

Schema lui Horner

```
#include<iostream>
using namespace std;

const int dim=100;
double horner(double p[dim],double x){
    double val=0;
    for(int i=dim-1;i>=0;i--){
        val=x*val+p[i];
    }
    return val;
}

double naiv(double p[dim],double x){
    double val=0;
    double t=1;
    for(int i=0;i<dim;i++){
        val+=p[i]*t;
        t*=x;
    }
    return val;
}

int main(void){
    double p[dim]={1,2,3};
    cout<<horner(p,10)<<endl;
    cout<<naiv(p,10)<<endl;

    return 0;
}
```