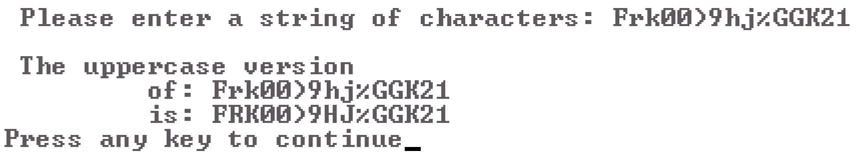
Program (`to_upper3.c`)

This program will loop through a mixed string of characters, converting lowercase letters, to uppercase letters.

Complete this program by adding your own functions `is_lowercase()`, and `lower_to_upper()` to this program.

```
/*
  Program to_upper.c
  written by: Joe Dorward
  Date: 07/15/00

  This program converts only the lowercase letters in a mixed string to
  their uppercase versions.
*/

#include <stdio.h>

int is_lowercase(char);   // function prototypes
char lower_to_upper(char);

void main(void)
{
char the_string[25];

int element_number = 0;

  printf(" Please enter a string of characters: ");
  gets(the_string);

  printf("\n The uppercase version \n \t of: %s \n",the_string);

  while (the_string[element_number] != '\0')
  {
    if (is_lowercase(the_string[element_number]))
    {
      the_string[element_number] = lower_to_upper(the_string[element_number]);
    }
    element_number++;
  }
  printf(" \t is: %s \n",the_string);
}
// ===============================================================================
    < add you function definitions here >
// ===============================================================================
```

Program output:



```
 Please enter a string of characters: Frk00>9hj%GGK21

 The uppercase version
         of: Frk00>9hj%GGK21
         is: FRK00>9HJ%GGK21
Press any key to continue_
```

Program (`to_lower.c`)

Make a copy of program `to_upper.c`, and change it to convert the uppercase letters in a mixed string to lowercase letters. Use your own functions `is_uppercase()`, and `upper_to_lower()`.

Extra Credit

Program (`interest1.c`)

Write a program write a program based on the variable declarations:

```
int deposit_term = 0,
    year_number = 0;   // used as loop counter

float deposited_sum = 0.0f,
      compounded_sum = 0.0f,
      annual_percentage_rate = 0.0f;
```

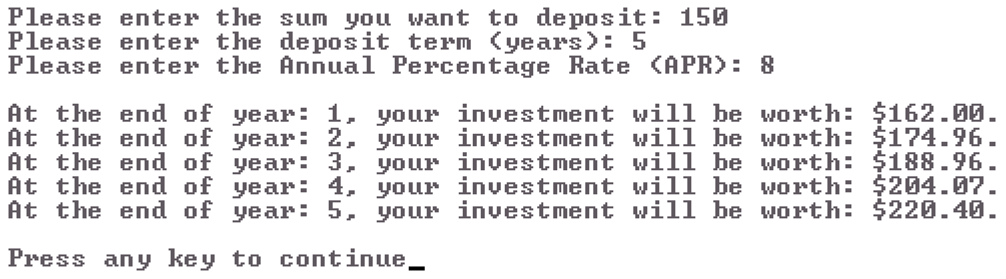
that will calculate the compounded interest on a sum of money deposited at a fixed rate of interest over a number of years.

The program should ask for:

- The sum being deposited
- The term of the deposit
- The rate of interest to be applied annually.

The program should calculate, and list, the value of the deposited sum at the end of each year.

Program output:



```
Please enter the sum you want to deposit: 150
Please enter the deposit term (years): 5
Please enter the Annual Percentage Rate (APR): 8

At the end of year: 1, your investment will be worth: $162.00.
At the end of year: 2, your investment will be worth: $174.96.
At the end of year: 3, your investment will be worth: $188.96.
At the end of year: 4, your investment will be worth: $204.07.
At the end of year: 5, your investment will be worth: $220.40.

Press any key to continue_
```