**Bài thực hành**

**Bài thực hành 1: Cài đặt và tìm hiểu luật Horner cho đa thức**

Giả sử chúng ta cần tìm giá trị của một đa thức tổng quát dạng:

P(x) = a0xn + a1xn-1 + a2xn-2 +....+ an (1)

tại một trị số x = k nào đó. Trong (1) các hệ số ai là các số thực đã cho. Chúng ta viết lại (1) theo thuật toán Horner dưới dạng:

P(k) = (...((a0k + a1)k+ a2k)+...+ an -1 )k + an

P0 = a0

P1 = P0k + a1

P2 = P1k + a2

P3 = P2k + a3

..................

P(x) = Pn = Pn-1k + an

Tổng quát ta có:

Pi = Pi-1x + ai với i 1

**Cài đặt**

Hàm Horner có ba tham số đầu vào:

- heso[]: mảng chứa các hệ số của đa thức (trong đó, phần tử đầu tiên là hệ số của số hạng bậc cao nhất)

- n: bậc của đa thức

- c: giá trị biến số muốn tính giá trị đa thức tại x

**Chương trình**

#include<iostream>

#include<iomanip>

#include<stdio.h>

#include<conio.h>

#include<math.h>

#define max 20

using namespace std;

float horner (float heso[], int n, float c) {

int i;

float p = heso[0];

for(int i = 1; i <=n; i++)

p = p\*c + heso[i];

return p;

}

int main() {

int i, n;

float x, heso[max];

cout << "Nhap vao bac n = ";

cin >> n;

cout << "Nhap vao he so da thuc: " <<endl;

for(int i = 0; i <= n ; i++) {

cout << "heso["<<i<<"] = ";

cin >> heso[i];

}

cout << "Nhap vao gia tri c can tinh: ";

cin >> x;

cout << "Gia tri cua da thuc can tim la: ";

cout << fixed << setprecision (3) << horner(heso, n, x);

return 0;

}

**Kết quả**

