

### 30 Small Programming Practices

1. Write a for loop to calculate the sum of all numbers from 1 to 1000.
2. Write a for loop to calculate the sum of all odd numbers from 1 to 1000.
3. Write a while-loop to read integer values from standard input until the user enters -1.
4. Write a loop to find out the minimum value stored in an array of int.

e.g.,  
 const int MAX\_CAP=100;  
 int nums[MAX\_CAP]={23,445, 12, 4, -34, 199, 24};  
 int len=7; //number of int stores in array noms

5. Write a function that performs arithmetic operations on two integers and return the result.

```
/* Return the result of addition or subtraction or multiplication or division or power
   op is '+', '-', '*', '/', '^'
   return x+y, x-y, x*y, x/y, x^y depends on the op.
   x^y means x raised to y-th power, i.e., x*x*x...x (for y times)
*/
```

```
int IntegerOperation (char op, int x, int y)
```

6. Write a function that calculates the n-th term of Fibonacci sequence, where the first term is 0, the second term is 1, and the third and subsequent term is the sum of the two previous terms.

```
int Fibonacci (int n)
```

7. Write a function that solves quadratic equation givens by three numbers a, b, and c.

```
// if the equation has no real roots, return false
// otherwise, set r1 and r2, and return true
bool SolveQuadraticEquation (double a, double b, double c, double & r1, double & r2)
// Note that function sqrt() can be used to calculate the square root of a non-negative number
```

**[For the following questions on arrays, you should also be able to solve them if a terminator value is used to indicate the length of the array, rather than using a separate len parameter. ]**

8. (display an array in reversed order, i.e., display last element first, ... display first element last)

```
void DisplayInReversed (int a[ ], int len)
```

8. (linear search) search for a value in an array of int

```
// search for num in the array a, return the index of num (if it's found); if not found, return -1
```

```
// the length of array a is given by len
```

```
int Search (int a[ ], int len, int num)
```

9. (bubble up largest number)

```
// Bubble up largest number to the last position in an array
```

```
// compare adjacent numbers in the array a (from 0th position to len-1), and swap them if the one in front is larger.
```

```
void BubbleUp (int a[ ], int len)
```

10. (find index of smallest/largest number in a sub-array)

```
// Return the index of smallest value in the sub-array a[start...end], i.e.,
```

```
// smallest among a[start], a[start+1], ... a[end]
```

```
int IndexofSmallest (int a[ ], int start, int end)
```

11. (Selection Sort) Making use of function in 10, write a function below to rearrange numbers in array a (with given length) in ascending order.

```
/* Hint: find the index of smallest value among a[0...len-1], swap it with a[0]
```

```
find the index of smallest value among a[1...len-1], swap it with a[1]
```

```
...
```

```
find the index of smallest value among a[len-2...len-1], swap it with a[len-2]
```

```
*/
```

```
int SelectionSort (int a[ ], int len)
```

12. (reverse an array)

```
// swap a[0] with a[len-1], a[1] with a[len-2]
```

```
// Hint: you can use two indices, front and end, and set them to point to first and last elements,
```

```
// and move them to meet in the middle
```

```
void ReverseArray (int a[ ], int len)
```

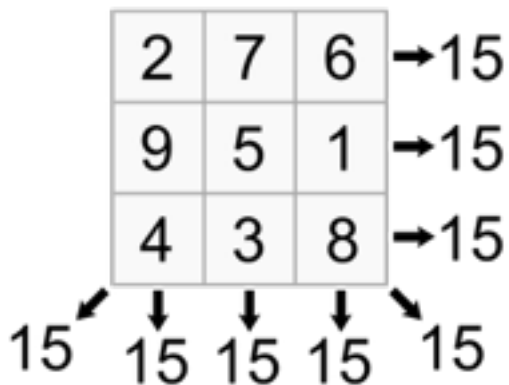
13. (Factorial: recursive implementation)

```
// n>=0, return n! (i.e., n*(n-1)*(n-2) ... *2*1
```

```
int factorial (int n)
```

14. (Compare two C strings, i.e., char array terminated using \0)  
// return true if str1 and str2 stores different strings  
// return false if they store identical strings  
bool strcmp (char str1[ ], char str2[ ])

15. (2D array) Declare a 2D array to represent the setup of a magic square (as illustrated below). Write a function that test whether the property of magic square is satisfied for a given 2D array. The sum of the three numbers of each row (each column, and the two diagonal lines) should be same, and all numbers are distinct.



16. (Read Inputs) Write a few lines of code to read a date given in the format of DD/MM/YYYY. and set the three integers to store the year, month and day. If the character separating the different parts of the date is not /, then report errors.

```
int day, month, year;
```