1. Week 2: Before Arrays

1.1 Cash Machine (ATM)

Some cash dispensers only contain £20 notes. When a user types in how much money they'd like to be given, you need to check that the amount requested can be dispensed exactly using only £20 notes. If not, a choice of the two closest (one lower, one higher) amounts is presented.

Exercise 1.1 Write a program that inputs a number from the user and then prompts them for a better choice if it is not correct. For example :

```
How much money would you like ? 175
I can give you 160 or 180, try again.
How much money would you like ? 180
OK, dispensing ...
```

or :

```
How much money would you like ? 25
I can give you 20 or 40, try again.
How much money would you like ? 45
I can give you 40 or 60, try again.
How much money would you like ? 80
OK, dispensing ...
```

In this assessment you may assume the input from the user is "sensible" i.e. is not a negative number etc. ■

1.2 Higher-Lower

In the game "higher-Lower", a user has to guess a secret number chosen by another. They then repeatedly guess the number, being only told whether their guess was greater, or less than the secret one.

Exercise 1.2 Write a program that selects a random number between 1 and 1000. The user is asked to guess this number. If this guess is correct, the user is told that they have chosen

the correct number and the game ends. Otherwise, they are told if their guess was too high or too low. The user has 10 goes to guess correctly before the game ends and they lose. ■

1.3 Secret Codes

Write a program that converts a stream of text typed by the user into a 'secret' code. This is achieved by turning every letter 'a' into a 'z', every letter 'b' into a 'y', every letter 'c' into and 'x' and so on.

Exercise 1.3 Write a function whose 'top-line' is :

```
int scode(int a)
```

that takes a character, and returns the secret code for this character. Note that the function **does** need to preserve the case of the letter, and that non-letters are returned unaffected.

When the program is run, the following input:

```
The Quick Brown Fox Jumps Over the Lazy Dog !
```

produces the following output :

```
Gsv Jfrxp Yildm Ulc Qfnkh Levi gsv Ozab Wlt !
```

■

1.4 Planet Bob

On the planet Bob, everyone's name has three letters. These names either take the form of consonant-vowel-consonant or else vowel-consonant-vowel. For the purposes here, vowels are the letters `{a,e,i,o,u}` and consonants are all other letters. There are two other rules :

1. The first letter and third letters of the name must always be the same.
2. The name is only 'valid' if, when you sum up the values of the three letters ($a = 1, b = 2$ etc.), the sum is prime.

The name "bob" is a valid name: it has the form consonant-vowel-consonant, the first letter and third letters are the same ('b') and the three letters sum to $19(2+15+2)$, which is prime. The name "aba' is **not** valid, since the sum of the three letters is $4(1+2+1)$ which is **not** prime.

Exercise 1.4 Write a program that outputs all the valid names and numbers them. The first few names should look like :

```
 1 aca
 2 aka
 3 aqa
 4 bab
 5 bib
 6 bob
 7 cac
 8 cec
 9 ded
10 did
11 dod
12 dud
13 ece
14 ege
```

```
15 eme
16 ese
17 faf
```

■

1.5 Monte Carlo Π

At :

`http://mathfaculty.fullerton.edu/mathews/n2003/montecarlopimod.html`

a square whose sides are of length r, and a quarter-circle, whose radius is of r are drawn.

If you throw random darts at the square, then many, but not all, also hit the circle. A dart landing at position (x,y) only hits the circle if $x^2+y^2 \leq r^2$.

The area of the circle is $\frac{\pi}{4}r^2$, and the area of the square is r^2.

Therefore, a way to approximate π, is to choose random (x,y) pairs inside the square h_a, and count the h_c ones that hit the circle. Then:

$$\pi \approx \frac{4h_c}{h_a} \tag{1.1}$$

Exercise 1.5 Write a program to run this simulation, and display the improving version of the approximation to π. ■

1.6 Hailstone Sequence

Hailstones sequences are ones that seem to always return to 1. The number is halved if even, and if odd then the next becomes `3*n+1`. For instance, when we start with the number 6, we get the sequence : `6, 3, 10, 5, 16, 8, 4, 2, 1` that has nine numbers in it. When we start with the number 11, the sequence is longer, containing 15 numbers : `11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1`.

Exercise 1.6 Write a program that :

- displays which initial number (less than 50,000) creates the **longest** hailstone sequence.
- displays which initial number (less than 50,000) leads to the **largest** number appearing in the sequence.

■