

1. Write a program to swap two integers
2. Write a program to swap two integers without using a temporary variable
3. Write a program to swap two integers without using a temporary variable and without involving arithmetic operations that may cause overflow.
4. Write a program to find whether the given number is Odd or Even
5. Write a program to find whether the given year is Leap Year or not
6. Write a program to find the greatest and smallest of three numbers.
7. Write a program to check whether a given integer is an Armstrong Number or Not.
8. Write a program to check whether a given integer is Palindrome or Not.
9. Write a program to find the Sum of the digits of a given number
10. Write a program to print the numbers that are divisible by 7 between 1 and N.
11. Write a program to print the sum of squares of all numbers between 1 and N.
12. Write a program to find if a given number is prime or not?
13. Write a program called **CheckPassFail** which prints "PASS" if the int variable "mark" is more than or equal to 50; or prints "FAIL" otherwise. The program shall always print "DONE" before exiting.
14. Write a program called **CheckOddEven** which prints "Odd Number" if the int variable "number" is odd, or "Even Number" otherwise. The program shall always print "BYE!" before exiting.
15. Write a program called **PrintNumberInWord** which prints "ONE", "TWO", ..., "NINE", "OTHER" if the int variable "number" is 1, 2, ..., 9, or other, respectively. Use (a) a "nested-if" statement; (b) a "switch-case" statement.
16. Write a program called **PrintDayInWord** which prints "Sunday", "Monday", ... "Saturday" if the int variable "day" is 0, 1, ..., 6, respectively. Otherwise, it shall print "Not a valid day". Use (a) a "nested-if" statement; (b) a "switch-case" statement.
17. Write a program called **SumAndAverage** to produce the sum of 1, 2, 3, ..., to 100. Also compute and display the average. The output shall look like:

The sum is 5050

The average is 50.5

18. Modify the above program to use a "while-do" loop instead of "for" loop.
19. Modify the above program to use a "do-while" loop.
20. Modify the above program to sum from 111 to 8899, and compute the average. Introduce an int variable called count to count the numbers in the specified range.
21. Modify the program to sum only the *odd* numbers from 1 to 100, and compute the average. (HINTS: n is an odd number if $n \% 2$ is not 0.)
22. Modify the program to sum those numbers from 1 to 100 that is divisible by 7, and compute the average.
23. Modify the program to find the "sum of the squares" of all the numbers from 1 to 100, i.e. $1^2 + 2^2 + 3^2 + \dots + 100^2$.
24. Write a program called **Product1ToN** to compute the product of integers 1 to 10 (i.e., $1 \times 2 \times 3 \times \dots \times 10$). Try computing the product from 1 to 11, 1 to 12, 1 to 13 and 1 to 14. Write down the product obtained and explain the results.

Hints: Declare an int variable called product (to accumulate the product) and initialize to 1.

Try: Compute the product from 1 to 11, 1 to 12, 1 to 13 and 1 to 14. Write down the product obtained and decide if the results are correct.

Try: Repeat the above, but use long to store the product. Compare the products obtained.

Hints: Product of 1 to 13 (=6227020800) is outside the range of int [-2147483648, 2147483647], but within the range of long. Take note that computer programs may not produce the correct answer even though everything seems correct!
25. Write a program called **HarmonicSum** to compute the sum of a harmonic series, as shown below, where $n=50000$. The program shall compute the sum from *left-to-right* as well as from the *right-to-left*. Obtain the difference between these two sums and explain the difference. Which sum is more accurate?

Hints: Beware that int/int will return int. i.e. 1 / 2 will give 0 as output.

$$Harmonic(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

26. Write a program called **ComputePI** to compute the value of π , using the following series expansion. Termination condition is at 10000000. Is this series suitable for computing π ? Compare with the actual value of π and tell your observation.

$$\pi = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \dots \right)$$

Hints: Add to sum if the denominator modulus 4 is 1, and subtract from sum if it is 3. Also try 10000, 100000, 1000000 as termination conditions and check the values obtained.

27. Write a program called **CozaLozaWoza** which prints the numbers 1 to 110, 11 numbers per line. The program shall print "Coza" in place of the numbers which are multiples of 3, "Loza" for multiples of 5, "Woza" for multiples of 7, "CozaLoza" for multiples of 3 and 5, and so on. The output shall look like:

```
1 2 Coza 4 Loza Coza Woza 8 Coza Loza 11
Coza 13 Woza CozaLoza 16 17 Coza 19 Loza CozaWoza 22
23 Coza Loza 26 Coza Woza 29 CozaLoza 31 32 Coza .....
```

28. Write a program called **Fibonacci** to display the first 20 Fibonacci numbers $F(n)$, where $F(n)=F(n-1)+F(n-2)$ and $F(1)=F(2)=1$. Also compute their average. The output shall look like:

The first 20 Fibonacci numbers are: 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765

The average is 885.5

29. Tribonacci numbers are a sequence of numbers $T(n)$ similar to Fibonacci numbers, except that a number is formed by adding the three previous numbers, i.e., $T(n)=T(n-1)+T(n-2)+T(n-3)$, $T(1)=T(2)=1$, and $T(3)=2$. Write a program called **Tribonacci** to produce the first twenty Tribonacci numbers.
30. Write a program called **ExtractDigits** to extract each digit from an int, in the reverse order. For example, if the int is 15423, the output shall be "3 2 4 5 1", with a space separating the digits.

Hints: Use $n \% 10$ to extract the least-significant digit; and $n = n / 10$ to discard the least-significant digit.

31. Write a program called **SquareBoard** that displays the following $n \times n$ ($n=5$) pattern using two nested for-loops. **Hints:**

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

Your program should use only two output statements, one EACH of the followings:

`Console.Write("# ");` // print # and a space, without newline

`Console.WriteLine();` // print a newline

32. Write a program called **CheckerBoard** that displays the following $n \times n$ ($n=7$) checkerboard pattern using two nested for-loops. **Hints:**

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

Your program should use only three output statements, one EACH of the followings:

`Console.WriteLine("# ");` // print # and a space, without newline

`Console.Write(" ");` // print a space, without newline

`Console.WriteLine();` // print a newline

33. Write a program called `Tables` to produce the multiplication table of 1 to 9 as shown using two nested for-loops:

```
* | 1 2 3 4 5 6 7 8 9
-----
1 | 1 2 3 4 5 6 7 8 9
2 | 2 4 6 8 10 12 14 16 18
3 | 3 6 9 12 15 18 21 24 27
4 | 4 8 12 16 20 24 28 32 36
5 | 5 10 15 20 25 30 35 40 45
6 | 6 12 18 24 30 36 42 48 54
7 | 7 14 21 28 35 42 49 56 63
8 | 8 16 24 32 40 48 56 64 72
9 | 9 18 27 36 45 54 63 72 81
```

34. Modify the above program to print the multiplication table of 1 to 12

35. Print each of the followings patterns using nested loops.

```
# # # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
(a) (b) (c) (d)
```

```
# # # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
(e) (f) (g) (h) (i)
```

Hints: On the main diagonal, $row = col$. On the opposite diagonal, $row + col = size + 1$, where row and col begin from 1.

36. Write a program called `KeyboardScanner` to prompt user for an int, a double, and a String. The output shall look like (the inputs are shown in bold):

Enter an integer: 12

Enter a floating point number: 33.44

Enter your name: Anand

Output: Hi! Anand, the sum of 12 and 33.44 is 45.44

37. Write a program called FileScanner to read an int, a double, and a String from a text file called "in.txt", and produce the following output:

The integer read is 12

The floating point number read is 33.44

The String read is "Anand"

Output: Hi! Anand, the sum of 12 and 33.44 is 45.44

Hints: You need to create a text file called "in.txt" with the following contents: (one per line)

12

33.44

Anand

38. Write a program called CircleComputation, which prompts user for a radius (in double) and compute the area and circumference of the circle rounded to 2 decimal places. The output shall look like:

Enter the radius: 1.2

The area is: 4.52

The circumference is: 7.53

39. Modify the above program. The program shall repeatedly prompt for the radius, until the user enters -1.
40. Write a program called ReverseString, which prompts user for a String, and prints the reverse of the String. The output shall look like:

Enter a String: abcdef

The reverse of the String "abcdef" is "fedcba".

41. Write a program called GradesAverage, which prompts user for the number of students, reads it from the keyboard, and saves it in an int variable called numStudents. It then prompts user for the grades of each of the students and saves them in an int array called grades. Your program shall check that the grade is between 0 and 100. A sample session is as follow:

Enter the number of students: 3

Enter the grade for student 1: 55

Enter the grade for student 2: 108

Invalid grade, try again...

Enter the grade for student 2: 56

Enter the grade for student 3: 57

The average is: 56.0

42. Write a method called printArray(), which takes an int array and print its contents in the form of {a1, a2, ..., an}. Take note that there is no comma after the last element.
43. Write a boolean method called contains(), which takes an array of integers and an int; and returns true if the array contains the given int.
44. Write a method called search(), which takes an array of int and an int; and returns the array index if the array contains the given int; or -1 otherwise.

45. Write a boolean method called `equals()`, which takes two arrays of `int` and returns `true` if the two arrays are exactly the same (i.e., same length and same contents).
46. Write a boolean method called `copyOf()`, which accepts an `int` Array and returns a copy of the given array.
47. Write a method called `reverse()`, which takes an array of `int` and reverse its contents. For example, the reverse of `{1, 2, 3, 4}` is `{4, 3, 2, 1}`.
48. Write a method called `swap()`, which takes two arrays of `int` and swap their contents if they have the same length. It shall return `true` if the contents are successfully swapped and print them.
49. Write a program called `GradesStatistics`, which reads in `n` grades (of `int` between 0 and 100, inclusive) and displays the average, minimum, maximum, median and standard deviation. Display the floating-point values upto 2 decimal places. Your output shall look like:

Enter the number of students: 4

Enter the grade for student 1: 50

Enter the grade for student 2: 51

Enter the grade for student 3: 56

Enter the grade for student 4: 53

{50,51,56,53}

The average is: 52.50

The median is: 52.00

The minimum is: 50

The maximum is: 56

The standard deviation is: 2.29

50. Write a method to print each of the followings patterns using nested loops in a class called **PrintPatterns**. The program shall prompt user for the size of the pattern.

# # # # # # # # # #	#	#
# # # # # # # # #	# # #	# # #
# # # # # # #	# # # # #	# # # # #
# # # # #	# # # # # # #	# # # # # # #
# # #	# # # # # # # # #	# # # # # # # # #
#	# # # # # # # # # # #	# # # # # # # # # # #
(a)	(b)	# # # # # # # # #
		# # # # # # #
		# # # # #
		# # #
		#
		(c)
1	1 2 3 4 5 6 7 8	1
1 2	1 2 3 4 5 6 7	2 1
1 2 3	1 2 3 4 5 6	3 2 1
		8 7 6 5 4 3 2 1
		7 6 5 4 3 2 1
		6 5 4 3 2 1

1 2 3 4	1 2 3 4 5	4 3 2 1	5 4 3 2 1
1 2 3 4 5	1 2 3 4	5 4 3 2 1	4 3 2 1
1 2 3 4 5 6	1 2 3	6 5 4 3 2 1	3 2 1
1 2 3 4 5 6 7	1 2	7 6 5 4 3 2 1	2 1
1 2 3 4 5 6 7 8	1	8 7 6 5 4 3 2 1	1
(d)	(e)	(f)	(g)

1	1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
1 2 1	1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 2 1	1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 3 2 1	1 2 3 4 5 4 3 2 1
1 2 3 4 5 4 3 2 1	1 2 3 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1	1 2 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1	1 2 1
1 2 3 4 5 6 7 8 7 6 5 4 3 2 1	1
(h)	(i)

1	1	1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
1 2	2 1	1 2 3 4 5 6 7 7 6 5 4 3 2 1
1 2 3	3 2 1	1 2 3 4 5 6 6 5 4 3 2 1
1 2 3 4	4 3 2 1	1 2 3 4 5 5 4 3 2 1
1 2 3 4 5	5 4 3 2 1	1 2 3 4 4 3 2 1
1 2 3 4 5 6	6 5 4 3 2 1	1 2 3 3 2 1
1 2 3 4 5 6 7	7 6 5 4 3 2 1	1 2 2 1
1 2 3 4 5 6 7 8 7 6 5 4 3 2 1	1	1
(j)	(k)	

```

1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
6 7 8 9 0 1 0 9 8 7 6
7 8 9 0 1 2 3 2 1 0 9 8 7
8 9 0 1 2 3 4 5 4 3 2 1 0 9 8
(1)

```

51. Write a method to print each of the following patterns using nested-loops in a class called **PrintTriangles**. The program shall prompt user for the numRows.

```

      1
    1 2 1
  1 2 4 2 1
1 2 4 8 4 2 1
  1 2 4 8 16 8 4 2 1
1 2 4 8 16 32 16 8 4 2 1
  1 2 4 8 16 32 64 32 16 8 4 2 1
1 2 4 8 16 32 64 128 64 32 16 8 4 2 1

```

(a) PowerOf2Triangle

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

```

(b) PascalTriangle1

```

      1
    1 1
  1 2 1
    1 3 3 1
  1 4 6 4 1
    1 5 10 10 5 1
  1 6 15 20 15 6 1

```

(c) PascalTriangle2

52. Write a program called BinarySearch that accepts an sorted int array and an integer as input and returns the index position of integer in the array, if found and -1 otherwise.