CSC9V4 Practical 2

A simple problem

In any programming language there are many ways to solve a problem. In C there are often many more than in other languages. A useful starting point is to read source code. A classic `first program' is to separate the digits of a number, for which you have a program below.

1. Read the program **on paper**: you should be able to describe the meaning of each line in words.

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

int main()
{
  int num, num_copy, count=0;

  printf("Enter an integer number: ");
  scanf("%d",&num);

  // Count how many digits in the number, e.g. 3 for 123,
  // i.e. divide (integer division!) by 10 until you get 0.
  num_copy = num;
  while(num_copy){
      num_copy = num_copy / 10;
      count++;
  }
  num_copy = num; // save the original number again

  printf("Digits of given number are: ");

  // Divide (integer division!) by 10^(count-1) to get the
  // count-th left digit, i.e 123/100 = 1
  //
  // Get the reminder after the division, i.e. 123 % 100 = 23
  // Cycle
  int x;
  while(count){
    count--;
    x = (int) pow(10, count);
    printf(" %d ", num_copy / x);
    num_copy = num_copy % x;
  }

}
```

2. What is pow(a, b)?

3. Compile and run the program and verify that it works as expected. Can you make it crash/work in a wrong way?

IF YOU ARE USING NETBEANS:

Use the External Console (strongly recommended):

- In the Project pan (left), right-click on the project name → Properties.
- Now Run → Console Type → Set to 'External Terminal'
- Press OK.

This is necessary due to an issue with the way Netbeans handles the scanf function.

4. Now that you have understood how to pick digits in a number, write a short program that reverses a number, e.g. from 123 returns 321. Various solutions are possible...

Pushing the limit a bit further

Moving into something more complex, let us explore UPC codes (Universal Product Code), those that appear on barcodes: **Retrieve and check the validity in any UPC code**.

All UPCs consist of an 11-digit sequence followed by a check digit. Consider the example UPC of "036000241457". The last digit is the check digit "7", and if the other numbers are correct then the check digit calculation must produce 7. The algorithm, by example, is

1. Add the odd number digits: 0+6+0+2+1+5 = 14
2. Multiply the result by 3: 14 × 3 = 42
3. Add the even number digits: 3+0+0+4+4 = 11
4. Add the two results together: 42 + 11 = 53
5. To calculate the check digit, take the remainder of (53 / 10), which is also known as (53 modulo 10), and subtract from 10. Therefore, the check digit value is 7.

Write a program that computes the check digit of an 11 digit number following the above algorithm!

Hint: scanf() is rather powerful. You know that you are expecting 11 digits. You can either input the 11-digit number and then split it, OR use scanf() to input 11 separate digits - work out how!

Check Point