

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC



Báo cáo môn học: Kỹ thuật lập trình

**Đề tài số 04:**

Xây dựng đa thức nội suy ứng với một bảng số cho trước theo  
công thức Newton

\*\*\*

**Giảng viên hướng dẫn: TS. Nguyễn Thị Thanh Huyền**

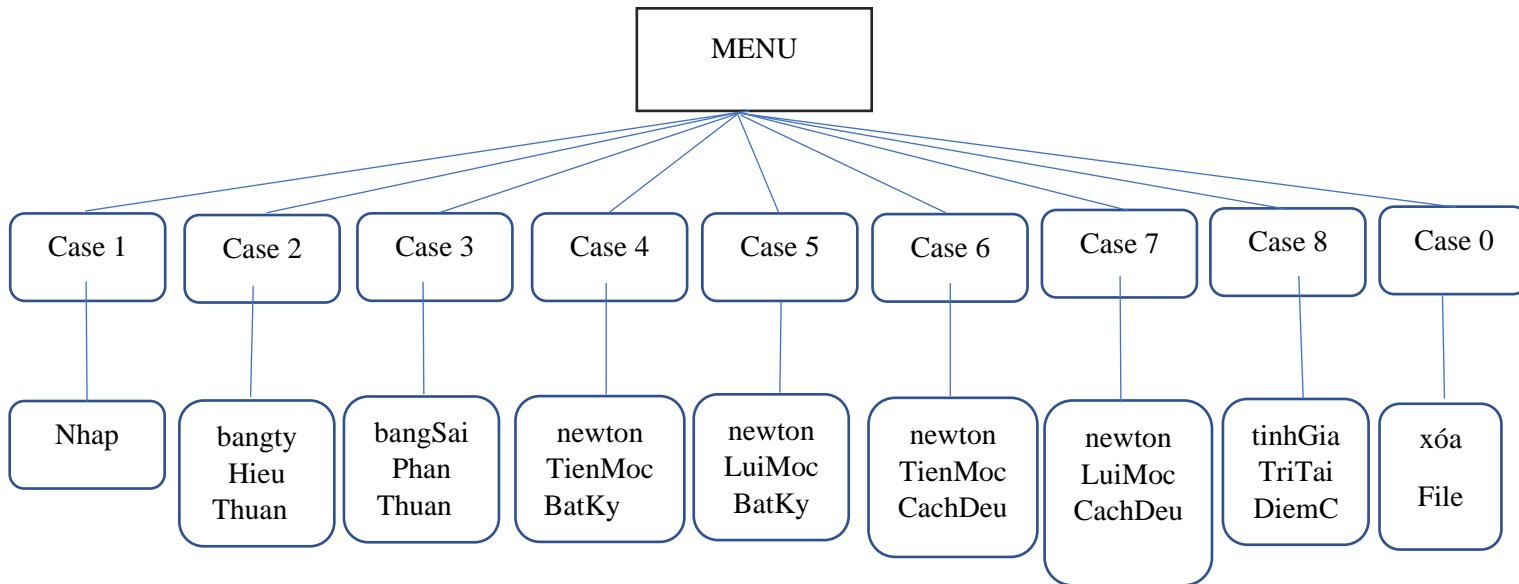
**Nhóm thực hiện: Nhóm 04**

1. Nguyễn Hữu Thuật	MSSV: 20185410
2. Lại Tiến Long	MSSV: 20185376

*Hà Nội, 2020*

## I.Thiết kế chương trình

### Bước 0: Tổng quan về Menu



### Bước 1: Mô tả MENU

- In các lựa chọn
- CASE 0: Xóa file đã tồn tại. Nếu không xóa các dữ liệu sẽ được ghi tiếp ở cuối file cũ.
- CASE 1: Nhập và kiểm tra các bộ giá trị
- CASE 2: Vẽ bảng tỷ hiệu thuận  
Kiểm tra xem người dùng đã nhập bộ giá trị chưa  
+ nếu chưa : thông báo yêu cầu nhập, trở về menu  
+ nếu đã nhập: Tiến hành vẽ bảng tỷ hiệu thuận => xuất file.
- CASE 3: Vẽ bảng sai phân thuận  
Kiểm tra xem người dùng đã nhập bộ giá trị chưa  
+ nếu chưa : thông báo yêu cầu nhập, trở về menu  
+ nếu đã nhập: Tiến hành kiểm tra độ lệch các mốc (h)  
$$h = |x[1] - x[0]|$$
$$\text{for } i = 1 \text{ to } n :$$
$$\text{if } |x[i] - x[i-1]| == h$$

Tiến hành vẽ bảng sai phân  
else: gửi thông báo không vẽ được, gợi ý chọn vẽ bảng tỷ hiệu. Trở về MENU
- CASE 4: Viết đa thức Newton tiến mốc bất kỳ  
Kiểm tra xem đã có bảng tỷ hiệu thuận  
+ nếu chưa : thông báo yêu cầu vẽ bảng, trở về menu  
+ nếu đã nhập: Tiến hành viết đa thức Newton tiến mốc bất kỳ

CASE 5: Viết đa thức Newton lùi mốc bất kỳ

Kiểm tra xem người dùng đã nhập bộ giá trị chưa

+ nếu chưa : thông báo yêu cầu nhập, trở về menu

+ nếu đã nhập:

- Vẽ bảng tỷ hiệu nghịch
- Tiến hành viết đa thức Newton lùi mốc bất kỳ

- CASE 6: Viết đa thức Newton tiến mốc cách đều

Kiểm tra xem đã có bảng sai phân thuận

+ nếu chưa : thông báo yêu cầu vẽ bảng, trở về menu

+ nếu đã nhập: Tiến hành viết đa thức Newton tiến mốc cách đều

- CASE 7: Viết đa thức Newton lùi mốc cách đều

Kiểm tra xem người dùng đã nhập bộ giá trị chưa

+ nếu chưa : thông báo yêu cầu nhập, trở về menu

+ nếu đã nhập:

- Tiến hành kiểm tra độ lệch các mốc (h)

$$h = |x[1] - x[0]|$$

for i = 1 to n:

$$\text{if } x[i] - x[i-1] == h$$

+ Tiến hành lập bảng sai phân nghịch

+ Tiến hành viết đa thức Newton lùi mốc bất kỳ

else: gửi thông báo không lập được, gợi ý chọn viết đa thức Newton lùi mốc bất kỳ. Trở về MENU

- CASE 8: Tính giá trị của đa thức tại điểm c nhập từ bàn phím

kiểm tra xem đã có đa thức Newton hay chưa:

- Nếu chưa: yêu cầu thực hiện lập đa thức Newton
- Nếu đã có:

\* Nhập c

\* Tiến hành chia các TH để đưa vào hàm tính (hoocne)

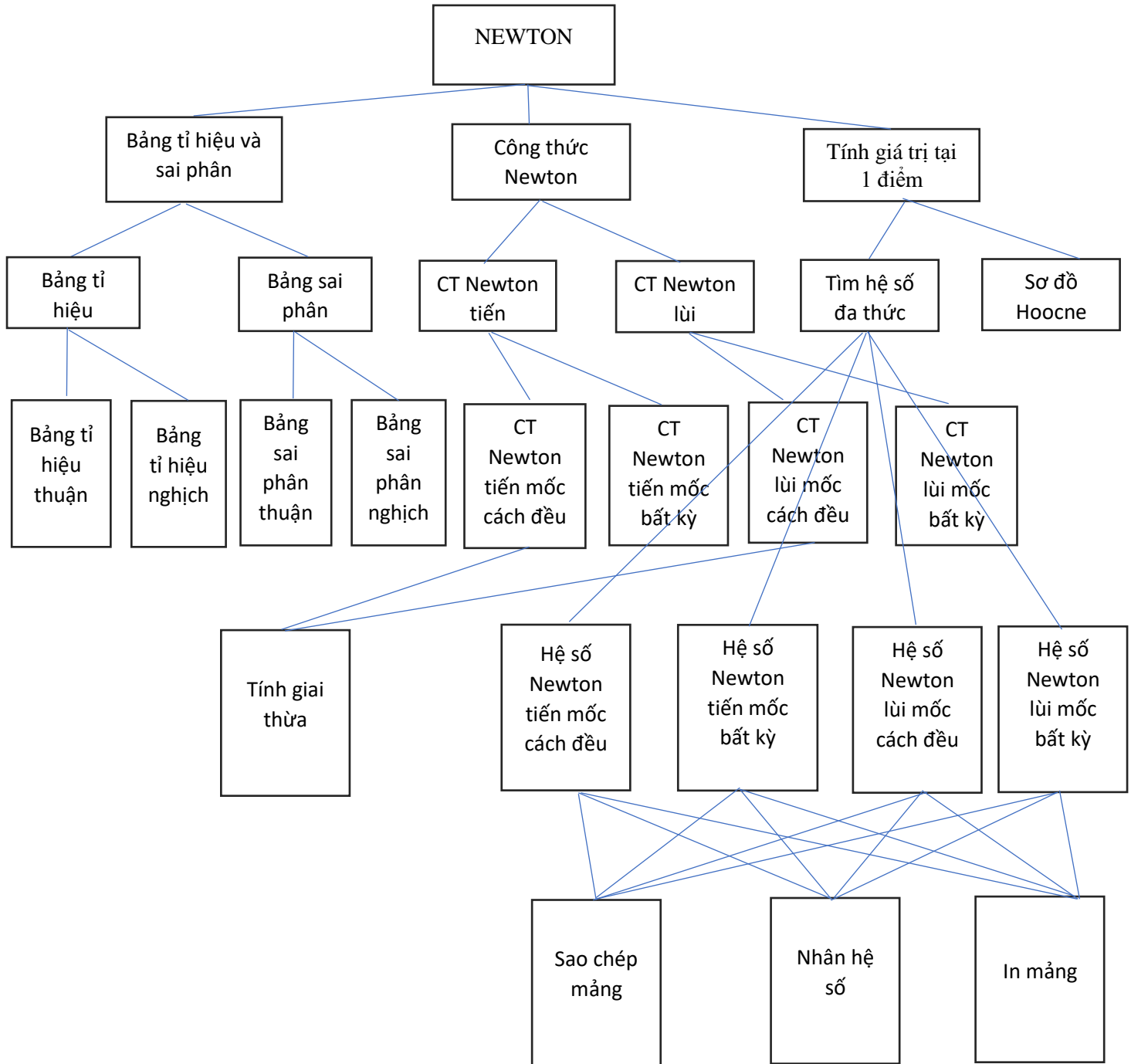
+ đa thức Newton tiến mốc bất kỳ => hệ số Newton tiến mốc bất kỳ => hoocne

+ đa thức Newton lùi mốc bất kỳ => hệ số Newton lùi mốc bất kỳ => hoocne

+ đa thức Newton tiến mốc cách đều => hệ số Newton tiến mốc cách đều => hoocne

+ đa thức Newton lùi mốc cách đều => hệ số Newton lùi mốc cách đều => hoocne

## Bước 2: Tổng quan các hàm trong chương trình



### **Bước 3: Tổng quan về ý tưởng**

- Dữ liệu vào: số các mốc (n) , các mốc điểm, điểm cần tính (c).
- Dữ liệu ra: giá trị đa thức tại điểm cần tính đó (P(c))
- Ý tưởng thuật toán:  
Lập được đa thức Newton và dựa vào sơ đồ Hoocne để tính giá trị biểu thức tại c.  
Hiển thị các bước tính toán ra File.

### **Bước 4: Function Newton**

- Dữ liệu vào: n, mảng x[] lưu mốc các điểm, mảng y[][] lưu ma trận trong quá trình tính giá trị các bảng, điểm cần tính (c).
- Dữ liệu ra: các bảng (tỷ hiệu, sai phân), các phương trình Newton( Newton tiến, Newton lùi), quá trình tính toán bằng lược đồ Hoocne. Hiển thị dưới dạng File.
- Phân loại dữ liệu đầu vào: Kiểm tra hiệu mốc các điểm để xem dữ liệu đầu vào phù hợp với dạng bảng và công thức nào.  
Kiểm tra c để đưa gợi ý cho người dùng sử dụng dạng Newton nào hiệu quả hơn.

### **Bước 5: Hàm nhập và hiển thị bảng mốc các điểm**

Function Nhập

- Dữ liệu vào: n, mảng x[], mảng y[][].
- Dữ liệu ra: bảng mốc các điểm.
- Ý tưởng: sử dụng mảng 1 chiều để lưu dãy mốc điểm x. Sử dụng mảng 2 chiều y để lưu các giá trị tương ứng với x và quá trình thao tác trong các bảng.

### **Bước 6: Lập bảng tỷ hiệu & sai phân**

- Dữ liệu vào: 2 mảng x[], y[][].
- Dữ liệu ra: 1 trong 4 bảng: bảng tỷ hiệu thuận, bảng tỷ hiệu nghịch, bảng sai phân thuận, bảng sai phân nghịch.
- Ý tưởng: dựa vào dữ liệu đầu vào phân 4 bảng trên thành 2 TH theo x[] => bảng tỷ hiệu hoặc bảng sai phân. Cho người dùng tự chọn loại bảng muốn hiển thị (thuật toán có gợi ý nếu đã nhập c).

### **Bước 7: Lập bảng sai phân thuận**

Function bangSaiPhanThuan

- Dữ liệu vào: 2 mảng x[], y[].
- Dữ liệu ra: bảng sai phân thuận.
- Sử dụng công thức hiệu hữu hạn tiến tổng quát :

$$\Delta^n y_k = \Delta^{n-1} y_{k+1} - \Delta^{n-1} y_k$$

- Áp dụng vào chương trình bằng cách chạy 2 vòng for lồng nhau:  
For i = 0 to n - 1:  
For j = 0 to n - i:  
y[j][i] = y[j + 1][i - 1] - y[j][i - 1]

### **Bước 8: Lập bảng sai phân nghịch**

Funtion bangSaiPhanNghich

- Dữ liệu vào: 2 mảng x[], y[][].
- Dữ liệu ra: bảng sai phân nghịch.
- Sử dụng công thức hiệu hữu hạn tiến tổng quát :

$$\Delta^n y_k = \Delta^{n-1} y_{k+1} - \Delta^{n-1} y_k$$

- Áp dụng vào chương trình bằng cách chạy 2 vòng for lồng nhau :  
For i = 1 to n - 1 :  
For j = n - 1 down to i - 1:  
y[j][i] = y[j][i - 1] - y[j - 1][i - 1]

### **Bước 9: Lập bảng tỷ hiệu thuận**

Funtion bangTyHieuThuan

- Dữ liệu vào: 2 mảng x[], y[][].
- Dữ liệu ra: bảng tỷ hiệu thuận.
- Sử dụng công thức tổng quát tính tỷ hiệu cấp n :

$$f[x_0; x_1; \dots; x_n] = \frac{f[x_1; \dots; x_n] - f[x_0; \dots; x_{n-1}]}{x_n - x_0}$$

- Áp dụng vào chương trình bằng cách chạy 2 vòng for lồng nhau :  
For i = 1 to n - 1 :  
For j = 0 to n - i:  
y[j][i] = (y[j + 1][i - 1] - y[j][i - 1]) / (x[i + j] - x[j])

### **Bước 10: Lập bảng tỷ hiệu nghịch**

Funtion bangTyHieuNghich

- Dữ liệu vào: 2 mảng x[], y[][].
- Dữ liệu ra: bảng tỷ hiệu nghịch.
- Sử dụng công thức tổng quát tính tỷ hiệu cấp n :

$$f[x_0; x_1; \dots; x_n] = \frac{f[x_1; \dots; x_n] - f[x_0; \dots; x_{n-1}]}{x_n - x_0}$$

- Áp dụng vào chương trình bằng cách chạy 2 vòng for lồng nhau :  
For i = 1 to n - 1 :  
For j = n down to n - i:  
y[j][i] = (y[j][i - 1] - y[j - 1][i - 1]) / (x[j] - x[j - i])

### **Bước 11: Hàm tính giai thừa để tính hệ số của đa thức Newton mốc cách đều**

Funtion giaiThua(n)

- Dữ liệu vào: n
- Dữ liệu ra: n!
- Sử dụng đệ quy:  
     If n == 0 or n == 1   return 1  
     Return n \* giaiThua(n-1)

## **Bước 12:** Viết đa thức Newton

- Dữ liệu vào: x[], y[], bảng sai phân hoặc tỷ hiệu
- Dữ liệu ra: Đa thức Newton
- Ý tưởng: từ bảng sai phân/ tỷ hiệu đã có bên trên chọn dạng biểu diễn đa thức Newton tương ứng.

## **Bước 13:** Hàm tính đa thức Newton tiến móc bất kỳ

Funtion newtonTienMocBatKy

- Dữ liệu vào: x[], y[], bảng tỷ hiệu thuận.
- Dữ liệu ra: Đa thức Newton tiến móc bất kỳ.
- Từ đa thức Newton tiến móc bất kỳ tổng quát:

$$P_n(x) = f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] \\ + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})f[x_0, x_1, \dots, x_n]$$

- Để biểu diễn đa thức, ta chia làm 2 phần: phần hệ số tự do và phần chứa x.  
     For i = 0 to n - 1:  
         If i == 0  
             Print y[0][0];  
         Else print y[0][i]  
         For j = 0 to i - 1 :  
             If x[j] < 0. Print (x + x[j])  
             Else if x[j] > 0. Print (x - x[j])  
             Else Print (x)  
         If i != (n-1)  
             If y[0][i+1] >= 0  
                 Print ( + )
- Lưu ý: đa thức nội suy Newton tiến xuất phát từ  $x_0$ , thường dùng để tính gần đúng  $f(c)$  với c gần  $x_0$ .

## **Bước 14:** Hàm tính đa thức Newton lùi móc bất kỳ

Funtion newtonLuiMocBatKy

- Dữ liệu vào: x[], y[], bảng tỷ hiệu thuận.
- Dữ liệu ra: Đa thức Newton lùi móc bất kỳ.
- Từ đa thức Newton lùi móc bất kỳ tổng quát:

$$P_n(x) = f(x_n) + (x - x_n)f[x_n, x_{n-1}] + (x - x_n)(x - x_{n-1})f[x_n, x_{n-1}, x_{n-2}] \\ + \dots + (x - x_n)(x - x_{n-1}) \dots (x - x_1)f[x_n, x_{n-1}, \dots, x_0]$$

- Để biểu diễn đa thức, ta chia làm 2 phần: phần hệ số tự do và phần chứa x.  
     For i = n down to 1:

```

If i == n
    Print y[n-1][0];
Else print y[n-1][n-i]

```

```

For j = n down to i + 1 :
    If x[j-1] < 0. Print (x + x[j-1])
    Else if x[j-1] > 0. Print (x - x[j-1])
    Else Print (x)
If i != 1
    If y[n-1][i-1] >= 0
        Print ( + )

```

- Lưu ý: đa thức nội suy Newton lùi xuất phát từ  $x_n$ , thường dùng để tính gần đúng  $f(c)$  với  $c$  gần  $x_n$ .

### **Bước 15: Hàm tính đa thức Newton tiến móc cách đều**

Funtion newtonTienMocCachDeu

- Dữ liệu vào: x[], y[[]], bảng tỷ hiệu thuận.
- Dữ liệu ra: Đa thức Newton tiến móc cách đều.
- Từ đa thức Newton tiến móc cách đều tổng quát:

$$\begin{aligned}
 P_n(x) = & f(x_0) + (x - x_0) \frac{\Delta y_0}{h} + (x - x_0)(x - x_1) \frac{\Delta^2 y_0}{h^2} + \dots + \\
 & + (x - x_0)(x - x_1) \dots (x - x_{n-1}) \frac{\Delta^n y_0}{n! h^n}
 \end{aligned}$$

- Để biểu diễn đa thức, ta chia làm 2 phần: phần hệ số tự do và phần chứa x.  
 For i = 0 to n - 1:  
 If i == 0  
 Print ((y[0][1]) / (giaiThua(i) \* pow(denta, i)) >= 0);  
 Else print (y[0][i] / (giaiThua(i) \* pow(denta, i))  
 For j = 0 to i - 1 :  
 If x[j] < 0. Print (x + x[j])  
 Else if x[j] > 0. Print (x - x[j])  
 Else Print (x)  
 If i != (n-1)  
 If y[0][i+1] >= 0 : Print ( + )

### **Bước 16: Hàm tính đa thức Newton lùi móc cách đều**

Funtion newtonLuiMocCachDeu

- Dữ liệu vào: x[], y[[]], bảng tỷ hiệu thuận.
- Dữ liệu ra: Đa thức Newton lùi móc cách đều.
- Từ đa thức Newton lùi móc cách đều tổng quát:



$$P_n(x) = y_n + \frac{t(t+1)}{2!} \Delta^2 y_n + \dots + \frac{t(t+1)(t+2)\dots(t+n-1)}{n!} \Delta^n y_n$$

- Để biểu diễn đa thức, ta chia làm 2 phần: phần hệ số tự do và phần chứa x.  
 For i = n down to 1:  
   If i == n  
     Print ((y[n - 1][1]) / (giaiThua(i) \* pow(denta, i)) >= 0);  
   Else print printf("%.2lf", (y[n - 1][n - i]) / (giaiThua(n - i) \* pow(denta, (n - i))))  
   For j = n down to i + 1 :  
     If :x[j - 1] < 0. Print (x + x[j - 1])  
     Else if: x[j - 1] > 0. Print (x - x[j - 1])  
     Else: Print (x)  
   If i != 1:  
     If y[n-1][i-1] >= 0:  
       Print ( + )

### **Bước 17: Tìm hệ số của đa thức tổng quát:**

$$P(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n \quad (a_0 \neq 0)$$

- Dữ liệu vào: n, x[], y[][].
- Dữ liệu ra: hệ số của đa thức ứng với từng bậc.
- Ý tưởng: Tận dụng ưu điểm “tính kế thừa” của đa thức Newton (mà đa thức Lagrange không có), tức tại mỗi đa thức con của đa thức P(x) sẽ có 2 đa thức nhân với nhau (trong đó có 1 đa thức luôn có dạng (x - a)) => Ý tưởng tương tự thuật toán đệ quy.

### **Bước 18: hàm sao chép mảng**

Dữ liệu vào: x[]

Dữ liệu ra: y[] = x[]

Funtion cpMang:

for i = 0 to n - 1 : y[i] = x[i]

### **Bước 19: hàm nhân hệ số vào mảng**

Dữ liệu vào: x[], p là số tự do

Dữ liệu ra: mảng x'[] với các phần tử của mảng thành x[i]\*p

Funtion nhanHeSo:

for i = 0 to n - 1: x[i] = x[i] \* p

### **Bước 20: hàm in mảng**

Dữ liệu vào: x[]

Dữ liệu ra: hiển thị x[] ra màn hình

Funtion inMangMotChieu:

for i = 0 to n - 1 :

print( x[i] ).

### **Bước 21:** Tìm hệ số của đa thức Newton tiến mốc bất kỳ

Funtion heSoNewtonTienMocBatKy

- Khởi tạo 2 mảng cấp phát động:

mangMu[][]: Ma trận lưu hệ số của từng đa thức con

tmpMangMu[]: Mảng lưu “hệ số” tạm thời của đa thức con trước để sử dụng cho đa thức con tiếp theo.

- Tìm hàng 0 của ma trận mangMu[i][j]

For i = 0 to n - 1:

If i == 0: mangMu[0][0] = y[0][0].

- Tìm hàng 1 của ma trận mangMu[i][j]

For i = 0 to n - 1:

If i == 1:

mangMu[1][0] = -x[0].

mangMu[1][1] = 1.

Sao chép mangMu[i] => tmpMangMu

mangMu[i] = mangMu[i] \* y[0][1]

- Trường hợp tổng quát

For i = 0 to n - 1:

For j = 0 to i:

For k = 0 to 1:

If k == 0 : mangMu[i][j + k] = mangMu[i][j + k] - tmpMangMu[j] \* x[i-1]

Else: mangMu[i][j + k] = mangMu[i][j + k] + tmpMangMu[j]

Sao chép mangMu[i] => tmpMangMu

mangMu[i] = mangMu[i] \* y[0][i]

### **Bước 22 :** Tìm hệ số của đa thức Newton tiến mốc cách đều

Funtion heSoNewtonTienMocCachDeu

- Khởi tạo 2 mảng cấp phát động:

mangMu[][]: Ma trận lưu hệ số của từng đa thức con

tmpMangMu[]: Mảng lưu “hệ số” tạm thời của đa thức con trước để sử dụng cho đa thức con tiếp theo.

denta =  $x[1] - x[0]$

- Tìm hàng 0 của ma trận mangMu[i][j]

For i = 0 to n - 1:

If i == 0: mangMu[0][0] = y[0][0].

- Tìm hàng 1 của ma trận mangMu[i][j]

For i = 0 to n - 1:

If i == 1:

mangMu[i][0] = -x[0].

mangMu[i][1] = 1.

Sao chép mangMu[i] => tmpMangMu

mangMu[i] = mangMu[i] \* (y[0][1] / denta)

- Trường hợp tổng quát

For i = 0 to n - 1:

For j = 0 to i:

For k = 0 to 1:

If k == 0: mangMu[i][j + k] = mangMu[i][j + k] - tmpMangMu[j] \* x[i-1]

Else: mangMu[i][j + k] = mangMu[i][j + k] + tmpMangMu[j] \* 1

Sao chép mangMu[i] => tmpMangMu

mangMu[i] = mangMu[i] \* (y[0][i] / (giaithua(i) \* pow(denta, i))

### **Bước 23 : Tìm hệ số của đa thức Newton lùi mốc bất kỳ**

Funtion heSoNewtonLuiMocBatKy

- Khởi tạo 2 mảng cấp phát động:

mangMu[][]: Ma trận lưu hệ số của từng đa thức con

tmpMangMu[]: Mảng lưu “hệ số” tạm thời của đa thức con trước để sử dụng cho đa thức con tiếp theo.

- Tìm hàng 0 của ma trận mangMu[i][j]

For i = n - 1 down to 0 :

If i == n - 1: mangMu[n - i - 1][0] = y[n - 1][0].

- Tìm hàng 1 của ma trận mangMu[i][j]

For i = n - 1 down to 0:

If i == n - 2:

$$\text{mangMu}[n - i - 1][0] = -x[n - 1].$$

$$\text{mangMu}[n - i - 1][1] = 1.$$

Sao chép mangMu[i] => tmpMangMu

$$\text{mangMu}[i] = \text{mangMu}[i] * y[n - 1][n - i - 1]$$

- Trường hợp tổng quát  
 For i = n - 1 down to 0:  
 For j = 0 to n - i - 1:  
 For k = 0 to 1:  
 If k == 0 :  $\text{mangMu}[n - i - 1][j + k] = \text{mangMu}[n - i - 1][j + k] - \text{tmpMangMu}[j] * x[I + 1]$   
 Else:  $\text{mangMu}[i][j + k] = \text{mangMu}[n - i - 1][j + k] + \text{tmpMangMu}[j] * 1$   
 Sao chép mangMu[i] => tmpMangMu  
 $\text{mangMu}[i] = \text{mangMu}[n - i - 1] * y[n - 1][n - i - 1])$

## **Bước 24 : Tìm hệ số của đa thức Newton lùi móc cách đều**

Funtion heSoNewtonLuiMocCachDeu

- Khởi tạo 2 mảng cấp phát động:  
 mangMu[][]: Ma trận lưu hệ số của từng đa thức con  
 tmpMangMu[]: Mảng lưu “hệ số” tạm thời của đa thức con trước để sử dụng cho đa thức con tiếp theo.
- Tìm hàng 0 của ma trận mangMu[i][j]  
 For i = n - 1 down to 0 :  
 If i == n - 1:  $\text{mangMu}[n - i - 1][0] = y[n - 1][0].$
- Tìm hàng 1 của ma trận mangMu[i][j]  
 For i = n - 1 down to 0:  
 If i == n - 2:  
 $\text{mangMu}[n - i - 1][0] = -x[n - 1].$   
 $\text{mangMu}[n - i - 1][1] = 1.$   
 Sao chép mangMu[i] => tmpMangMu  
 $\text{mangMu}[i] = \text{mangMu}[i] * (y[n - 1][n - i - 1] / \text{denta})$
- Trường hợp tổng quát  
 For i = n - 1 down to 0:  
 For j = 0 to n - i - 1:  
 For k = 0 to 1:  
 If k == 0 :  $\text{mangMu}[n - i - 1][j + k] = \text{mangMu}[n - i - 1][j + k] - \text{tmpMangMu}[j] * x[i + 1]$

```

Else: mangMu[i][j + k] = mangMu[n - i - 1][j + k] + tmpMangMu[j] * 1
Sao chép mangMu[i] => tmpMangMu
mangMu[i] = mangMu[n - i - 1] * y[n - 1][n - i - 1] / (giaiThua(n-i-1) * pow(denta,(n - i -
1))

```

### **Bước 25 : Sử dụng sơ đồ Hoocne để tính giá trị đa thức tại 1 điểm xác định.**

- Dữ liệu vào: n, c,  $a_i$  ( $i=1,n$ ).
- Dữ liệu ra:  $P(c)$ .
- Ý tưởng: Đặt  $p = a_0$   
Lặp  $i = 1$  to  $n$  :  $p = p * c + a_i$

### **Bước 26 : Hàm tính giá trị bằng Hoocne**

Funtion hoocne(arr[], n)

Dữ liệu vào: c;

$p = arr[0]$

for  $i = 0$  to  $n$  :

$p = p * c + arr[i];$

return p;

### **Bước 27: lấy p chữ số thập phân sau dấu phẩy**

Dữ liệu vào: số cần hiển thị num, số chữ số sau dấu phẩy p (vd: num = 123.45678, p= 2)

Dữ liệu ra: số num với phần thập phân có p chữ số (num = 123.45)

Ý tưởng: Chia 1 số thành 2 phần:

Phần 1: Phần số nguyên + dấu “.”

Phần 2: Phần số thập phân. Ta chú trọng xử lý phần này.

### **Bước 28: Hàm hiển thị số thập phân**

Funtion hienThiSoTP

current\_digit : biến lưu số thập phân đang xét

next\_digit : biến lưu số thập phân ngay sau số thập phân đang xét

if (num < 0 && num > -1) : print(“-0.”)

else : print( (int)num. ) //in phần nguyên của num và dấu “.”

//xử lý phần thập phân

for (i = 0 to n - 1) :

```

current_digit = (int) ((num * pow (10, i + 1)) % 10 )
next_digit = (int) ((num * pow (10, i + 2)) % 10 )
if (i == n - 1 && next_digit > 4 ) // bước làm tròn
    print ( abs (current_digit + 1) )
else: print ( abs (current_digit + 1) )

```

### **Bước 29: Hàm xóa nội dung trong file**

Ý tưởng: sử dụng chế độ “w” để xóa dữ liệu trong file

Function clearFile

```

FILE * f = fopen (“newton.txt” , “w”)

fclose(f)

```

### **Bước 30: Các hàm in kết quả ra file**

```

void ghiFile1 /*Hàm in bảng các mốc điểm */
void ghiFile2 /*Hàm in bảng sai phân thuận ra FILE*/
void ghiFile3 /*Hàm in bảng sai phân nghịch ra FILE*/
void ghiFile4 /*Hàm in bảng tỷ hiệu thuận ra FILE*/
void ghiFile5 /*Hàm in bảng tỷ hiệu nghịch ra FILE*/
void ghiFileHoocle /*Hàm in lược đồ Hoocne ra FILE*/
void inDaThuc1 /*Hàm in đa thức Newton tiến mốc bất kỳ ra FILE*/
void inDaThuc2 /*Hàm in đa thức Newton lùi mốc bất kỳ ra FILE*/
void inDaThuc3 /*Hàm in đa thức Newton tiến mốc cách đều ra FILE*/
void inDaThuc4 /*Hàm in đa thức Newton lùi mốc cách đều ra FILE*/
void inHeSo/*Hàm hiển thị ma trận hệ số và hệ số ra File*/

```

Ý tưởng chung: lấy các kết quả đã được thực hiện từ các hàm chức năng bên trên để truyền vào hàm.

Coppy các thao tác in ra màn hình và sửa thành .

Các thao tác được sử dụng trong các hàm in kết quả ra file là:

- FILE \* fout = fopen (“newton.txt”, “a”) : khai báo biến fout (sử dụng con trỏ file) để mở tệp Newton.txt
- chế độ: “a” : mở 1 file văn bản và ghi tiếp vào cuối văn bản, nếu nó chưa tồn tại thì file mới được tạo
- fclose( fout )

## II. Báo cáo kết quả chạy chương trình

### VD1:

Kết quả chạy trong FILE .txt

=====

\*.Da lua chon Item 1: Nhap cac moc (x,y)

Bang cac moc noi suy da nhap la:

x	0.00	1.00	-5.00	7.00
---	------	------	-------	------

---

y	1.00	-4.00	-94.00	302.00
---	------	-------	--------	--------

\_Goi y\_: + Neu la moc cach deu chon Item 3 de in bang sai phan thuan !

+ Neu la moc khong cach deu chon Item 2 de in bang ty hieu thuan !

\*.Da lua chon Item 4: Lap va in ra da thuc noi suy Newton tien bat ky

Da thuc noi suy Newton tien moc bat ky la:

$$P(x) = 1.00 - 5.00x - 4.00x(x - 1.00) + 1.00x(x - 1.00)(x + 5.00)$$

\_Goi y\_: Tiep tục chon Item 8 neu ban muon tinh gia tri cua da thuc tai 1 diem bang so do Hoocone !

\*. Da chon Item 8: In ra he so cua da thuc voi so mu lan luot

Tinh gia tri cua da thuc tai 1 diem bang so do hoocone

Ma tran he so la:

1.00	0.00	0.00	0.00
0.00	-5.00	0.00	0.00
0.00	4.00	-4.00	0.00
0.00	-5.00	4.00	1.00

He so mu cua da thuc la:

Mu 0 : 1.00

Mu 1 : -6.00

Mu 2 : 0.00

Mu 3 : 1.00

Nhap gia tri can tinh :

5.000000

Qua trinh tinh toan trong luoc do Hoocne:

Qua trinh tinh midP la:

midP(1) = 5.00

midP(2) = 25.00

midP(3) = 95.00

Qua trinh tinh p la:

p(0) = 1.00

p(1) = 5.00

p(2) = 19.00

p(3) = 96.00

1.00    0.00    -6.00    1.00

5.00    25.00    95.00

-----  
1.00    5.00    19.00    96.00

Gia tri cua da thuc tai 5.00 la: f( 5.00 ) = 96.00

=====



## VD2:

Kết quả chạy trong FILE .txt

=====

\*.Da lua chon Item 1: Nhap cac moc (x,y)

Bang cac moc noi suy da nhap la:

x	-4.00	-2.00	0.00	2.00	4.00	6.00
---	-------	-------	------	------	------	------

---

y	-2759.00	-127.00	1.00	-71.00	-727.00	-1199.00
---	----------	---------	------	--------	---------	----------

\_Goi y\_: + Neu la moc cach deu chon Item 3 de in bang sai phan thuan !

+ Neu la moc khong cach deu chon Item 2 de in bang ty hieu thuan !

\*.Da lua chon Item 3: Lap va in ra bang sai phan thuan

Bang sai phan Thuan la:

x	Y	Y^1	Y^2	Y^3	Y^4	Y^5
-4.00	-2759.00	2632.00	-2504.00	2304.00	-2688.00	3840.00
-2.00	-127.00	128.00	-200.00	-384.00	1152.00	
0.00	1.00	-72.00	-584.00	768.00		
2.00	-71.00	-656.00	184.00			
4.00	-727.00	-472.00				
6.00	-1199.00					

\_Goi y\_: + Co the tiep tục chon Item 6 ung voi truong hop bang sai phan Thuan va In ngay da thuc noi suy newton tien moc cach deu !

+ hoac Item 7 neu muon in lai bang sai phan Nghich va va In ngay da thuc noi suy newton lui moc cach deu !

\*.Da lua chon Item 6: Lap va in ra da thuc noi suy Newton tien cach deu

Da thuc Newton tien moc cach deu la:

$$P(x) = -2759.00 + 1316.00(x + 4.00) - 313.00(x + 4.00)(x + 2.00) + 48.00(x + 4.00)(x + 2.00)x - 7.00(x + 4.00)(x + 2.00)x(x - 2.00) + 1.00(x + 4.00)(x + 2.00)x(x - 2.00)(x - 4.00)$$

\_Goi y\_: Tiep tục chọn Item 8 nếu bạn muốn tính giá trị của đa thức tại 1 điểm bảng số do Hoocone !

\*. Đã chọn Item 8: In ra hệ số của đa thức với số mũ lần lượt

Tính giá trị của đa thức tại 1 điểm bảng số do hoocone

Ma trận hệ số là:

-1199.00	0.00	0.00	0.00	0.00	0.00
1416.00	-236.00	0.00	0.00	0.00	0.00
552.00	-230.00	23.00	0.00	0.00	0.00
-768.00	704.00	-192.00	16.00	0.00	0.00
0.00	-144.00	132.00	-36.00	3.00	0.00
0.00	-96.00	40.00	20.00	-10.00	1.00

Hệ số mũ của đa thức là:

Mu 0 : 1.00

Mu 1 : -2.00

Mu 2 : 3.00

Mu 3 : 0.00

Mu 4 : -7.00

Mu 5 : 1.00

Nhập giá trị cần tính :

1.500000

Qua trình tính toán trong lược đồ Hoocone:

Qua trình tính midP là:

midP(1) = 1.50

midP(2) = -8.25

$$\text{midP}(3) = -12.37$$

$$\text{midP}(4) = -14.06$$

$$\text{midP}(5) = -24.09$$

Qua trình tính p là:

$$p(0) = 1.00$$

$$p(1) = -5.50$$

$$p(2) = -8.25$$

$$p(3) = -9.37$$

$$p(4) = -16.06$$

$$p(5) = -23.09$$

Số đo hoocne là:

$$1.00 \quad -7.00 \quad 0.00 \quad 3.00 \quad -2.00 \quad 1.00$$

$$1.50 \quad -8.25 \quad -12.37 \quad -14.06 \quad -24.09$$

-----

$$1.00 \quad -5.50 \quad -8.25 \quad -9.37 \quad -16.06 \quad -23.09$$

Giá trị của đa thức tại 1.50 là:  $f(1.50) = -23.09$

=====

### VD3:

Tìm đa thức nội suy Newton lùi cho hàm  $y = \sin(\pi x)$  và tính giá trị tại  $x = 1/6$

x	-1.5	0	0.6667	0.8	3.5
y = f(x)	1	0	0.8660	0.5878	-1

Chạy chương trình, lấy kết quả từ file .txt

=====

\*.Đã lựa chọn Item 1: Nhập các mốc (x,y)

Bảng các mốc nội suy đã nhập là:

x	-1.50	0.00	0.67	0.80	3.50
---	-------	------	------	------	------

---

y	1.00	0.00	0.87	0.59	-1.00
---	------	------	------	------	-------

\_Goi y\_: + Nếu là mốc cách đều chọn Item 3 để in bảng sai phân thuận !

+ Nếu là mốc không cách đều chọn Item 2 để in bảng tỷ hiệu thuận !

\*.Đã lựa chọn Item 5: Lập Bảng tỷ hiệu NGHỊCH và in ra đa thức nội suy Newton lùi bất kỳ

Bảng tỷ hiệu NGHỊCH là:

x	Y	T <sup>1</sup>	T <sup>2</sup>	T <sup>3</sup>	T <sup>4</sup>
-1.50	1.00				
0.00	0.00	0.66			
0.67	0.87	1.210	0.91		
0.80	0.59	-2.08	-4.23	-2.23	
3.50	-1.00	0.58	0.53	1.36	0.72

Đa thức nội suy Newton lùi với mốc bất kỳ là:

$$P(x) = -1.00 - 0.59(x - 3.50) + 0.53(x - 3.50)(x - 0.80) + 1.36(x - 3.50)(x - 0.80)(x - 0.67)0.72(x - 3.50)(x - 0.80)(x - 0.67)x$$

\_Goi y\_: Tiếp tục chọn Item 8 nếu bạn muốn tính giá trị của đa thức tại 1 điểm bảng số do Hocne !

\*. Da chon Item 8: In ra he so cua da thuc voi so mu lan luot

Tinh gia tri cua da thuc tai 1 diem bang so do hoocne

Ma tran he so la:

-1.00	0.00	0.00	0.00	0.00
2.06	-0.58	0.00	0.00	0.00
1.48	-2.27	0.53	0.00	0.00
-2.53	7.71	-6.75	1.36	0.00
0.00	-1.34	4.07	-3.57	0.72

He so mu cua da thuc la:

Mu 0 : 0.00

Mu 1 : 3.00

Mu 2 : -2.33

Mu 3 : -2.66

Mu 4 : 0.01

Nhap gia tri can tinh :

0.166667

Qua trinh tinh toan trong luoc do Hoocne:

Qua trinh tinh midP la:

midP(1) = 0.12

midP(2) = -0.34

midP(3) = -0.41

midP(4) = 0.51

Qua trinh tinh p la:

$$p(0) = 0.72$$

$$p(1) = -2.09$$

$$p(2) = -2.50$$

$$p(3) = 3.09$$

$$p(4) = 0.51$$

So do hocne la:

$$0.72 \quad -2.21 \quad -2.15 \quad 3.50 \quad 0.00$$

$$0.12 \quad -0.34 \quad -0.41 \quad 0.51$$

-----

$$0.72 \quad -2.09 \quad -2.50 \quad 3.09 \quad 0.51$$

Gia tri cua da thuc tai 0.17 la:  $f(0.17) = 0.51$

=====

Vay  $f(1/6) \approx 0.51$

#### VD4:

Tìm đa thức nội suy Newton lùi cho hàm  $y = \ln x$  và tính giá trị tại  $x = 1.5$  với các mốc nội suy trong bảng:

x	1	2	3
$y = f(x)$	0	0.6932	1.0986

Chạy chương trình, lấy kết quả từ file .txt

\*.Da lua chon Item 1: Nhap cac moc (x,y)

Bang cac moc noi suy da nhap la:

x      1.0000 2.0000 3.0000

---

y      0.0000 0.6932 1.0986

\_Goi y\_: + Neu la moc cach deu chon Item 3 de in bang sai phan thuan !

+ Neu la moc khong cach deu chon Item 2 de in bang ty hieu thuan !

\*.Da lua chon Item 7: Lap Bang sai phan NGHICH va in ra da thuc noi suy Newton tien cach deu

Bang sai phan NGHICH la:

x	Y	Y^1	Y^2
1.0000	0.0000		
2.0000	0.6932	0.6932	
3.0000	1.0986	0.4054	- 0.2878

Da thuc Newton lui moc cach deu la:

$$P(x) = 1.10 + 0.41(x - 3.00) + -0.14(x - 3.00)(x - 2.00)$$

\_Goi y\_: Tiep tục chon Item 8 neu ban muon tinh gia tri cua da thuc tai 1 diem bang so do Hoocone !

\*. Da chon Item 8: In ra he so cua da thuc voi so mu lan luot

Tinh gia tri cua da thuc tai 1 diem bang so do hoocone

Ma tran he so la:

1.0986	0.0000	0.0000
-1.2162	0.4054	0.0000
-0.8634	0.7195	-0.1439

He so mu cua da thuc la:

Mu 0 : -0.9999

Mu 1 : 1.2222

Mu 2 : -0.3333

Nhap gia tri can tinh :

1.500000

Qua trinh tinh toan trong luoc do Hoocne:

Qua trinh tinh midP la:

midP(1) = -0.2158

midP(2) = 1.3636

Qua trinh tinh p la:

p(0) = -0.1439

p(1) = 0.9091

p(2) = 0.3826

So do hoocne la:

-0.1439 1.1249 -0.9810



$$-0.2158 \quad 1.3636$$

-----

$$-0.1439 \quad 0.9091 \quad 0.3826$$

Gia tri cua da thuc tai 1.5000 la:  $f(1.5000) = 0.3826$

=====

==

Vậy  $\ln(1.5) = 0.3826$

### III. Tổng kết

#### 1. Về phương pháp:

- Ưu điểm: Khắc phục được nhược điểm của phương pháp Lagrange (là không phải tính lại từ đầu với mỗi mốc dữ liệu mới).
- Hạn chế: Số bậc của đa thức nội suy gắn liền với số mốc nội suy, tức để độ chính xác của kết quả càng cao thì số mốc nội suy càng lớn, nhưng độ phức tạp của đa thức cũng tăng theo.

#### 2. Về chương trình:

- Chương trình đã thực hiện được toàn bộ yêu cầu của đề bài.
- Chương trình tương đối thân thiện với người dùng, phần Menu có kèm theo hướng dẫn.
- Mặc dù đã rất cố gắng nhưng chương trình vẫn còn nhiều hạn chế như: thiết kế chương trình chưa được tối ưu, hiển thị kết quả ra màn hình còn chưa đẹp, độ phức tạp của một số hàm lớn, một số hàm in ra FILE còn cồng kềnh,...

#### 3. Ứng dụng:

- Đơn giản hóa cách tính giá trị của hàm cho trước (Ví dụ:  $\sin x$ ,  $\log x$ ,...).
- Biểu diễn các mốc điểm cho trước thành một đa thức nội suy.
- .....

## LỜI CẢM ƠN

*Kỳ học 20192 vừa qua thực sự là một kỳ học rất đặc biệt với toàn thể giảng viên, sinh viên trường đại học Bách khoa Hà Nội trước tình hình dịch bệnh diễn biến hết sức phức tạp. Tuy nhiên, nhờ sự chỉ đạo kịp thời, quyết liệt của lãnh đạo Nhà trường, lãnh đạo Viện, Bách khoa là một trong những ngôi trường đầu tiên trên cả nước đón sinh viên quay trở lại.*

*Đối với môn học này, Chúng em xin được gửi lời cảm ơn trân thành tới cô Nguyễn Thị Thanh Huyền vì đã cùng Chúng em vượt qua kỳ học đầy thách thức vừa qua, cũng như là người hướng dẫn Chúng em thực hiện đề tài này.*

*Mặc dù đã rất cố gắng, xong trong quá trình làm bài báo cáo khó tránh khỏi những sai sót, rất mong Cô thông cảm. Đồng thời, do kỹ thuật lập trình còn nhiều hạn chế, kiến thức vận dụng mới mẻ cũng như thời gian có hạn nên bài tập lớn còn hạn chế, thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp quý báu từ Cô để có thêm nhiều kinh nghiệm cũng như hoàn thành tốt hơn các bài tập lớn khác.*

*Em xin chân thành cảm ơn!*