COMP0002 Programming Principles

Programming Notes and Exercises 3

Purpose: Writing programs that include functions.

Goal: Complete as many of the exercise questions as you can. If you are keeping up, you need to do at least the core questions. Some questions are more challenging and are designed to stretch the more confident programmers. Don't worry if you can't do them all now, but be prepared to come back and try them later on.

Feedback: It is important that you get feedback on your exercise answers so that you know they are correct, that you are not making common mistakes, that the program code is properly presented and that you are confident you have solved the problem properly. To do this, get your answers reviewed during a lab or mentor session.

NOTE: Keep a copy of all your exercise answers.

Reading Input from the Keyboard

There are various ways of reading input from the keyboard, or really to read from `stdin` that by default is connected to the keyboard. One way is to use `scanf`:

```
#include<stdio.h>
...
int n;
scanf(“%d”, &n); // Read an integer, can also use %i
```

The `scanf` function attempts to read from the *input buffer*. The input buffer holds the sequence of characters that the user types in, including the newline character. If the input buffer is empty when `scanf` is called the program is suspended until the user types their input and presses `<return>`. Then the entire line of input is copied into the buffer and the program allowed to resume. Hence, scanf gets to work on an entire line of input at one go, not single characters as they are typed.

The first parameter to the `scanf` function is the formatting string that determines how `scanf` will attempt to interpret the characters in the input buffer. The marker `“%d”` means try to convert the buffer contents into an integer value and store the value into the variable given as the second parameter. If the conversion is not possible, for example the user has not entered digit characters, then a zero is stored. The `&n` means pass a pointer to the variable `n` to `scanf`, this is a topic we will return to later in the module to explain.

Reading a double value is done the same way:

```
double d;
scanf(“%lf”, &d); // Read a double into d
```

If the input buffer contains several numbers separated by spaces, they can be read by a single `scanf`:

```
int a,b;
double c;
scanf("%d %d %lf", &a, &b, &c);
```

A side-effect of this behaviour is that if the user enters several numbers when your code expects to see only one, it will read one number but leave the other number and any other input in the buffer.

The next `scanf` will then read the next number from the existing contents of the input buffer and not wait for the user to type anything else.

If you want to clear the rest of the input buffer you can use this code to read the remaining characters and discard them:

```
char ch;
while ((ch = getchar()) != '\n' && ch != EOF);
```

`getchar` is another library function, which reads a single character.

Core Questions

Q3.1 Write a program to input the length of the sides of a triangle, and prints the area and length of the perimeter of the triangle. If the input values do not represent a triangle then display an error message instead.

> If a, b and c are the sides of the triangle then:
> perimeter = a + b + c
> and
>
> $$area = \sqrt{s(s-a)(s-b)(s-c)}$$
>
> where s is the *semiperimeter*
>
> $$s = \frac{1}{2}(a+b+c)$$

The calculation of the perimeter, semi-perimeter and area should be done by separate functions.

Hint: the standard C maths library provides a function called `sqrt` to calculate square roots. When compiling a program using the maths library, remember to add the -lm flag at the end of the command line if you get a linker error saying that a function cannot be found.

Q3.2 Write a program that defines and calls a function to raise an integer to a positive integer power (e.g., x^y). Provide a version of the function that uses a loop and another that uses recursion.

Q3.3 Write a program to determine if a long int (type long) entered via the keyboard is a palindrome (i.e., represents the same value when reversed, for example 123454321).

Q3.4 Write a program that uses a function to calculate the product of a sequence of numbers specified by the user. For example, if the user specifies 4 to 8, the function calculates 4*5*6*7*8. Any range can be used, including the use of negative numbers, and the program must correctly determine the values in the range.

Q3.5 Write a program that reads an integer between 0 and 999 and "verbalises it". For example, if the program is given 123 it would display "one hundred and twenty three".

Hint: Write functions to deal with ranges of numbers, such as single digits between "zero" and "nine", numbers between 10 and 19 and so on..

Q3.6 a) Write a program that asks for a number to be entered, stores the number as a long integer (type long) and uses a function to determine if it is a prime number. The program should reject any

input that cannot be recognised as a long integer (e.g., '1234p678') and ask the user to try again to enter a valid number. Spaces before or after the number are allowed but ignored. Otherwise, there should not be any other characters allowed (e.g., '12234 abc'), just the digits of the number and the newline ('\n') at the end of the input.

b) Modify the program so it will keep asking for new input numbers until the user enters 'stop'.

Q3.7 Twin prime numbers occur when the value of the two prime numbers differ by less than or equal to two. For example, 2 and 3, 3 and 5, and 11 and 13 are twin prime numbers. Write a program to print all the twin prime numbers between a specified range.

Q3.8 A strong number is defined as an integer where the sum of the factorials of the digits forming the number is equal to the original number. For example: 1! + 4! + 5! = 1 + 24 + 120 = 145. Write a program to find all the strong numbers within a specified range.

Q3.9 Write a program that inputs the time in 24 hour format and prints it out in 24 hour format. For example, 20:08 would be printed as 8.20pm. Note that scanf can be used to match patterns like this:

```
scanf("%d:%d", &hours, &minutes); // Match input like 20:08
```

Q3.10 Consider displaying a large digit formed from star characters:

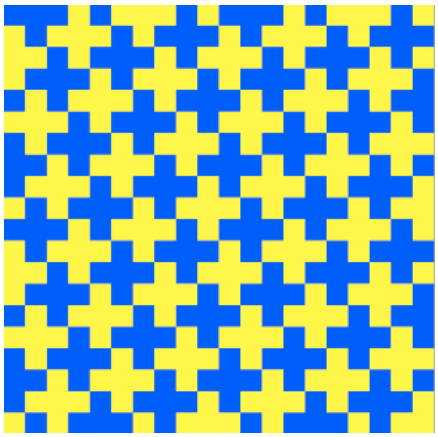
```
******
     *
******
*
******
```

Write a program that includes a set of functions for displaying each of the digits 0 to 9 and minus, in the way shown above. When one of the functions is called it should display *one line* of a large digit, with the line to display being given as a parameter (e.g., big2(3) would display the 3rd line of a big 2). A further function should be included that takes an integer argument and displays the integer in big digits, for example, 123 would appear as:

```
 *  ****** ******
**       *      *
 *  ****** ******
 *  *           *
*** ****** ******
```

Note: You can only output normal characters left-to-right, line-by-line, so think carefully about how your functions work.

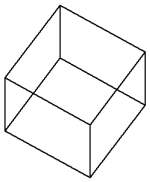
Q3.11 Write a drawing program to display this pattern:



Hint: it looks like a pattern of '+' symbols, but that is not the actual pattern to focus on!

Challenge Questions

Q3.12 Write a drawing program to display wire frame shapes, for example:



Q3.13 Write a program that inputs a date in the format dd-mm-yyyy and determines if the date is valid or not. If the date is valid, print the day name for the date and whether the year is a leap year or not.

Q3.14 Write a program that inputs two dates with the format dd-mm-yyyy and, if both dates are valid, prints the number of days between the two dates.

Q3.15 Find out about the Towers of Hanoi game and write a program to generate solutions.