Q1

- *(Calculating the Sum of Even Integers)* Write a program that calculates and prints the sum of the even integers from 2 to 30.

Q2

- *(Calculating the Product of Odd Integers)* Write a program that calculates and prints the product of the odd integers from 1 to 15.

Q3

- *(Triangle Printing Program)* Write a program that prints the following patterns separately,one below the other. Use for loops to generate the patterns. All asterisks (*) should be printed by a single printf statement of the form printf("*"); (this causes the asterisks to print side by side).

- [*Hint:* The last two patterns require that each line begin with an appropriate number of blanks.]

Q3

```
(A)              (B)              (C)              (D)
*                *********        *********                *
**               ********         ********                **
***              *******          *******                ***
****             ******           ******                ****
*****            *****            *****                *****
******           ****              ****               ******
*******          ***                ***              *******
********         **                  **             ********
*********        *                    *            *********
```

Q4

- *(Calculating Sales)* An online retailer sells five different products whose retail prices are shown in the following table:

Product number	Retail price
1	$ 2.98
2	$ 4.50
3	$ 9.98
4	$ 4.49
5	$ 6.87

Q4

Write a program that reads a series of pairs of numbers as follows:

a) Product number

b) Quantity sold for one day

Your program should use a switch statement to help determine the retail price for each product.

Your program should calculate and display the total retail value of all products sold last week.

Q5

(Table of Decimal, Binary, Octal and Hexadecimal Equivalents) Write a program that

prints a table of the binary, octal and hexadecimal equivalents of the decimal numbers in the range 1 through 255

Q6

- *(Multiples)* Write a function multiple that determines for a pair of integers whether the second integer is a multiple of the first. The function should take two integer arguments and return 1 (true) if the second is a multiple of the first, and 0 (false) otherwise

Q7

- *(Parking Charges)* Write a function that displays a solid square of any character ex : # whose side is specified in integer parameter side. For example, if side is 5, the function displays:

```
#####
#####
#####
#####
#####
```

Q8

(Temperature Conversions) Implement the following integer functions:

a) Function celsius returns the Celsius equivalent of a Fahrenheit temperature.

b) Function fahrenheit returns the Fahrenheit equivalent of a Celsius temperature.

c) Use these functions to write a program that prints charts showing the Fahrenheit equivalents of all Celsius temperatures from 0 to 100 degrees, and the Celsius equivalents of all Fahrenheit temperatures from 32 to 212 degrees. Print the outputs in a neat tabular format that minimizes the number of lines of output while remaining readable.

Q9

(Perfect Numbers) An integer number is said to be a *perfect number* if its factors, including 1 (but not the number itself), sum to the number. For example, 6 is a perfect number because 6 = 1 + 2 + 3. Write a function perfect that determines if parameter number is a perfect number.
Use this function in a program that determines and prints all the perfect numbers between 1 and 1000.

Print the factors of each perfect number to confirm that the number is indeed perfect.

Q10

(Greatest Common Divisor) The *greatest common divisor* (*GCD*) of two integers is the largest integer that evenly divides each of the two numbers. Write function gcd that returns the greatest common divisor of two integers.

Q11

- Write a function qualityPoints that inputs a student's average and returns 4 if a student's average is 90–100, 3 if the average is 80–89, 2 if the average is 70–79, 1 if the average is 60–69, and 0 if the average is lower than 60.

Q12

- Given a binary array nums, return *the maximum number of consecutive* 1*'s in the array*.