Designs 6 (Functions)

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 (`math1.c`)

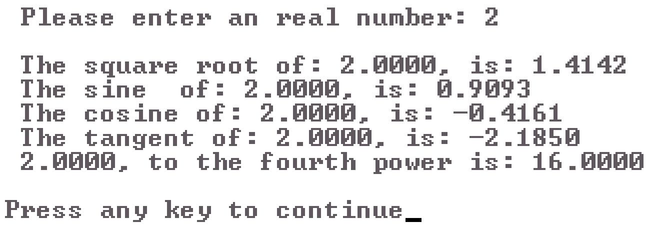
Make a copy of program `sqrt.c`, name it `math.c`, and add calls to the functions `sin()`, `cos()`, `tan()`, and `pow()`.

You can change the program to use only one variable, and call the `sqrt()` function in the `printf()` like:

```
  printf("\n The square root of: %0.4lf,",the_number);
  printf(" is: %0.4lf \n",sqrt(the_number));
```

You don’t need to know what these functions are to call them.

Program output:



```
 Please enter an real number: 2

 The square root of: 2.0000, is: 1.4142
 The sine  of: 2.0000, is: 0.9093
 The cosine of: 2.0000, is: -0.4161
 The tangent of: 2.0000, is: -2.1850
 2.0000, to the fourth power is: 16.0000

Press any key to continue_
```

Program (`calculate1.c`)

Copy the program `fun1.c`, name it `calculate1.c`, and add to it the function prototypes:

```
int sub_two(int,int);
int mul_two(int,int);
int div_two(int,int);
```

Then based on the function `add_two()` in the lecture handout, add function declarations for them, and call them in the program.

This program only needs two variables, and the function calls can be done inside a `printf()` statement too.

Program (even2.c)

Copy program even1.c, name it even2.c, and change the function prototype to:

```
char is_it_even2(int);
```

This function will return the character ‘e’ to the program if its integer parameter is even, and the character ‘o’ if the integer parameter is odd.

Re-write the program, and the function declaration to do this.

Note:
In C, characters must have a single quote mark around them.

Program (is_upper2.c)

Copy the program is_lower2.c, name it is_upper2.c, and change the function prototype to:

```
int is_uppercase(char);
```

This function will return 1 if the character parameter is an uppercase letter, and 0 otherwise.

Program (is_digit2.c)

Based on the program is_digit1.c, write a program with the prototype:

```
int is_a_digit(char);
```

This function will return 1 if the character parameter is a digit, and 0 otherwise.

Program (to_lower2.c)

Copy program to_upper2.c in the lecture handout, name it to_lower2.c, and change its function prototype to:

```
char upper_to_lower(char);
```

Then change the function, so that it returns the lowercase version of an uppercase letter.

Program (`strlen2.c`)

Write a program that asks the user to enter a string, the program then tells the user how many characters are in the string.

The program simulates the function `strlen()`.

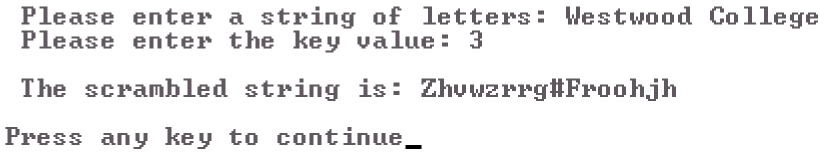
There are **NO** functions used in this program.

- Don’t use the function `strlen()`
- Strings end with the null (‘\0’) character
- Use a `while()` loop to inspect each element of the string
- Count the characters in the string until you find a null (‘\0’)

Program (`scramble1.c`)

Write a program that asks the user for a string of letters, and a “key” integer. The program will scramble the string by adding the value of the key to each character in the string. The program will then print out the scrambled string.

Program output:



```
 Please enter a string of letters: Westwood College
 Please enter the key value: 3

 The scrambled string is: Zhvwzrrg#Froohjh

Press any key to continue_
```