Schema lui Horner

```cpp
#include<iostream>
using namespace std;
const int dim = 1000;


double sumprod(double a[dim], double x) {
	double s = 0, p = 1;
	for (int k = 0; k < dim; k++) {
		s += a[k] * p;
		p *= x;
	}
	return s;
}

double horner(double a[dim], double x) {
	double val = 0;
	for (int k = dim - 1; k >= 0; k--) {
		val = x * val + a[k];
	}
	return val;
}

double si(double x) {
	double s = 0, p = x;
	for (int k = 1; k < dim; k += 2) {
		s += p / k;
		p *= -x * x / ((k + 1) * (k + 2));
	}
	return s;
}

int main() {
	double a[dim] = {};  //a=[0, 0, 0, ...]
	double p = 1;
	//coeficientii pentru sinus integral
	for (int k = 1; k < dim; k += 2) {
		a[k] = p / k;
		p *= -1.0 / ((k + 1) * (k + 2));
	}
	double x = 10;
	cout.precision(12);
	cout << "sumprod=" << sumprod(a, x) << endl;
	cout << "sHorner=" << horner(a, x) << endl;
	cout << "sinInt =" << si(x) << endl;		return 0;
}

/*
sumprod=-nan(ind)
sHorner=1.65834759422
sinInt =1.65834759422

Press any key to close this window . . .
*/
```