Programming with C

Terry Marris November 2010

4 The Double Number Type

Previously we looked at some string handling functions. Now we look at arithmetic with numbers of type *double*.

4.1 Arithmetic Operators and Precedence

Numbers of type *double* are the familiar numbers with a decimal point. We can perform the usual arithmetic operations: add +, subtract -, multiply * and divide / on them.

The familiar rules of precedence apply: brackets first, then multiplication and division, then addition and subtraction. For example, in working out the Celsius equivalent of a given temperature in degrees Fahrenheit

celcius = 5 * (Fahrenheit - 32) / 9

the brackets are worked out first, then the multiplication, then the division.

4.2 Constants

The essential property of a constant is that its value remains unchanged during program execution.

```
const double pi = 3.1416;
```

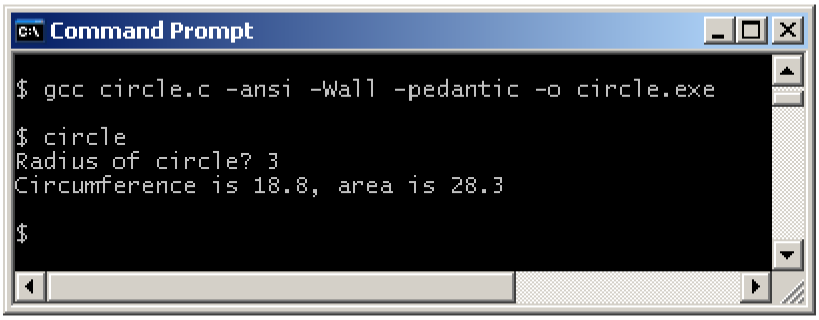
const is an example of a *type qualifier*. It specifies that the object it refers to cannot be modified during a program run. *const* is used to define π in the program shown below. The program calculates the area and circumference of a circle from a given radius.

```
/* circle.c: circumference and area of a circle */

#include <stdio.h>
#include <stdlib.h>

int main()
{
  const double pi = 3.1416;
  char string[BUFSIZ];
  double radius, circumference, area;

  printf("Radius of circle? ");
  gets(string);
  radius = atof(string);
  circumference = 2 * pi * radius;
  area = pi * radius * radius;
  printf("Circumference is %0.1f, area is %0.1f\n",
                                        circumference, area);
  return 0;
}
```



```
$ gcc circle.c -ansi -Wall -pedantic -o circle.exe

$ circle
Radius of circle? 3
Circumference is 18.8, area is 28.3

$
```

Since *radius*, *circumference* and *area* are all variables of type *double*, we can declare them all on the same line.

```
double radius, circumference, area;
```

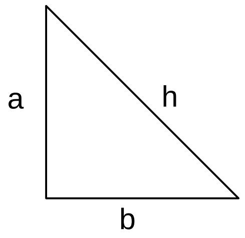
Advantages of using constants include:

- maintenance: if the value of a constant is to be changed by a programmer, the change is made in one just one place in the program. If a constant was not used you would have to search through the entire program looking for every occurrence of the value to be changed.

- readability: a descriptive name e.g. *pi*, is easier to understand than some arbitrary value e.g. 3.14159265.

4.3 Power and Square Root

The next program uses the familiar theorem of Pythagoras to calculate the length of the third side of a right-angled triangle.



$$h^2 = a^2 + b^2$$

$$\therefore \quad h = \sqrt{a^2 + b^2}$$

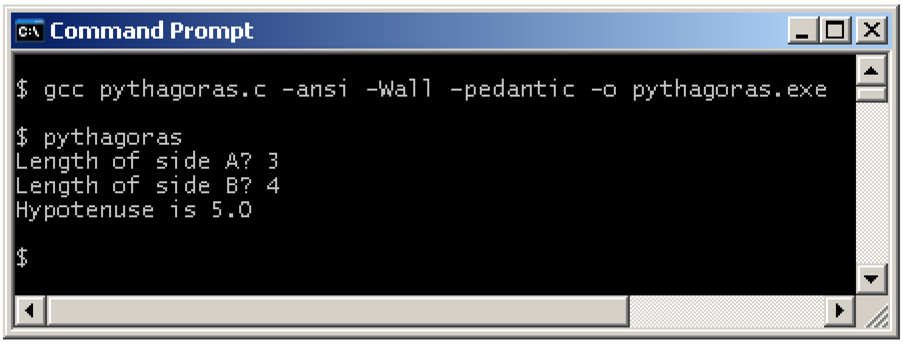
The C library *math.h* contains the functions *pow()* and *sqrt()* that raises a number to a given power, and finds the square root of a number respectively. *pow(a, 2)* will raise *a* to the power of *2*. *sqrt(X)* will give the square root of *X*. Of course, *X* should not be negative. *sqrt()* and *pow()* are used in the program shown below.

```
/* pythagoras.c: finds the hypotenuse of a right-angled triangle */

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

int main()
{
  char string[BUFSIZ];
  double a, b, h;

  printf("Length of side A? ");
  gets(string);
  a = atof(string);
  printf("Length of side B? ");
  gets(string);
  b = atof(string);
  h = sqrt(pow(a, 2) + pow(b, 2));
  printf("Hypotenuse is %0.1f\n", h);
  return 0;
}
```



```
$ gcc pythagoras.c -ansi -Wall -pedantic -o pythagoras.exe

$ pythagoras
Length of side A? 3
Length of side B? 4
Hypotenuse is 5.0

$
```

4.4 The Trigonometric Functions

The trigonometric functions, defined in *math.h*, include *tan()*, *sin()* and *cos()*. These trig functions require angles to be provided in *radians*. This is no problem provided we remember that

$$\text{radians} = \text{degrees} * \frac{\pi}{180}$$

and

$$\text{degrees} = \text{radians} * \frac{180}{\pi}$$

We use the *sin()* function in the next program.

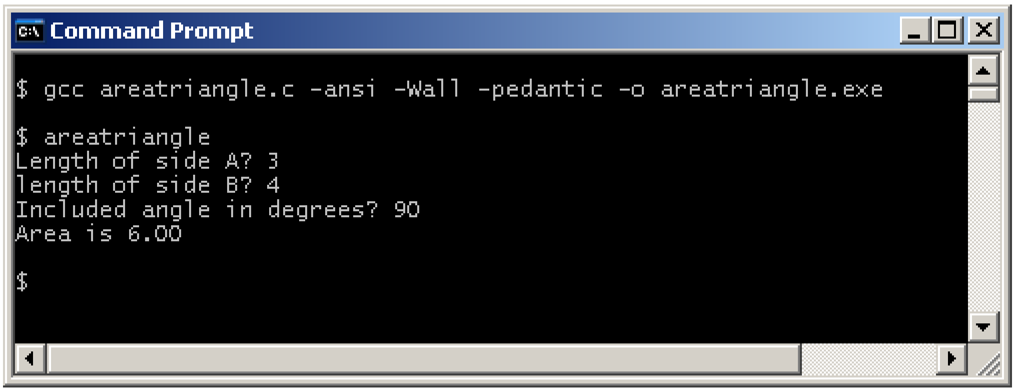
The area of a triangle is a = ½ab sin(θ) where θ is the angle included by sides a and b.

```
/* areatriangle.c: calculates the area of a triangle */

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

int main()
{
  const double pi = 3.1416;
  char string[BUFSIZ];
  double a, b, degrees, radians, area;

  printf("Length of side A? ");
  gets(string);
  a = atof(string);
  printf("length of side B? ");
  gets(string);
  b = atof(string);
  printf("Included angle in degrees? ");
  gets(string);
  degrees = atof(string);
  radians = degrees * pi / 180;
  area = 0.5 * a * b * sin(radians);
  printf("Area is %0.2f\n", area);
  return 0;
}
```



```
$ gcc areatriangle.c -ansi -Wall -pedantic -o areatriangle.exe

$ areatriangle
Length of side A? 3
length of side B? 4
Included angle in degrees? 90
Area is 6.00

$
```

Since the included angle is 90°we have a right-ang led triangle. The area of a triangle is

½ base x perpendicular height = ½ x 3 x 4
= 6

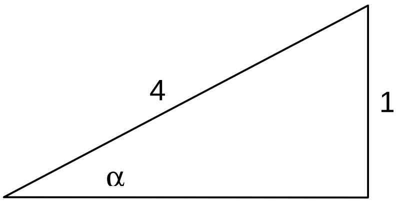
which is the same as previously calculated using the ½ a b sin(θ) formula.

It is often useful to check the result of a calculation by using a different, independent method.

4.5 The Inverse Trigonometric Functions

The inverse trig functions include *acos()*, *asin()* and *atan()*. You would use the inverse trig functions to determine the angle involved. For example ...

A cyclist comes across a near vertical hill with a gradient of 1 in 4. This means that for every four metres travelled, the road rises (or falls) one metre.



$$\sin\alpha = \frac{1}{4}$$

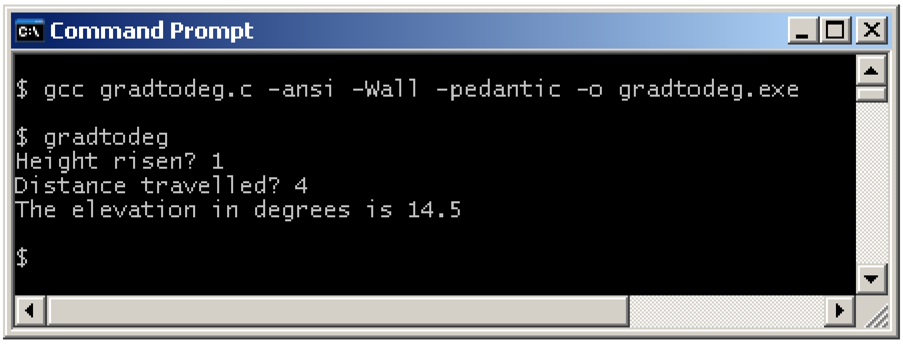
$$\sin\alpha = 0.25$$

$$\Rightarrow \alpha \approx 14^\circ \text{(from tables)}$$

We take a look at the program.

```
/* gradtodeg.c: converts a gradient to its equivalent in degrees */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
  const double pi = 3.1416;
  char string[BUFSIZ];
  double h, d, ratio, radians, degrees;
  printf("Height risen? ");
  gets(string);
  h = atof(string);
  printf("Distance travelled? ");
  gets(string);
  d = atof(string);
  ratio = h / d;              /* FAILS if d = 0 */
  radians = asin(ratio);
  degrees = radians * 180 / pi;
  printf("The elevation in degrees is %0.1f\n", degrees);
  return 0;
}
```



```
$ gcc gradtodeg.c -ansi -Wall -pedantic -o gradtodeg.exe

$ gradtodeg
Height risen? 1
Distance travelled? 4
The elevation in degrees is 14.5

$
```

The ratio is defined as height risen divided by distance travelled. If the distance travelled is zero then the ratio is undefined since you cannot divide by zero. We see how to deal with this kind of problem in a later chapter.

There are some restrictions on the ratios. If you are using *asin(ratio)* or *acos(ratio)*, the ratio must be between 0.0 and 1.0 inclusive. If you are using *atan(ratio)*, the ratio must be between 0.0 and about 573.0.

Exercise 4.1

1. A bicycle's gear ratio is traditionally calculated from

gearRatio = wheelDiameter x chainWheel ÷ rearSprocket

A ratio of 100 would help you speed down hills. A ratio of 30 will help you toil up them. Typically, a wheel size is 27 inches, the number of teeth on a chainwheel (which is attached to the cranks, which are themselves attached to the pedals) is between 52 and 32, and the number of teeth on a rear sprocket between 13 and 28. Write and test a program that will input wheel size, and the number of teeth on both a chainwheel and a sprocket, and which outputs the gear ratio correct to one decimal place.

2. Write and test a program that converts kilometres to miles. You might like to use the fact that 1 km = 0.621 miles.

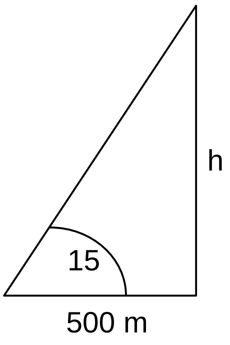
3. Hero's formula for finding the area of a triangle is

$$A = \sqrt{s(s - a)(s - b)(s - c)} \text{ where } s = \frac{a + b + c}{2}$$

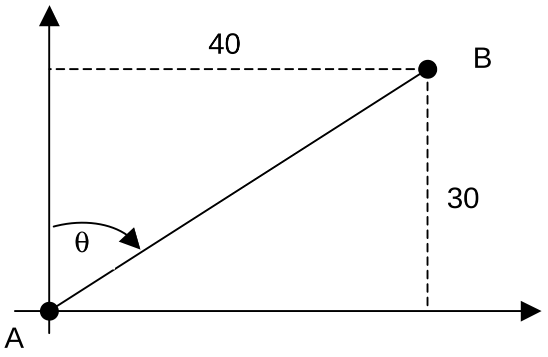
Write and test a program that inputs the lengths of three sides of a triangle and outputs its area.

4. The volume of a sphere is $v = \frac{4}{3}\pi r^3$ where r is its radius. Write and test a program that inputs the radius of a sphere and outputs its volume.

5. Surveying. It is useful to know the height of tall buildings since they are a hazard to low flying aircraft. At 500m from the base, the angle of elevation to the top of a building is 15°. What is the height of the buildin g?



6. Navigation. An airplane pilot. currently flying at point A, is seeking to reach point B, which is 40 miles east, and 30 miles north from A. What bearing should the pilot be flying on?



We have seen how to perform arithmetic with numbers of type *double*, and how to use some of the functions provided in *math.h*.

Next we see examine the numeric data type *integer*.

Bibliography

Kernighan B and Ritchie D *The C Programming Language* Prentice Hall 1988
Mark Williams Company *ANSI C A Lexical Guide* Prentice Hall 1988
Godfrey C and Siddons A *Four-Figure Tables* Cambridge University Press 1962