Date / Time:	Friday, March 17 2017 at 17:30
Duration:	2 hours
Attention:	**ANY TYPE OF COMMUNICATION IS STRICTLY PROHIBITED**
	Write down your section, name and ID# at the top of all used pages

Participation:

Problem 1: Consider below a C++ function *float kahan(float num1, float num2, float& compensation)* that implements the **Kahan Summation Algorithm** for high-precision compensated summation of two float arguments *float num1* and *float num2*:

```
float kahan(float num1, float num2, float &compensation)
{
        float result;
        num2 -= compensation;
        result = num1 + num2;
        compensation = (result - num1) - num2;
        return result;
}
```

Using this function, write a C++ function *float pi(int n)* that computes the value π by the following formula:

$$\pi = 16\sum_{k=0}^{n}\frac{(-1)^k}{(2k+1)5^{2k+1}} - 4\sum_{k=0}^{n}\frac{(-1)^k}{(2k+1)239^{2k+1}} = \left(\frac{16}{1*5} - \frac{4}{1*239}\right) - \left(\frac{16}{3*5^3} - \frac{4}{3*239^3}\right) + \left(\frac{16}{5*5^5} - \frac{4}{5*239^5}\right) - \cdots$$

The initial value of *float compensation* is *0.0*.

float pi (int n) {   ~~int i=0~~

~~int length = n~~

~~for~~ ~~fint~~ ~~int i=0; i<n; i++) }~~

~~p~~ ~~Math.Power (2, i) / (2i+1)/ Math.Bower (3, 2i+1))~~

~~float~~ ~~float~~ ~~f=0;~~ *for(int i=0; i<n; i++) {*

float num1 = 16(Math.Power(-1, i)/(2i+1)/ Math.Power(5,(2i+1));

float num2 =   4(Math.Power(-1, i)/(2i+1). ~~239~~ *Math.Power(239,(2k+1));*

float pi =0;

pi = Kahan (num 1; num 2; ~~comp~~ 0)i;   }

return pi; }

3

Use the backside, if needed