Lab 03 - Tracing & Functions

Direction: Submit typed work in the Labs directory of your github repository and/or as an attachment on Google classroom under the accurate Lab03 assessment. All submissions should have their appropiate extensions.

Part A: In class

Your objective is to construct the trace table (or list) for the caller `F(196)` where `F()` is defined as follows

```
bool F(int n)
{
  int o, t, h, r;
  o = n % 10;
  t = n / 10 % 10;
  h = n / 100 % 10;
  r = 100 * K(h,3);
  r += 10 * K(t,2);
  r += K(o,1);
  n = 100 * h + 10 * t + o;
  bool s = (r - n > 0);
  return !s;
}
```

where `K()` is defined as follows

```
int K(int& n,int e)
{
  n = (e * n + e) % 10;
  return ((n * n + (10 - e)) % 10);
}
```

Warning: Do not write any trace table for K(). You will receive a 0 if you do so.

Part B: Take home

Your objective is to write a complete program that defines and test the following functions. Furthermore, construct a trace table (or list) for each caller provided.

- ☐ A void function named `Swaps()` that takes four double reference parameters. It swaps the values of the parameters so that every parameter has a different value than its original value and all values are maintained. Trace `Swaps(a,b,c,d)` where `a`, `b`, `c` and `d` are double variables that equal 12.0, 7.0 ,32.0 and 17.0 respectively.
- ☐ An int function named `ConsecutiveSum()` that takes an int parameter. It returns the sum of five consecutive integers starting with the square of the parameter minus 4. Trace `ConsecutiveSum(9)`.