

# C Programming Exercises

## Data Types

2.1 Use the internet to find a complete list of C's keywords.

2.2 Write a program that outputs:

```
*****
* Hello World *
*****
```

2.3 Write a program that inputs two numbers and then calculates their average.

2.4 Print out the value of PI; use the *atan* function from the math library.

---

## Expressions

3.1 Write a program that inputs a 4 digit year and then calculates whether or not it is a leap year.

3.2 Calculate the number of days in the inclusive date range “1<sup>st</sup> January 2000” to “1<sup>st</sup> January 3000”.

3.3 Write a program that prints out the largest *int*, *long*, *float*, *double* and *long double*. You should use the header files `<limits.h>` and `<float.h>`.

---

## Pointers

4.1 Which of the following code fragments compile and for those that do, write down the final values of the all variables used:

```
a)  int x = 100;
    int* p = &x;
    (*p)--;

b)  int x = 100;
    int y = 200;
    int* p1 = &x;
    int* p2 = &y;
    p1 = p2;
    (*p1)--;

c)  int* p1;
    int* p2;
    x = 100;
```

```
p1 = &x;  
p2 = 200;
```

d) `int x = 50;  
int* p1 = &x;  
int** p2 = &x;`

e) `int* p = &x;  
x = 99;  
(*p)++;`

f) `int x = 50;  
int* p = &x;  
int** q = &p;  
**q = *p + x;`

g) `int x = 50;  
int* p = &x;  
x = &p + x;`

h) `double x = 50;  
int* p = &x;  
(*p)++;`

#### 4.2 Which of the following code fragments compile?

a) `int x = 100;  
const int* p = &x;  
(*p)--;`

b) `int x = 100;  
int* const p = &x;  
*p--;`

c) `int x = 100;  
const int* const p = &x;  
x = (*p)--;`

d) `int x = 100;  
const int** p = &x;  
x = *p;`

#### 4.3 What do the following code fragments do?

a) `int x = 100;  
int* p = &x;  
*p = x++;`

b) `int x = 100;  
int* p = &x;`

```
    x = *p++;  
c)  int x = 100;  
    int* p = &x;  
    *p++;
```

---

## If Statements and Loops

5.1 Convert the following code to use the conditional expression operator:

```
int x = 50, y;  
  
if(x > 100)  
    y = 200;  
else  
    y = 300;
```

5.2 Convert the following code to use if statements:

```
int a = 5, b = 6, y = 2;  
int x;  
x = (y > (a > b) ? 1 : 2) ? 3 : 4;
```

- 5.3 Write a program that prints out the square, cubes and fourth power of the first 20 integers. Use printf to lay out the results in neat columns.
- 5.4 Write a program that prints out the first 20 Fibonacci numbers.
- 5.5 Write a program that calculates the ratio of successive pairs of the first Fibonacci numbers.
- 

## Functions

- 6.1 Write a function that calculates the area of a rectangle. Decide how many input parameters your function needs. The area should be returned from the function. Write a test program that calls your function with different sets of test data.
- 6.2 Convert the last 3 examples of the last set of exercises to work with functions. Write tests programs for each function.
- 6.3 Write a function that rotates the values of 3 variables. For example:

```
int x = 100;  
int y = 200;  
int z = 300;  
Rotate( ... );  
// x is now 200  
// y is now 300
```

```
// z is now 100
```

---

## Arrays

- 7.1 Write a function that takes a string and capitalises the first character of the string and ensures the remaining characters are converted to lower case. Use the following test data:

```
UpperFirst("test1");  
UpperFirst("mIXedCaSe");  
UpperFirst("UPPER");  
UpperFirst("lower");  
UpperFirst("oPPosITE");
```

- 7.2 Write a function that takes an *int* array as a parameter and doubles the value of each element of the array.
- 7.3 Write a function that takes two *int* arrays (same size) as parameters and adds the arrays together, element by element. Print out the array as part of the function.
- 7.4 Modify the previous example to return the sum of the two input arrays instead of printing the result.