Chapter Three: Thinking Concurrently

do – while Loop

Apart from the for – loop there are two other ways of repeating. One is the do – while loop. This tests its expression at the end of the loop and if it is true, then the loop will continue.

```
do
{  statements
   ...
} while ( expression );
```

```
public static void main(String args[])
{  int   x;
   do
   {  x = IBIO.inputInt("enter a number less than 100 ");
   } while (x >= 100);
   IBIO.output("thank you");
}
```

In this example the program will continue to ask for a number if you type in a number larger than 100.

Pr 3.1 Change the program so that it only accepts numbers that are even and are larger than 0 and less than 100.

Primes

A prime is a number evenly divisible only by the number itself and one.

```
public static void main(String args[])
{  int i = 1;
   int x = IBIO.inputInt("Enter a number: ");

   do
   {  i++;
   } while (x % i != 0);

   IBIO.output(x + " is divisible by " + i);
}
```

This last program will accept a number and keep dividing it by 2,3,4,5,6, etc. until it finds one number that goes evenly into it. Note that it will always find one because the number goes into itself. So if a prime number was input into the program then the output would be that number itself, or else the output would be the smallest number that goes into it. Note that in the declaration of i it has also been initialised.

Pr 3.2 Change the last program so that it only accepts numbers that are greater than 1 and outputs the word prime if indeed the number is prime and otherwise outputs the smallest prime that goes into the number.

One useful way of testing your program is to put the main part into an infinite while loop.

```
do
{
   // main part of program to test here
} while ( true );
```

Digit sum

Given any number the following program will add up the digits in that number. So if 345 was entered, the program would calculate 3+4+5 = 12. It would start with n = 345, then it would divide 345 by 10 and write down the remainder, which is 5, then it would divide 345 by 10. Because we are dealing with whole numbers it would get a whole number answer and write down 34. This process continues until there are no more digits left in the number.

```
public static void main(String args[])
{
   int      sum = 0;
   int      n   = IBIO.inputInt(" enter a number ");

   do
   {  int   digit = n % 10;  // get right most digit
      sum = sum + digit;     // add to units digits
      n = n / 10;            // make new number
   } while ( n != 0);
   IBIO.output("the sum of the digits of the number is " + sum);
}
```

Pr 3.3 Change this program so that it will add up the cubes of the digits of the number. So if the input number was 345 it would go $3^3 + 4^3 + 5^3$.

Pr 3.4 Consider the sequence. If a number was even then the next number would be half of that number, if the number was not even then the next number would be got by multiplying that number by 3 and then adding 1. eg if 7 was the starting number then that number is odd so it is multiplied by 3 and 1 added to get 22, 7, 22, 11, 34. This sequence continues until it eventually arrives at 1. Write a program that will allow you to input a number and then it continues this sequence until it eventually arrives at 1. I want to know how many steps it takes. Eg starting at 3 the sequence is 3, 10, 5, 16, 8, 4, 2, 1 and that takes 7 steps.

while Loop

This is an alternative way of expressing a loop. In this form, the test is done first, before the execution of its statements.

```
while ( expression )
{  statements
   ...
}
```