

# **DANH DÁCH ĐỀ TÀI**

## **ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN**

( Dành cho chương trình CLC\_HTDN/ CLC Tiếng Nhật)

### **Mã đề tài:**

<b>Đề 101:</b> .....	2
<b>Đề 201:</b> .....	4
<b>Đề 301:</b> .....	7
<b>Đề 401:</b> .....	21
<b>Đề 601:</b> .....	43
<b>Đề 701:</b> .....	44
<b>Đề 801:</b> .....	49
<b>Đề 901:</b> .....	57
<b>Đề A01:</b> .....	60
<b>Đề B01:</b> .....	70
<b>Đề C01:</b> .....	73
<b>The End</b> .....	81

**Đề 101:**

Ước lượng đa thức  $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  tại  $x=c$ .

*Khai báo dữ liệu:* Danh sách đặc A (mảng A) gồm  $n+1$  hệ số của đa thức  $P(x)$ .

*Yêu cầu:*

- Đọc  $n$  và các hệ số từ file \*.INP hoặc tạo ngẫu nhiên.
- Ghi kết quả  $P(c)$  ra file \*.OUT hoặc màn hình.

**Đề 102:** Ước lượng đa thức  $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  tại  $x=c$ .

*Khai báo dữ liệu:* Danh sách liên kết A (một chiều hoặc hai chiều) gồm  $n+1$  hệ số của đa thức  $P(x)$ .

*Yêu cầu:*

- Đọc  $n$  khá lớn và các hệ số từ file \*.INP hoặc tạo ngẫu nhiên.
- Ghi kết quả  $P(c)$  ra file \*.OUT hoặc màn hình.

**Đề 103:** Biến đổi đa thức  $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  theo  $P(x-c)$ .

*Khai báo dữ liệu:* Mảng 2 chiều A gồm  $n+1$  hàng và  $n+1$  cột để tính các hệ số  $(x-c)^k$ .

*Yêu cầu:*

- Đọc  $n$  và các hệ số từ file \*.INP hoặc tạo ngẫu nhiên.
- Ghi kết quả là  $c$  các hệ số của  $P(x-c)$  ra file \*.OUT hoặc màn hình.

**Đề 104:** Tìm nghiệm đa thức  $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  bằng phương pháp chia đôi.

*Khai báo dữ liệu:* Danh sách đặc A (mảng A) gồm  $n+1$  hệ số của đa thức  $P(x)$ .

*Yêu cầu:*

- Đọc  $n$  và các hệ số từ file \*.INP hoặc tạo ngẫu nhiên.
- Nhập  $a, b$  cho khoảng nghiệm  $[a, b]$  với  $P(a)P(b) < 0$ .
- Ghi kết quả là nghiệm  $x=c$  ra file \*.OUT hoặc màn hình.

**Đề 105:** Tìm hạng ma trận.

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận A cần tính hạng gồm  $m$  hàng và  $n$  cột.

*Yêu cầu:*

- Dùng các phép biến đổi sơ cấp để đưa ma trận về dạng hình thang.
- Đọc  $m, n$  và ma trận A từ file \*.INP hoặc tạo ngẫu nhiên.
- Ghi kết quả số  $r$  là hạng của ma trận A ra file \*.OUT hoặc màn hình.

**Đề 106:** Tính định thức của ma trận vuông cấp  $n$ .

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận A vuông cần tính định thức gồm  $n$  hàng

và  $n$  cột.

*Yêu cầu:*

- Dùng các phép biến đổi để đưa định thức về dạng tam giác.
- Đọc  $n$  và ma trận  $A$  từ file \*.INP hoặc tạo ngẫu nhiên.
- Ghi kết quả số  $d$  là định thức của ma trận  $A$  ra file \*.OUT hoặc màn hình.

**Đề 107:** Tính định thức của ma trận vuông cấp  $n$ .

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận  $A$  vuông cần tính định thức gồm  $n$  hàng và  $n$  cột.

*Yêu cầu:*

- Dùng kỹ thuật đệ quy bằng cách khai triển theo một hàng hoặc cột để hạ cấp định thức.
- Đọc  $n$  và ma trận  $A$  từ file \*.INP hoặc tạo ngẫu nhiên.
- Ghi kết quả số  $d$  là định thức của ma trận  $A$  ra file \*.OUT hoặc màn hình.

**Đề 108:** Tính ma trận nghịch đảo.

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận  $A$  vuông cần tính nghịch đảo gồm  $n$  hàng và  $n$  cột.

*Yêu cầu:*

- Biến đổi sơ cấp về hàng trên ma trận  $[A|E]$   $n$  hàng và  $2n$  cột sao cho  $A$  trở thành  $E$  thì  $E$  trở thành  $A^{-1}$ .
- Đọc  $n$  và ma trận  $A$  từ file \*.INP hoặc tạo ngẫu nhiên.
- Ghi kết quả là ma trận nghịch đảo  $A^{-1}$  ra file \*.OUT hoặc màn hình.

**Đề 109:** Giải hệ phương trình Crame cấp  $n$  ( $n$  phương trình,  $n$  ẩn)

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận  $A$  vuông cấp  $n$  và mảng một chiều  $B$  gồm  $n$  số.

*Yêu cầu:*

- Dùng công thức Crame để tính nghiệm qua định thức.
- Đọc  $n$ , ma trận  $A$  và cột  $B$  từ file \*.INP hoặc tạo ngẫu nhiên.
- Ghi kết quả  $n$  số  $x_1, x_2, \dots, x_n$  là nghiệm của phương trình ra file \*.OUT hoặc màn hình.

**Đề 110:** Giải hệ phương trình Crame cấp  $n$  ( $n$  phương trình,  $n$  ẩn)

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận  $A$  vuông cấp  $n$  và mảng một chiều  $B$  gồm  $n$  số.

*Yêu cầu:*

- Dùng phương pháp Solexki để tách ma trận A thành tích hai ma trận tam giác.
- Đọc  $n$ , ma trận A và cột B từ file \*.INP hoặc tạo ngẫu nhiên.
- Ghi kết quả  $n$  số  $x_1, x_2, \dots, x_n$  là nghiệm của phương trình ra file \*.OUT hoặc màn hình.

### **Đề 201:**

Dùng phương pháp Chia đôi để tìm nghiệm đa thức bậc  $n$ :  $P(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x + a_{n+1} = 0$ .

Khai báo dữ liệu: Mảng 2 chiều cho ma trận A cấp  $n$ .

*Yêu cầu:*

- Đọc số nguyên dương  $n$  và ma trận vuông cấp  $n$  từ file DATA.INP.
- Tính tổng của từng cột để có các hệ số theo:  $a_i = a_{0i} + a_{1i} + a_{2i} + \dots + a_{ni}$ , với mọi  $i=0, 1, 2, \dots, n$ .
- Ghi kết quả ra file hoặc lên màn hình.

**Đề 202:** Dùng phương pháp Dây cung để tìm nghiệm đa thức bậc  $n$ :  $P(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x + a_{n+1} = 0$ .

Khai báo dữ liệu: Mảng 2 chiều cho ma trận A cấp  $n$ .

*Yêu cầu:*

- Đọc số nguyên dương  $n$  và ma trận vuông cấp  $n$  từ file DATA.INP.
- Tính tổng của từng hàng để có các hệ số theo:  $a_i = a_{i0} + a_{i1} + a_{i2} + \dots + a_{in}$ , với mọi  $i=0, 1, 2, \dots, n$ .
- Ghi kết quả ra file hoặc lên màn hình.

**Đề 203:** Dùng phương pháp Gauss để giải hệ phương trình gồm  $n$  phương trình,  $n$  ẩn:  
 $A.X=B$

Khai báo dữ liệu: Mảng 2 chiều cho ma trận  $A$ .

Yêu cầu:

- Đọc từ file DATA.INP gồm số nguyên dương  $m$ , và  $m$  hàng tiếp theo, mỗi hàng gồm  $m$  phần tử các số thực (Ta được ma trận vuông  $A$  cấp  $m$ ).
- Xóa hàng ở giữa. Đặt  $n=m-1$ . Ta được hệ phương trình gồm  $n$  phương trình,  $n$  ẩn, với cột cuối cùng là vector  $B$ .
- Lưu kết quả ra file DATA.OUT

**Đề 204:** Dùng phương pháp Gauss để giải hệ phương trình gồm  $n$  phương trình,  $n$  ẩn:  
 $A.X=B$

Khai báo dữ liệu: Mảng 2 chiều cho ma trận  $A$  cấp  $n$ .

Yêu cầu:

- Đọc số nguyên dương  $n$  và ma trận vuông cấp  $n$  từ file DATA.INP.
- Thêm cột cuối cùng, mà mỗi phần tử  $B_i$  là tổng của các phần tử đứng trước nó trên hàng đó, tức là:

$$B_i = A_{i1} + A_{i2} + \dots + A_{in}$$

- Lưu kết quả ra file DATA.OUT

**Đề 205:** Dùng phương pháp Gauss để giải hệ phương trình gồm  $n$  phương trình,  $n$  ẩn:  
 $A.X=B$

Khai báo dữ liệu: Mảng 2 chiều cho ma trận  $A$  cấp  $n$ .

Yêu cầu:

- Từ file DATA.INP , hãy đọc số nguyên dương  $n$  và số nguyên  $m$  (với  $m > n$ ), tiếp theo đọc  $n$  hàng, mỗi hàng gồm  $m$  số thực để tạo ma trận  $A$  gồm  $n$  hàng  $m$  cột.
- Cộng dồn các phần tử phía sau kể từ cột thứ  $n+1$  trở đi để có được các hệ số  $B$  tương ứng, tức là:

$$B_i = a_{i(n+1)} + a_{i(n+2)} + a_{i(n+3)} + \dots + a_{im}, \text{ với mọi } i=1, 2, \dots, n$$

- Lưu kết quả ra file DATA.OUT

**Đề 206:** Dùng phương pháp Cramer để giải hệ phương trình gồm  $n$  phương trình ,  $n$  ẩn:  $A.X=B$

Khai báo dữ liệu: Mảng 2 chiều cho ma trận  $A$ .

Yêu cầu:

- Đọc từ file DATA.INP gồm số nguyên dương  $m$  , và  $m$  hàng tiếp theo, mỗi hàng gồm  $m$  phần tử các số thực (Ta được ma trận vuông  $A$  cấp  $m$ ).
- Xóa hàng ở giữa. Đặt  $n=m-1$ . Ta được hệ phương trình gồm  $n$  phương trình,  $n$  ẩn, với cột cuối cùng là vector  $B$ .
- Lưu kết quả ra file DATA.OUT

**Đề 207:** Dùng phương pháp Cramer để giải hệ phương trình gồm  $n$  phương trình ,  $n$  ẩn:  $A.X=B$

Khai báo dữ liệu: Mảng 2 chiều cho ma trận  $A$  cấp  $n$ .

Yêu cầu:

- Đọc số nguyên dương  $n$  và ma trận vuông cấp  $n$  từ file DATA.INP.
- Thêm cột cuối cùng, mà mỗi phần tử  $B_i$  là tổng của các phần tử đứng trước nó trên hàng đó, tức là:

$$B_i = A_{i1} + A_{i2} + \dots + A_{in}$$

- Lưu kết quả ra file DATA.OUT

**Đề 208:** Dùng phương pháp Cramer để giải hệ phương trình gồm  $n$  phương trình,  $n$  ẩn:  $A.X=B$

Khai báo dữ liệu: Mảng 2 chiều cho ma trận  $A$  cấp  $n$ .

Yêu cầu:

- Từ file DATA.INP, hãy đọc số nguyên dương  $n$  và số nguyên  $m$  (với  $m > n$ ), tiếp theo đọc  $n$  hàng, mỗi hàng gồm  $m$  số thực để tạo ma trận  $A$  gồm  $n$  hàng  $m$  cột.

- Cộng dồn các phần tử phía sau kể từ cột thứ  $n+1$  trở đi để có được các hệ số  $B$  tương ứng, tức là:

$$B_i = a_{i(n+1)} + a_{i(n+2)} + a_{i(n+3)} + \dots + a_{im}, \text{ với mọi } i=1, 2, \dots, n$$

- Lưu kết quả ra file DATA.OUT

### **Đề 301:**

Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc  $N$  ( $N \geq 7$ ) phần tử số của danh sách liên kết từ file DAYSO.IN

b) Thực hiện xóa phần tử ở vị trí  $\frac{N}{2}$  (nếu  $N$  lẻ thì làm tròn), hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Sau khi thực hiện câu (b), ta có được một danh sách liên kết có  $M$  phần tử  $(a_0, a_1, \dots, a_{m-1})$ . Biết rằng các phần tử này là hệ số của đa thức sau:  $G(x) = \sum_{i=0}^{M-1} a_i x^i$ . Hãy tìm các nghiệm của phương trình  $G(x) = 0$  bằng **phương pháp chia đôi** (với sai số  $\varepsilon = 10^{-5}$ ), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết đơn* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 302:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

- a) Đọc  $N$  ( $N \geq 7$ ) phần tử số của danh sách liên kết từ file DAYSO.IN
- b) Thực hiện thêm một phần tử (số) vào vị trí  $\frac{N}{2}$  (làm tròn nếu  $N$  lẻ), hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.
- c) Sau khi thực hiện câu b), chúng ta nhận được một danh sách liên kết có  $M$  phần tử ( $a_0, a_1, \dots, a_{M-1}$ ). Biết rằng các phần tử này là hệ số của đa thức sau:  $G(x) = \sum_{i=0}^{M-1} a_i x^i$
- Tìm các nghiệm của phương trình  $G(x) = 0$  bằng **phương pháp tiếp tuyến** (với sai số  $\varepsilon = 10^{-5}$ ), sau đó hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết đơn* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 303:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

- a) Đọc vào  $N$  danh sách từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN bao gồm khai báo  $N$  hàng ( $N \geq 5$ ), mỗi hàng gồm  $M$  phần tử số,  $M=N-1$ . Mỗi hàng này tương ứng là một danh sách liên kết.
- b) Thực hiện thêm vào mỗi danh sách này một phần tử tại vị trí  $\frac{M}{2}$  (làm tròn nếu  $M$  lẻ), sau đó hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.
- c) Sau khi thực hiện ở câu b), chúng ta sẽ nhận được  $N$  danh sách liên kết, mỗi danh sách này sẽ có  $N$  phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau (mỗi danh sách tương ứng mỗi hàng của ma trận):

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

Hãy tìm nghiệm  $X = (x_0, x_1, \dots, x_N)^T$  của hệ phương trình  $AX = B$  bằng **phương pháp lặp đơn**, với  $B = (b_1, b_2, \dots, b_N)^T$  (với  $B$  là mảng một chiều được nhập vào), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết đơn* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---



---

**Đề 304:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

- a) Đọc vào  $N$  danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N$  hàng ( $N \geq 5$ ), mỗi hàng gồm  $M$  phần tử số,  $M=N+1$ . Mỗi hàng này tương ứng là một danh sách được đọc vào.
- b) Thực hiện thêm xóa bớt ở mỗi danh sách một phần tử tại vị trí  $\frac{M}{2}$  (có làm tròn nếu  $M$  lẻ), hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.
- c) Sau khi thực hiện ở câu b), chúng ta sẽ nhận được  $N$  danh sách liên kết, mỗi danh sách sẽ có  $N$  phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo ma trận sau (mỗi danh sách tương ứng mỗi hàng của ma trận):

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

Hãy tìm nghiệm  $X = (x_0, x_1, \dots, x_N)^T$  của hệ phương trình  $AX = B$  bằng **phương pháp Gauss - Siedel**, với  $B = (b_1, b_2, \dots, b_N)^T$  (với  $B$  là mảng một chiều bất kỳ được nhập vào), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết đơn* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 305:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

- a) Đọc vào  $N$  danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N$  hàng ( $N \geq 5$ ), mỗi hàng gồm  $M$  phần tử số,  $M=N-1$ . Mỗi hàng này tương ứng là một danh sách).
- b) Thực hiện thêm thêm vào mỗi danh sách liên kết một phần tử tại vị trí  $\frac{M}{2}$  (có làm tròn nếu  $M$  lẻ), hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Sau khi thực hiện ở câu b), chúng ta sẽ nhận được  $N$  danh sách liên kết, mỗi danh sách liên kết sẽ có  $N$  phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau (mỗi danh sách tương ứng mỗi hàng của ma trận):

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

Hãy tìm nghiệm  $X = (x_0, x_1, \dots, x_N)^T$  của hệ phương trình  $AX = B$  bằng **bảng phương pháp giảm dư**, với  $B = (b_1, b_2, \dots, b_N)^T$  (với B là mảng một chiều được nhập vào), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết đơn* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

**Đề 306:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào  $N=3$  danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N = 3$  hàng, mỗi hàng gồm  $M$  phần tử số ( $M \geq 7$ ). Mỗi hàng này tương ứng là một danh sách liên kết).

b) Thực hiện xóa hai phần tử vào vị trí  $\frac{M}{4}$  và  $\frac{M}{2}$  (làm tròn nếu  $M$  lẻ) đối với mỗi danh sách **thứ nhất** và **thứ ba** (mỗi danh sách này xóa 2 phần tử số); thực hiện xóa một phần tử vào vị trí  $\frac{M}{4}$  đối với danh sách thứ hai, sau đó hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Kết quả sau khi thực hiện ở câu b), chúng ta nhận được các danh sách liên kết: thứ nhất ( $a_i, i = 1 \dots M - 2$ ), thứ hai ( $b_i, i = 1 \dots M - 1$ ) và thứ ba ( $c_i, i = 1 \dots M - 2$ ). Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau (ma trận ba đường chéo- Tridiagonal matrix\*):

$$A = \begin{pmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 \\ a_1 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & a_2 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & a_3 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & b_{M-2} & c_{M-2} \\ 0 & 0 & 0 & 0 & a_{M-2} & b_{M-1} \end{pmatrix}$$

Hãy tìm nghiệm  $X(x_0, x_1, \dots, x_N)$  của hệ phương trình  $AX = Q$ , với  $Q = (q_1, q_2, \dots, q_N)^T$  (với  $Q$  là mảng một chiều được nhập vào), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*

\* Tridiagonal matrix [https://en.wikipedia.org/wiki/Tridiagonal\\_matrix](https://en.wikipedia.org/wiki/Tridiagonal_matrix)

**Yêu cầu:** Sử dụng *danh sách liên kết đơn* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 307** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc  $N$  ( $N \geq 7$ ) phần tử số của danh sách liên kết từ file DAYSO.IN

b) Thực hiện xóa phần tử trước vị trí  $\frac{N}{2}$  (làm tròn nếu  $N$  lẻ), hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Sau khi thực hiện câu b), chúng ta nhận được một danh sách liên kết có  $M$  phần tử  $(a_0, a_1, \dots, a_{M-1})$ ,  $M = N - 1$ . Biết rằng các phần tử này là hệ số của đa thức sau:

$$G(x) = \sum_{i=0}^{M-1} a_i x^i$$

Hãy tìm các nghiệm của phương trình  $G(x) = 0$  **phương pháp chia đôi** (với sai số  $\varepsilon = 10^{-5}$ ), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết kép* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 308:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc  $N$  ( $N \geq 7$ ) phần tử số của danh sách liên kết từ file DAYSO.IN

b) Thực hiện thêm một phần tử sau phần tử thứ  $\frac{N}{2}$  (làm tròn nếu  $N$  lẻ), hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Sau khi thực hiện câu b), chúng ta nhận được một danh sách liên kết có  $M$  phần tử  $(a_0, a_1, \dots, a_{M-1})$ ,  $M = N + 1$ . Biết rằng các phần tử này là hệ số của đa thức sau:

$$G(x) = \sum_{i=0}^{M-1} a_i x^i$$

Hãy tìm các nghiệm của phương trình  $G(x) = 0$  bằng **phương pháp tiếp tuyến** (với sai số  $\varepsilon = 10^{-5}$ ), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết kép* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 309:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào  $N$  danh sách từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN bao gồm  $N$  hàng ( $N \geq 5$ ), mỗi hàng gồm  $M$  phần tử số,  $M=N-1$ . Mỗi hàng này tương ứng là một danh sách liên kết.

b) Thực hiện thêm vào mỗi danh sách liên kết một phần tử trước phần tử thứ  $\frac{M}{2}$  (làm tròn nếu  $M$  lẻ), hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Kết quả sau khi thực hiện ở câu b), chúng ta nhận được  $N$  danh sách liên kết, mỗi danh sách có  $N$  phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau (mỗi danh sách tương ứng mỗi hàng của ma trận):

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

Hãy tìm nghiệm  $X = (x_0, x_1, \dots, x_N)^T$  của hệ phương trình  $AX = B$  bằng **phương pháp lặp đơn**, với  $B = (b_1, b_2, \dots, b_N)^T$  ( $B$  là mảng một chiều bất kỳ nhập vào), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết kép* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 310:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào  $N$  danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N$  hàng ( $N \geq 5$ ), mỗi hàng gồm  $M$  phần tử số,  $M=N+1$ . Mỗi hàng này tương ứng là một danh sách liên kết.

b) Thực hiện xóa bớt ở mỗi danh sách liên kết một phần tử sau phần tử thứ  $\frac{M}{2}$  (làm tròn nếu  $M$  lẻ), hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Kết quả sau khi thực hiện ở câu b), chúng ta nhận được  $N$  danh sách liên kết, mỗi danh sách liên kết có  $N$  phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau (mỗi danh sách tương ứng mỗi hàng của ma trận):

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

Hãy tìm nghiệm  $X = (x_0, x_1, \dots, x_N)^T$  của hệ phương trình  $AX = B$  bằng **phương pháp Gauss - Siedel**, với  $B = (b_1, b_2, \dots, b_N)^T$  ( $B$  là mảng một chiều bất kỳ nhập vào), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng danh sách liên kết kép để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

**Đề 311:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào  $N$  danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N$  hàng ( $N \geq 5$ ), mỗi hàng gồm  $M$  phần tử số,  $M=N-1$ . Mỗi hàng này tương ứng là một danh sách).

b) Thực hiện thêm thêm vào mỗi danh sách liên kết một phần tử trước vị trí  $\frac{M}{2}$  (có làm tròn nếu  $M$  lẻ), hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Sau khi thực hiện ở câu b), chúng ta nhận được  $N$  danh sách liên kết, mỗi danh sách liên kết có  $N$  phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau (mỗi danh sách tương ứng mỗi hàng của ma trận):

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

Hãy tìm nghiệm  $X(x_0, x_1, \dots, x_N)$  của hệ phương trình  $AX = B$  bằng **phương pháp giảm dư**, với  $B = (b_1, b_2, \dots, b_N)^T$ , hiển thị kết quả và lưu kết quả ra file **RESULT2.OUT**.

**Yêu cầu:** Sử dụng *danh sách liên kết kép* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file **RESULT1.OUT**, **RESULT2.OUT**.

---

**Đề 312:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào  $N=3$  danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N = 3$  hàng, mỗi hàng gồm  $M$  phần tử số ( $M \geq 7$ ). Mỗi hàng này tương ứng là một danh sách liên kết).

b) Thực hiện xóa hai phần tử vào vị trí  $\frac{M}{4}$  và  $\frac{M}{2}$  (làm tròn nếu  $M$  lẻ) đối với mỗi danh sách **thứ nhất** và **thứ ba** (mỗi danh sách này xóa 2 phần tử số); thực hiện xóa một phần tử vào vị trí  $\frac{M}{4}$  đối với danh sách thứ hai, sau đó hiển thị kết quả và lưu kết quả ra file **RESULT1.OUT**.

c) Kết quả sau khi thực hiện ở câu b), chúng ta nhận được các danh sách liên kết: danh sách thứ nhất ( $a_i, i = 1 \dots M - 2$ ), danh sách thứ hai ( $b_i, i = 1 \dots M - 1$ ) và danh sách thứ ba ( $c_i, i = 1 \dots M - 2$ ). (ma trận ba đường chéo- Tridiagonal matrix\*):

$$A = \begin{pmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 \\ a_1 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & a_2 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & a_3 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & b_{M-2} & c_{M-2} \\ 0 & 0 & 0 & 0 & a_{M-2} & b_{M-1} \end{pmatrix}$$

Hãy tìm nghiệm  $X(x_0, x_1, \dots, x_N)$  của hệ phương trình  $AX = Q$ , với  $Q = (q_1, q_2, \dots, q_N)^T$  (với  $Q$  là mảng một chiều được nhập vào), hiển thị kết quả và lưu kết quả ra file **RESULT2.OUT**.

\* Tridiagonal matrix [https://en.wikipedia.org/wiki/Tridiagonal\\_matrix](https://en.wikipedia.org/wiki/Tridiagonal_matrix)

**Yêu cầu:** Sử dụng *danh sách liên kết kép* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file **RESULT1.OUT**, **RESULT2.OUT**.

---

**Đề 313:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

- Đọc  $N$  ( $N \geq 7$ ) phần tử số của danh sách liên kết từ file DAYSO.IN
- Thực hiện xóa một phần tử trong danh sách này, hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.
- Sau khi thực hiện câu b), chúng ta nhận được một danh sách có  $M$  ( $M=N-1$ ) phần tử số  $(a_0, a_1, \dots, a_{M-1})$ . Biết rằng các phần tử này là hệ số của đa thức sau:

$$G(x) = \sum_{i=0}^{M-1} a_i x^i$$

Hãy tìm các nghiệm của phương trình  $G(x) = 0$  bằng **phương pháp chia đôi** (với sai số  $\varepsilon = 10^{-5}$ ), hiển thị kết quả và lưu kết quả ra file *RES\_2.OUT*

**Yêu cầu:** Sử dụng *danh sách liên kết kiểu hàng đợi* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 314:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

- Đọc  $N$  ( $N \geq 7$ ) phần tử số của danh sách liên kết từ file DAYSO.IN
- Thực hiện xóa một phần tử trong danh sách này, hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.
- Sau khi thực hiện câu b), chúng ta nhận được một danh sách có  $M$  ( $M=N-1$ ) phần tử số  $(a_0, a_1, \dots, a_{M-1})$ . Biết rằng các phần tử này là hệ số của đa thức sau:

$$G(x) = \sum_{i=0}^{M-1} a_i x^i$$

Hãy tìm các nghiệm của phương trình  $G(x) = 0$  bằng **phương pháp tiếp tuyến** (với sai số  $\varepsilon = 10^{-5}$ ), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*

**Yêu cầu:** Sử dụng *danh sách liên kết kiểu hàng đợi* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 315:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào  $N$  ( $N \geq 5$ ), danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N$  hàng mỗi hàng gồm  $M$  phần tử số ( $M = N - 1$ ). Mỗi hàng này tương ứng là một danh sách liên kết).

b) Thực hiện thêm vào mỗi danh sách một phần tử, hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Sau khi thực hiện ở câu b), chúng ta nhận được  $N$  danh sách, mỗi danh sách có  $N$  phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

Hãy tìm nghiệm  $X = (x_0, x_1, \dots, x_N)^T$  của hệ phương trình  $AX = B$  bằng **phương pháp lập đơn**, với  $B = (b_1, b_2, \dots, b_N)^T$  (với  $B$  là mảng một chiều được nhập vào), sau đó hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết kiểu hàng đợi* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file văn bản DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 316:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào ( $N \geq 5$ ), danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N$  hàng, mỗi hàng gồm  $M$  phần tử số ( $M = N + 1$ ). Mỗi hàng này tương ứng là một danh sách liên kết).

b) Thực hiện xóa bớt ở mỗi danh sách một phần tử, hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Kết quả sau khi thực hiện ở câu b), chúng ta nhận được  $N$  danh sách liên kết, mỗi danh sách này có  $N$  phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau (mỗi danh sách là mỗi hàng của ma trận):

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

Hãy tìm nghiệm  $X = (x_0, x_1, \dots, x_N)^T$  của hệ phương trình  $AX = B$  bằng **phương pháp Gauss - Siedel**, với  $B = (b_1, b_2, \dots, b_N)^T$ , hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.



**Yêu cầu:** Sử dụng *danh sách liên kết kiểu hàng đợi* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file văn bản DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 317:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào  $N=3$  danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N = 3$  hàng, mỗi hàng gồm  $M$  phần tử số ( $M \geq 7$ ). Mỗi hàng này tương ứng là một danh sách liên kết).

b) Thực hiện xóa hai phần tử đối với danh sách thứ nhất và thứ ba (mỗi danh sách xóa hai phần tử), xóa một phần tử đối với danh sách thứ hai, hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Kết quả sau khi thực hiện ở câu b), chúng ta nhận được các danh sách: danh sách thứ nhất ( $a_i, i = 1 \dots M - 2$ ), danh sách thứ hai ( $b_i, i = 1 \dots M - 1$ ) và danh sách thứ ba ( $c_i, i = 1 \dots M - 2$ ). (ma trận có ba đường chéo - Tridiagonal matrix\*):

$$A = \begin{pmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 \\ a_1 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & a_2 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & a_3 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & b_{M-2} & c_{M-2} \\ 0 & 0 & 0 & 0 & a_{M-2} & b_{M-1} \end{pmatrix}$$

Hãy tìm nghiệm  $X(x_0, x_1, \dots, x_N)$  của hệ phương trình  $AX = Q$ , với  $Q = (q_1, q_2, \dots, q_N)^T$  (với  $Q$  là mảng một chiều được nhập vào), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

\* Tridiagonal matrix [https://en.wikipedia.org/wiki/Tridiagonal\\_matrix](https://en.wikipedia.org/wiki/Tridiagonal_matrix)

**Yêu cầu:** Sử dụng *danh sách kiểu hàng đợi* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 318:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc  $N$  ( $N \geq 7$ ) phần tử số của danh sách liên kết từ file DAYSO.IN

b) Thực hiện xóa một phần tử trong danh sách này, hiển thị kết quả và lưu kết quả ra file. *RESULT1.OUT*.

c) Sau khi thực hiện câu b), chúng ta nhận được một danh sách liên kết có M phần tử số ( $M = N - 1$ ):  $(a_0, a_1, \dots, a_{M-1})$ . Biết rằng các phần tử này là hệ số của đa thức sau:

$$G(x) = \sum_{i=0}^{M-1} a_i x^i$$

Hãy tìm các nghiệm của phương trình  $G(x) = 0$  bằng **phương pháp chia đôi** (với sai số  $\varepsilon = 10^{-5}$ ), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết kiểu ngăn xếp* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 319:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc N ( $N \geq 7$ ) phần tử số của danh sách liên kết từ file DAYSO.IN

b) Thực hiện xóa một phần tử trong danh sách này, hiển thị kết quả và lưu kết quả ra file. *RESULT1.OUT*.

c) Sau khi thực hiện câu b), chúng ta nhận được một danh sách có M ( $M=N-1$ ) phần tử  $(a_0, a_1, \dots, a_{M-1})$ . Biết rằng các phần tử này là hệ số của đa thức sau:

$$G(x) = \sum_{i=0}^{M-1} a_i x^i$$

Hãy tìm các nghiệm của phương trình  $G(x) = 0$  bằng **phương pháp dây cung** (với sai số  $\varepsilon = 10^{-5}$ ), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết kiểu ngăn xếp* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 320:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào ( $N \geq 5$ ), danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N$  hàng, mỗi hàng gồm  $M$  phần tử số ( $M = N - 1$ ). Mỗi hàng này tương ứng là một danh sách liên kết).

b) Thực hiện thêm vào mỗi danh sách một phần tử, hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Sau khi thực hiện ở câu b), chúng ta nhận được  $N$  danh sách liên kết, mỗi danh sách có  $N$  phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau (mỗi danh sách tương ứng với mỗi hàng của ma trận):

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

Hãy tìm nghiệm  $X = (x_0, x_1, \dots, x_N)^T$  của hệ phương trình  $AX = B$  bằng **phương pháp lặp đơn**, với  $B = (b_1, b_2, \dots, b_N)^T$  (với  $B$  là mảng một chiều được nhập vào), sau đó hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết kiểu ngăn xếp* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

**Đề 321:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào ( $N \geq 5$ ), danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN gồm  $N$  hàng, mỗi hàng gồm  $M$  phần tử số ( $M = N + 1$ ). Mỗi hàng này tương ứng là một danh sách liên kết).

b) Thực hiện xóa bớt ở mỗi danh sách một phần tử, hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Sau khi thực hiện ở câu b), chúng ta nhận được  $N$  danh sách liên kết, mỗi danh sách có  $N$  phần tử. Biết rằng các phần tử này là các giá trị được biểu diễn theo trận sau (mỗi danh sách tương ứng với mỗi hàng của ma trận):

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

Hãy tìm nghiệm  $X = (x_0, x_1, \dots, x_N)^T$  của hệ phương trình  $AX = B$  bằng **phương pháp Gauss - Siedel**, với  $B = (b_1, b_2, \dots, b_N)^T$  (với B là mảng một chiều được nhập vào), sau đó hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*.

**Yêu cầu:** Sử dụng *danh sách liên kết kiểu ngăn xếp* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

---

**Đề 322:** : Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

a) Đọc vào  $N=3$  danh sách liên kết từ file DAYSO.IN (Gợi ý: trong file DAYSO.IN bao gồm  $N = 3$  hàng, mỗi hàng gồm M phần tử số ( $M \geq 7$ ). Mỗi hàng này tương ứng là một danh sách liên kết).

b) Thực hiện xóa hai phần tử đối với danh sách thứ nhất và thứ ba (mỗi danh sách xóa hai phần tử), xóa một phần tử đối với danh sách thứ hai, hiển thị kết quả và lưu kết quả ra file *RESULT1.OUT*.

c) Kết quả sau khi thực hiện ở câu b), chúng ta nhận được các danh sách: thứ nhất ( $a_i, i = 1 \dots M - 2$ ), thứ hai ( $b_i, i = 1 \dots M - 1$ ) và thứ ba ( $c_i, i = 1 \dots M - 2$ ). (ma trận có ba đường chéo chính- Tridiagonal matrix):

$$A = \begin{pmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 \\ a_1 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & a_2 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & a_3 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & b_{M-2} & c_{M-2} \\ 0 & 0 & 0 & 0 & a_{M-2} & b_{M-1} \end{pmatrix}$$

Hãy tìm nghiệm  $X(x_0, x_1, \dots, x_N)$  của hệ phương trình  $AX = Q$ , với  $Q = (q_1, q_2, \dots, q_N)^T$  (với Q là mảng một chiều được nhập vào), hiển thị kết quả và lưu kết quả ra file *RESULT2.OUT*

Lưu ý: Tridiagonal matrix [https://en.wikipedia.org/wiki/Tridiagonal\\_matrix](https://en.wikipedia.org/wiki/Tridiagonal_matrix)

**Yêu cầu:** Sử dụng *danh sách liên kết kiểu ngăn xếp* để thực hiện các công việc trên.

**Dữ liệu đầu vào:** file DAYSO.IN

**Dữ liệu đầu ra:** Kết quả thực hiện được lưu ở 2 file *RESULT1.OUT*, *RESULT2.OUT*.

**Tài liệu tham khảo:**

[1] Phan Chí Tùng, *Bài giảng Cấu trúc dữ liệu*, Khoa CNTT, ĐHBK-ĐHĐN, lưu hành nội bộ.

[2] Đỗ Thị Tuyết Hoa, *Bài giảng Phương pháp tính*, Khoa CNTT, ĐHBK-ĐHĐN, lưu hành nội bộ

#### Đề 401:

Bờm mới tìm được một tài liệu định nghĩa số rõ ràng như sau: Với số nguyên dương  $n$ , ta tạo số mới bằng cách lấy tổng bình phương các chữ số của nó, với số mới này ta lại lặp lại công việc trên. Nếu trong quá trình đó, ta nhận được số mới là 1, thì số  $n$  ban đầu được gọi là số rõ ràng. Ví dụ, với  $n = 19$ , ta có:

$$19 \rightarrow 82 (= 1^2 + 9^2) \rightarrow 68 \rightarrow 100 \rightarrow 1$$

Như vậy, 19 là số rõ ràng.

Không phải mọi số đều rõ ràng. Ví dụ, với  $n = 12$ , ta có:

$$12 \rightarrow 5 \rightarrow 25 \rightarrow 29 \rightarrow 85 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4 \rightarrow 16 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145$$

Bờm rất thích thú với định nghĩa số rõ ràng này và thách đố phú ông: Cho một số nguyên dương  $n$ , tìm số  $S(n)$  là số rõ ràng liền sau số  $n$ , tức là  $S(n)$  là số rõ ràng nhỏ nhất lớn hơn  $n$ . Tuy nhiên, câu hỏi đó quá dễ với phú ông và phú ông đã đố lại Bờm: Cho hai số nguyên dương  $n$  và  $k$  ( $1 \leq n, k \leq 10^{15}$ ). Hãy tìm số  $S^k(n) = S(S(\dots S(n)))$  là số rõ ràng liền sau thứ  $k$  của  $n$

Bạn hãy giúp Bờm giải câu đố này nhé!

Dữ liệu vào từ file văn bản CLEAR.INP có dạng:

- Dòng đầu là số  $t$  ( $0 < t \leq 20$ )
- $t$  dòng sau, mỗi dòng chứa 2 số nguyên  $n$  và  $k$ .

Kết quả ra file văn bản CLEAR.OUT gồm t dòng, mỗi dòng là kết quả tương ứng với dữ liệu vào.

CLEAR.INP	CLEAR.OUT
2	19
18 1	1000000000
1 145674807	

## Đề 402:

Thu thập dữ liệu comment từ các trang web.

Yêu cầu:

- Input: Địa chỉ URL một mục tin
- Output: Nội dung các comment của mục tin

Ví dụ: Với mục tin

← → ↻ [vnexpress.net/suc-khoe/so-ca-nhiem-ncov-len-233-4078685.html](https://vnexpress.net/suc-khoe/so-ca-nhiem-ncov-len-233-4078685.html)

**VN EXPRESS**  
Bảo tiếng Việt nhiều người xem nhất

Quảng cáo  
Yumangel! **STRESS VÌ COVID-19** ĐAU ĐẦU DỪNG CHỖ Y  
Độc kỷ hướng dẫn sử dụng trước khi dùng

🏠 Video Thời sự Góc nhìn Thế giới Kinh doanh Giải trí Thể thao Pháp luật Giáo dục **Sức khỏe** Đời sống Du lịch

Tin tức Các bệnh Tư vấn Khỏe đẹp Dân ông Dinh dưỡng Ung thư Vaccine

**Sức khỏe** Tin tức

Thứ sáu, 3/4/2020, 06:07 (GMT+7)

### Số ca nhiễm nCoV lên 233

6h ngày 3/4, Bộ Y tế ghi nhận thêm 6 ca nCoV, gồm một nhân viên công ty Trường Sinh, 5 người từ nước ngoài về và được cách ly ngay.

Như vậy 24 giờ qua ghi nhận 11 bệnh nhân mới, tổng cả nước 233 ca Covid-19. 11 bệnh nhân xuất viện trong ngày 2/4, đưa tổng số người khỏi bệnh lên 75. Số ca đang điều trị là 158.

Bệnh nhân mới liên quan đến Bệnh viện Bạch Mai là nữ, 57 tuổi, ở xã Kim Lư, huyện Na Rì, tỉnh Bắc Kạn, nhân viên Công ty Trường Sinh, "**bệnh nhân 231**".

Ngày 27/3, người này cùng đoàn của Bệnh viện Bạch Mai công tác tại tỉnh Hà Nam. Sau khi nhận thông tin về ca dương tính từ Bạch Mai, Trung tâm Kiểm soát Bệnh tật tỉnh Hà Nam đã cách ly người này, lấy mẫu bệnh phẩm chuyển đến Viện Vệ sinh Dịch tễ Trung ương. Kết quả xét nghiệm cho thấy bà dương tính, đang điều trị tại Bệnh viện Đa khoa tỉnh Hà Nam.

"Bệnh nhân 231" nâng tổng số ca liên quan đến Bệnh viện Bạch Mai lên 43, trong đó có 27 nhân viên công ty Trường Sinh cung cấp dịch vụ ăn uống.

Có các comment sau:

## Ý kiến (150)

Quan tâm nhất | Mới nhất

Cứ tưởng tượng ra số người ở nước ngoài về mà không được cách ly ngay thì không biết giờ này có thêm không biết bao nhiêu bệnh nhân nữa...



Nguyệt Tú Nguyễn - 4h trước

👍 1,209 Thích | [Trả lời](#) | [Chia sẻ](#) ~

[Xem thêm 4 trả lời](#)

Vẫn là Công ty Trường Sinh, hy vọng những người này chưa lây ra cộng đồng !



Chicharito80 - 4h trước

👍 453 Thích | [Trả lời](#) | [Chia sẻ](#) ~

[Xem thêm 6 trả lời](#)

Lại là công ty Trường Sinh?



Nqhuu - 4h trước

👍 259 Thích | [Trả lời](#) | [Chia sẻ](#) ~

[Xem thêm 1 trả lời](#)

Với số tăng hàng ngày như vậy cho thấy VN mình đã kiểm soát được dịch bệnh.



My Phuong - 4h trước

👍 112 Thích | [Trả lời](#) | [Chia sẻ](#) ~

[Xem thêm 2 trả lời](#)

Trường sinh ơi là Trường sinh



Dây nhựa giả mây - 4h trước

👍 109 Thích | [Trả lời](#) | [Chia sẻ](#) ~

Tôi chỉ muốn mỗi sáng mở vnexpress ra con số sẽ ko tăng .



Kim Mộc Thủy - 5h trước

👍 92 Thích | [Trả lời](#) | [Chia sẻ](#) ~

Yêu cầu lấy tất cả các comments về và lưu dưới dạng .csv

### Đề 403:

Cho một danh sách  $L$  để chứa các số nguyên. Ký hiệu  $\text{Length}(L)$  là số phần tử trong danh sách. Xét các thao tác căn bản trên danh sách:

- Phép chèn  $\text{Insert}(v, i)$ : Nếu  $1 \leq i \leq \text{Length}(L) + 1$ , thao tác này chèn một số  $v$  vào vị trí  $i$  của danh sách, nếu không thao tác này không có hiệu lực. (Trường hợp  $i = \text{Length}(L) + 1$  thì  $v$  sẽ được thêm vào cuối danh sách).
- Phép xóa  $\text{Delete}(i)$ : Nếu  $1 \leq i \leq \text{Length}(L)$ , thao tác này xóa phần tử thứ  $i$  trong danh sách, nếu không thao tác này không có hiệu lực.

Cho danh sách  $L = [ ]$  (rỗng) và  $n$  thao tác thuộc một trong hai loại, hãy in ra các phần tử theo đúng thứ tự trong danh sách cuối cùng.

### Input

- Dòng 1 chứa số nguyên dương  $n \leq 10^5$
- $n$  dòng tiếp, mỗi dòng cho thông tin về một thao tác. Mỗi dòng bắt đầu bởi một ký tự thuộc  $\{I, D\}$ . Nếu ký tự đầu dòng là “I” thì tiếp theo là hai số nguyên  $v, i$  tương ứng với phép chèn  $\text{Insert}(v,i)$ , nếu ký tự đầu dòng là D thì tiếp theo là số nguyên tương ứng với phép xóa  $\text{Delete}(i)$ . Các giá trị  $v, i$  là số nguyên Integer.

## Output

Các phần tử trong danh sách cuối cùng theo đúng thứ tự

Sample Input	Sample Output
8 I 5 1 I 6 1 I 7 1 I 8 3 I 1 2 D 4 I 9 2 D 1	9 1 6 5

## Đề 404:

Một giáo viên cần giảng  $n$  vấn đề được đánh số từ 1 đến  $n$  ( $n \leq 10000$ ). Mỗi một vấn đề  $i$  cần có thời gian là  $t_i$  ( $i = 1..n$ ). để giảng  $n$  vấn đề đó thì giáo viên có các buổi đã được phân có độ dài là  $L$  ( $L \leq 500$ ).

- Một vấn đề thì phải giải quyết trong một buổi .
- Vấn đề  $i$  phải được giảng trước vấn  $i + 1$  với mọi  $i = 1..(n-1)$ .



Học sinh có thể ra về sớm nếu như buổi giảng đã kết thúc, tuy nhiên nếu thời gian ra về đó quá sớm so với buổi giảng thì thật là phí. Chính vì thế người ta đánh giá buổi lên lớp bằng giá trị DI như sau:

$$DI = \begin{cases} 0 & \text{nếu } t = 0 \\ -C & \text{nếu } 1 \leq t \leq 10 \\ (t - 10)^2 & \text{nếu } t > 10 \end{cases}$$

Trong đó  $t$  là thời gian thừa của buổi lên lớp đó,  $C$  là một hằng số .

Yêu cầu: Hãy xếp lịch dạy sao cho tổng số các buổi là cần ít nhất có thể được. Trong các lịch dạy ít nhất đó, hãy tìm lịch dạy sao cho tổng số DI là nhỏ nhất có thể được.

Dữ liệu vào từ file SCHEDULING.INP

- Dòng đầu là số  $n$  (số vấn đề cần giảng) .
- Dòng tiếp theo là  $L$  và  $C$
- Dòng cuối cùng là  $N$  số thể hiện cho  $t_1, t_2, \dots, t_n$  .

Kết quả ra file SCHEDULING.OUT

- Dòng đầu tiên là số buổi .
- Dòng tiếp theo là tổng DI nhỏ nhất đạt được

SCHEDULING.INP	SCHEDULING.OUT
10	6
120 10	2700
80 80 10 50 30 20 40 30 120 100	

#### Đề 405: (Hái nấm)

Bé Bông đi hái nấm trong  $N$  khu rừng đánh số từ 1 đến  $N$ , nhưng chỉ có  $M$  khu rừng có nấm. Việc di chuyển từ khu rừng thứ  $i$  sang khu rừng thứ  $j$  tốn  $t_{ij}$  đơn vị thời gian. Đến

khu rừng  $i$  có nấm, cô bé có thể dừng lại để hái nấm. Nếu tổng số đơn vị thời gian cô bé dừng lại ở khu rừng thứ  $i$  là  $d_i$  ( $d_i > 0$ ), thì cô bé hái được:

$$\left\lfloor \frac{S_i}{2} \right\rfloor + \left\lfloor \frac{S_i}{4} \right\rfloor + \dots + \left\lfloor \frac{S_i}{2^{d_i}} \right\rfloor$$

cây nấm tại khu rừng đó (trong đó  $S_i$  là số lượng nấm có tại khu rừng  $i$ ,  $[x]$  là phần nguyên của  $x$ ).

Giả thiết rằng ban đầu cô bé ở khu rừng thứ nhất và đi hái nấm trong thời gian không quá  $P$  đơn vị.

Yêu cầu: Hãy tính số lượng cây nấm nhiều nhất mà cô bé có thể hái được.

Dữ liệu vào từ file văn bản MUSHROOM.INP

- Dòng đầu tiên chứa ba số nguyên dương  $M$  ( $M \leq 10$ ),  $N$  ( $0 < M \leq N \leq 100$ ) và  $P$  ( $P \leq 0000$ );
- $M$  dòng tiếp theo, mỗi dòng chứa 2 số nguyên dương  $r$  và  $S_r$  nghĩa là khu rừng  $r$  có  $S_r$  nấm ( $S_r \leq 109$ );
- Dòng thứ  $i$  trong  $N$  dòng cuối cùng chứa  $N$  số nguyên dương  $t_{ij}$  ( $t_{ij} \leq 10000$ ), ( $i, j=1, \dots, N$ ).

Kết quả ghi ra file văn bản MUSHROOM.OUT: số lượng cây nấm nhiều nhất bé Bông có thể hái được

MUSHROOM.INP	MUSHROOM.OUT
2 2 2	3
1 5	
2 10	
0 3	
3 0	

Thuật toán RSA được thiết kế dựa trên độ khó của bài toán phân tích ra thừa số nguyên tố trên tập số nguyên  $Z_n$ . Cho số nguyên dương  $n = p * q$ , với  $p, q$  là 2 số nguyên tố rất lớn. Khi biết  $n$ , muốn tìm  $p, q$  thì phải giải bài toán phân tích ra thừa số nguyên tố, công việc này đòi hỏi phải thực hiện một số lượng các phép tính vô cùng lớn.

a) Tạo khóa:

- Tạo ngẫu nhiên 2 số nguyên tố  $p, q$  khác nhau và rất lớn (có số ký tự ít nhất là 100), sau đó tính:  $n = p * q$ ;  $\Phi(n) = (p - 1) * (q - 1)$ .

- Chọn ngẫu nhiên 1 số  $e$  sao cho  $1 < e < \Phi(n)$ , với  $e$  là số nguyên tố cùng nhau với  $\Phi(n)$ .

- Tính số nghịch đảo  $d$  của  $e$  đối với  $\Phi(n)$ :  $1 < d < \Phi(n)$ ,  $ed = 1 \pmod{\Phi(n)}$ .

Ở đây  $d$  là số mũ bí mật;  $e$  là số mũ công khai.

- Cặp  $K_U = \{e, n\}$  được gọi là khoá công khai.

- Cặp  $K_R = \{d, n\}$  được gọi là khoá bí mật.

b) Mã hóa: Cho thông điệp  $M$ , khoá công khai  $K_U = \{e, n\}$ .

Văn bản mã hóa:  $C = M^e \pmod n$

c) Giải mã

Dùng khoá bí mật  $K_R = \{d, n\}$

Khôi phục văn bản gốc  $M = C^d \pmod n$ .

**Đề tài 502:** Bài toán vận tải

Ta cần vận tải vật liệu từ 2 kho:  $K1$  và  $K2$ , đến 3 công trường xây dựng:  $C1, C2, C3$ .

Tổng số vật liệu có ở mỗi kho, tổng số vật liệu yêu cầu của mỗi công trường, khoảng cách từ mỗi kho đến mỗi công trường được cho ở bảng sau:

$\begin{matrix} km \\ \backslash \\ Kho \end{matrix}$	CT	$C_1$ $15T$	$C_2$ $20T$	$C_3$ $25T$
$K_1$ $20T$	$5km$	$x_{11}$	$x_{12}$	$x_{13}$
$K_2$ $40T$	$4km$	$x_{21}$	$x_{22}$	$x_{23}$

Hãy lập kế hoạch vận chuyển thế nào để thỏa mãn 3 điều kiện sau:

- 1) Các kho giải phóng hết vật liệu.
- 2) Các công trường nhận đủ vật liệu cần thiết.
- 3) Tổng số  $T \times km$  phải thực hiện là ít nhất.

Hướng dẫn

1. Gọi  $x_{ij}$  ( $i=1,2; j=1,2,3$ ) lần lượt là số tấn vật liệu vận chuyển từ kho  $K_i \rightarrow C_j \quad \forall x_{ij} \geq 0$
2. Số tấn vật liệu từ kho K1 đến 3 công trường:  $x_{11} + x_{12} + x_{13}$
3. Số tấn vật liệu từ kho K2 đến 3 công trường:  $x_{21} + x_{22} + x_{23}$
4. Số tấn vật liệu công trường C1 nhận được từ 3 kho:  $x_{11} + x_{21}$
5. Số tấn vật liệu công trường C2 nhận được từ 3 kho:  $x_{12} + x_{22}$
6. Số tấn vật liệu công trường C3 nhận được từ 3 kho:  $x_{13} + x_{23}$
7. Tổng số  $T \times km$ :  $5x_{11} + 7x_{12} + 2x_{13} + 4x_{21} + 3x_{22} + 6x_{23}$

Mô hình toán học:

$$(1) f(x) = 5x_{11} + 7x_{12} + 2x_{13} + 4x_{21} + 3x_{22} + 6x_{23} \rightarrow \min$$

$$(2) \begin{cases} x_{11} + x_{12} + x_{13} = 20 \\ x_{21} + x_{22} + x_{23} = 40 \\ x_{11} + x_{21} = 15 \\ x_{12} + x_{22} = 20 \\ x_{13} + x_{23} = 25 \end{cases}$$

$$(3) x_{ij} \geq 0 (i = 1, 2; j = 1, 2, 3)$$

**Đề tài 503:** Bài toán dự báo doanh số bán hàng

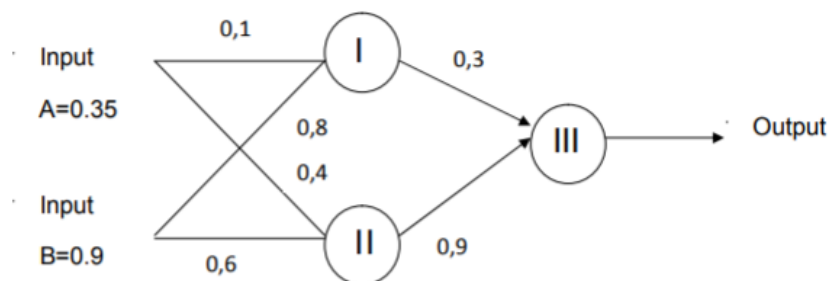
Cho dữ liệu doanh số bán hàng của thương hiệu sữa Vinamilk

TT	Tháng năm	Ngày		Doanh thu
1	201606	1		87637330000
2	201606	2		51592350000
3	201606	3		70710450000
	....	...		...

Bài toán dự báo doanh số

Để thử nghiệm mô hình dự báo và thay đổi các thông số khác nhau đối với mạng nơron nhân tạo, ta xây dựng một công cụ mô phỏng mạng nơron truyền thẳng sử dụng thuật toán lan truyền ngược sai số.

Giả sử chúng ta có một mạng Nơron như sau:



Với mạng Nơron cho trước các trọng số như hình với hàm kích hoạt là hàm sigmoid, với hệ số hiệu chỉnh bias bằng 1 và hệ số  $h = 1$ :

$$\text{Hàm sigmoid : } y = \frac{1}{1+e^{-t}}$$

$$\text{và đạo hàm của hàm sigmoid: } \frac{df(t)}{dt} = \frac{e^t}{(1+e^t)^2}$$

Mẫu gồm 3 phần:

Đầu vào:  $A = 0.35$ ,  $B = 0.9$ .

Đầu ra mong muốn:  $z = 0.5$ .

Bước 1) Trước hết chúng ta sẽ tính đầu ra cho các Nơron :

Nơron 1:  $t_1 = (0.35 \times 0.1) + (0.9 \times 0.8) + 1 = 1.755$ .

$$Y_1 = \frac{1}{1 + e^{-1.755}} = 0.8525.$$

Từ đó suy ra:

Nơron 2:  $t_2 = (0.9 \times 0.6) + (0.35 \times 0.4) + 1 = 1.68$ .

$$Y_2 = \frac{1}{1 + e^{-1.68}} = 0.842.$$

Từ đó suy ra:

Nơron 3:  $t_3 = (0.3 \times 0.8525) + (0.9 \times 0.842) + 1 = 2.0143$ .

$$Y_3 = \frac{1}{1 + e^{-2.0143}} = 0.8823.$$

Từ đó suy ra:

Bước 2) Bước tiếp theo là chúng ta sẽ tính tín hiệu lỗi.

$$d_3 = z - y = 0.5 - 0.8823 = -0.3823.$$

Lan truyền tín hiệu lỗi:

Nơron 1:  $d_1 = -0.3823 \times 0.3 = -0.11469$ .

Nơron 2 :  $d_2 = -0.3823 \times 0.9 = -0.344$

Bước 3) Bây giờ chúng ta sẽ dựa vào tín hiệu lỗi của mỗi Nơron để thay đổi trọng số weights.

$$W'_{A1} = W_{A1} + \eta \delta_1 \frac{df(t)}{dt} x_1 = 0.1 + 1(-0.11469) \frac{e^{1.755}}{(1+e^{1.755})^2} \cdot 0.35 = 0.089.$$

$$W'_{B1} = W_{B1} + \eta \delta_1 \frac{df(t)}{dt} x_1 = 0.8 + 1(-0.11469) \frac{e^{1.755}}{(1+e^{1.755})^2} \cdot 0.35 = 0.789.$$

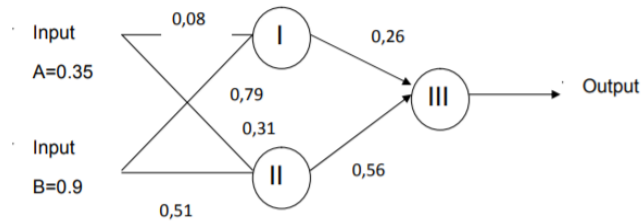
$$W'_{A2} = W_{A2} + \eta \delta_2 \frac{df(t)}{dt} x_2 = 0.4 + 1(-0.344) \frac{e^{1.68}}{(1+e^{1.68})^2} \cdot 0.9 = 0.3067.$$

$$W'_{B2} = W_{B2} + \eta \delta_2 \frac{df(t)}{dt} x_2 = 0.6 + 1(-0.344) \frac{e^{1.68}}{(1+e^{1.68})^2} \cdot 0.9 = 0.5067.$$

$$W'_{13} = W_{13} + \eta \delta_3 \frac{df(t)}{dt} y_1 = 0.3 + 1(-0.3823) \frac{e^{2.0143}}{(1+e^{2.0143})^2} \cdot 0.8525 = 0.258.$$

$$W'_{23} = W_{23} + \eta \delta_3 \frac{df(t)}{dt} y_2 = 0.6 + 1(-0.3823) \frac{e^{2.0143}}{(1+e^{2.0143})^2} \cdot 0.842 = 0.5589.$$

Bước 4) Cho ra kết quả:



Tài liệu tham khảo:

- [1] <https://towardsdatascience.com/simple-neural-network-implementation-in-c-663f51447547>
- [2] <https://medium.com/analytics-vidhya/building-neural-network-framework-in-c-using-backpropagation-8ad589a0752d>
- [3] <https://codereview.stackexchange.com/questions/191498/simple-neural-network-in-c>
- [4] <https://courses.cs.washington.edu/courses/cse599/01wi/admin/Assignments/bpn.html>
- [5] <https://www.codeproject.com/Articles/1237026/Simple-MLP-Backpropagation-Artificial-Neural-Netwo>
- [6] <https://takinginitiative.wordpress.com/2008/04/23/basic-neural-network-tutorial-c-implementation-and-source-code/>

**Đề tài 504:** Xây dựng và tính toán tập hợp điểm của đường cong tham số B-spline đồng nhất

*Mô tả:* Đường cong B-spline đồng nhất (*uniform*) là một đường cong tham số thường được sử dụng trong đồ họa máy tính, tính toán nội suy..... Một đường cong B-spline được xác định bằng tập hợp  $(n + 1)$  đỉnh điều khiển  $P_0(x_0, y_0, z_0)$  đến  $P_n(x_n, y_n, z_n)$ , với  $p$  là bậc của đường cong. Phương trình đường cong B-spline uniform bậc  $p$ :

$$C(u) = \sum_{i=0}^n N_{i,p}(u) \cdot P_i$$

Giả sử  $u_i = i$ , vector nút đồng nhất  $U = [0, 1, 2, \dots, n + p + 1]$ .

Các hàm cơ sở B-spline  $N_{i,p}(u)$  thứ  $i$  bậc  $p$  được xác định trong các khoảng giá trị của vector nút  $U$  như sau :

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, & i \leq u \leq i+1 \\ 0, & \text{khác} \end{cases} \\ N_{i,k}(u) = \frac{u-i}{k} N_{i,k-1}(u) + \frac{i+k+1-u}{k} N_{i+1,k-1}(u-1) \end{cases}$$

với  $i = 0 \dots n+p-k, k = 1 \dots p$ .

*Thuật toán:* Để xác định một điểm trên đường cong B-spline ứng với tham số  $u$ , ta triển khai tính toán đệ quy các đa thức  $N_{i,p}(u)$  và đa thức  $C(u)$ . Sử dụng cấu trúc dữ liệu mảng động để lưu trữ và tính toán xử lý.

*Input:* File data.txt mô tả bậc của đường cong và tọa độ các đỉnh điều khiển  $P_i$

```
n = 3 //Bậc của đường cong
P0 = (10, 0, 0) //Tọa độ đỉnh thứ nhất
P1 = (20, 30, 0)
P2 = (50, 40, 0)
P3 = (80, 0, 0)
```

Tập  $m$  tham số  $\{u_i\} = \{0, 1, 2.5, 3.5, 4, 5, 6.5, 8\}$

*Output:* Tọa độ  $m$  điểm  $C(u_i) = (x, y, z)$  trên đường cong tương ứng với các tham số  $u_i$  của đường cong B-spline đồng nhất.

*Tài liệu tham khảo:*

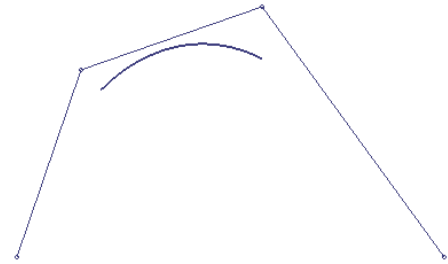
- [1] Tài liệu/slide về đường cong tham số do GV cung cấp.
- [2]Advanced Graphics and HCI - B-Splines, Dr Neil Dodgson, University of Cambridge Computer Laboratory,  
<https://www.cl.cam.ac.uk/teaching/1999/AGraphHCI/SMAG/node4.html>, 23/10/2019.
- [3]Uniform Cubic B-Spline Curves, Howard J. Hamilton, University of Regina,  
<http://www2.cs.uregina.ca/~anima/408/Notes/Interpolation/UniformBSpline.htm> ,  
23/10/2019.
- [4] <http://nurbscalculator.in/>, 21/11/2019.

**Nội dung hướng dẫn:**



1) Tìm hiểu đường cong B-spline đồng nhất: Định điều khiển, bậc, phương trình, vector nút đồng nhất, tính chất, ứng dụng.

2) Triển khai ví dụ tính bằng tay cụ thể tọa độ điểm  $C(u)$  với giá trị tham số  $u$  tự chọn trong khoảng  $[0, 1]$  của đường cong B-spline đồng nhất bậc  $p = 3$ , có  $n + 1 = 4$  đỉnh  $P_0(10, 0)$ ,  $P_1(20, 30)$ ,  $P_2(50, 40)$ ,  $P_3(80, 0)$ ; véc tơ nút  $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$ . Biểu diễn hình học của đường cong và khung điều khiển được mô tả như hình sau.



- a) Tính theo cách sử dụng phương trình
- b) Tính theo cách sử dụng ma trận
- 3) Xây dựng thuật toán. Đánh giá độ phức tạp
- 4) Lập trình triển khai thuật toán
- 5) Đánh giá kết quả

**Đề tài 505:** Xây dựng và tính toán tập hợp điểm của đường cong tham số B-spline không đồng nhất

*Mô tả:* Đường cong B-spline không đồng nhất (*non-uniform*) là một đường cong tham số thường được sử dụng trong đồ họa máy tính, tính toán nội suy..... Một đường cong B-spline được xác định bằng tập hợp  $(n+1)$  điểm điều khiển  $P_0(x_0, y_0, z_0)$  đến  $P_n(x_n, y_n, z_n)$ , với  $p$  được gọi bậc của nó. Phương trình đường cong B-spline non-uniform bậc  $p$ :

$$C(u) = \sum_{i=0}^{n-1} N_{i,p}(u) \cdot P_i$$

Tập vector nút  $U = \{u_0, \dots, u_{n+p}\} \forall u_i \leq u_{i+1}$  với  $u_i \in \mathbb{R}$  tập các số thực

Các hàm cơ sở B-spline  $N_{i,p}(u)$  thứ  $i$  bậc  $p$  được xác định trong các khoảng giá trị của vector nút  $U$  như sau :

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u), u_i \leq u < u_{i+k+1} \end{cases}$$

**Thuật toán:** Để xác định một điểm trên đường cong B-spline ứng với tham số  $u$ , ta triển khai tính toán đệ quy các đa thức  $N_{i,p}(u)$  và đa thức  $C(u)$ . Sử dụng cấu trúc dữ liệu mảng động để lưu trữ và tính toán xử lý.

**Input:** File *data.txt* mô tả bậc của đường cong và tọa độ các đỉnh điều khiển  $P_i$

```
n = 3 //Bậc của đường cong
P0 = (10, 0, 0) //Tọa độ đỉnh thứ nhất
P1 = (20, 30, 0)
P2 = (50, 40, 0)
P3 = (80, 0, 0)
Tập m tham số {u_i} = { 0.0, 0.1, 0.2, 0.4, 0.64, 0.82; 0.95, 1.0}
```

**Output:** Tọa độ  $m$  điểm  $C(u_i) = (x, y, z)$  trên đường cong tương ứng với các tham số  $0 \leq u_i \leq 1$  của đường cong B-spline không đồng nhất.

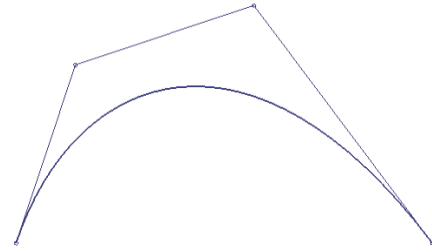
**Tài liệu tham khảo:**

- [1] Tài liệu/slide về đường cong tham số do GV cung cấp.
- [2]Advanced Graphics and HCI - B-Splines, Dr Neil Dodgson, University of Cambridge Computer Laboratory,  
<https://www.cl.cam.ac.uk/teaching/1999/AGraphHCI/SMAG/node4.html>, 23/10/2019.
- [3]Uniform Cubic B-Spline Curves, Howard J. Hamilton, University of Regina,  
<http://www2.cs.uregina.ca/~anima/408/Notes/Interpolation/UniformBSpline.htm> ,  
23/10/2019.
- [4] <http://nurbscalculator.in/>, 21/11/2019.

**Nội dung hướng dẫn :**

- 1) Tìm hiểu đường cong B-spline không đồng nhất: Định điều khiển, bậc, phương trình, vector nút đồng nhất, tính chất, ứng dụng.

2) Triển khai ví dụ tính bằng tay cụ thể tọa độ điểm  $C(u)$  với tham số  $u$  tự chọn trong khoảng  $[0, 1]$  của một đường cong B-spline không đồng nhất bậc  $p = 2$ , có  $n + 1 = 4$  đỉnh  $P_0(10, 0)$ ,  $P_1(20, 30)$ ,  $P_2(50, 40)$ ,  $P_3(80, 0)$ ; véc tơ nút  $U = \{0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0\}$ . Biểu diễn hình học của đường cong và khung điều khiển được mô tả như hình sau.



- a) Tính theo cách sử dụng phương trình
- b) Tính theo cách sử dụng ma trận
- 3) Xây dựng thuật toán. Đánh giá độ phức tạp
- 4) Lập trình triển khai thuật toán
- 5) Đánh giá kết quả

### **Đề tài 506:** Nội suy đường cong Bézier

*Mô tả:* Đường cong Bézier là một đường cong tham số thường được sử dụng trong đồ họa máy tính, tính toán nội suy.... Một đường cong Bézier  $C(u)$  được xác định bằng tập hợp các điểm điều khiển  $P_0(x_0, y_0, z_0)$  đến  $P_n(x_n, y_n, z_n)$ , với  $n$  được gọi bậc của đường cong Bézier. Điểm kiểm soát đầu và cuối là các điểm nút của đường cong. Phương trình biểu diễn đường cong Bézier bậc  $n$ :

$$C(u) = \sum_{i=0}^n P_i B_{i,n}(u) \quad \text{với } 0 \leq u \leq 1.$$

Các đỉnh điều khiển  $P_i$  có tọa độ:  $P_i(x_i, y_i, z_i) \quad \forall x_i, y_i, z_i \in \mathbb{R}, \quad i = 0 \dots n.$

$B_{i,n}(u)$  là đa thức Bernstein (*Bernstein polynomial*) có bậc  $n$ :

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad i = 0 \dots n$$

*Input:* File dữ liệu *data.txt* chứa  $m$  tập điểm  $D_i(x_i, y_i, z_i)$  mà ta sẽ xây dựng một đường cong Bézier nội suy đi qua các điểm  $D_i$  này.

*Output:* Tập  $(n + 1)$  đỉnh điều khiển  $P_0(x_0, y_0, z_0)$  đến  $P_n(x_n, y_n, z_n)$  của một đường cong Bézier bậc  $n$  nội suy

*Yêu cầu:* Sử dụng các cấu trúc dữ liệu (mảng, danh sách, ...) và thuật toán để tính toán xử lý.

*Tài liệu tham khảo:*

- [1] Tài liệu/slide về đường cong tham số do GV cung cấp.

[2]Advanced Graphics and HCI - B-Splines, Dr Neil Dodgson, University of Cambridge Computer Laboratory,  
<https://www.cl.cam.ac.uk/teaching/1999/AGraphHCI/SMAG/node4.html>, 23/10/2019.

[3]Uniform Cubic B-Spline Curves, Howard J. Hamilton, University of Regina,  
<http://www2.cs.uregina.ca/~anima/408/Notes/Interpolation/UniformBSpline.htm> ,  
23/10/2019.

[4] <http://nurbscalculator.in/>, 21/11/2019.

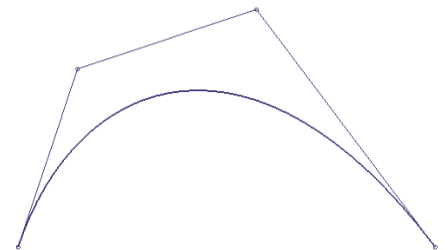
[5] Đỗ Thị Tuyết Hoa, Bài giảng môn Phương pháp tính, 2007

[6] Duy Thúc Trần dịch, Giải hệ phương trình tuyến tính,  
[https://matrixcalc.org/vi/slu.html?fbclid=IwAR1rsZgsHHY2bMdfQp0ccXNLNqV2r0kjjB-HFqB\\_Q11JuAFCE-dnKC\\_AlC8#solve-using-Gaussian-elimination%28%7B%7B4,7,21,73,4%7D,%7B7,21,73,273,1%7D,%7B21,73,273,1057,-1%7D,%7B73,273,1057,4161,-5%7D%7D%29](https://matrixcalc.org/vi/slu.html?fbclid=IwAR1rsZgsHHY2bMdfQp0ccXNLNqV2r0kjjB-HFqB_Q11JuAFCE-dnKC_AlC8#solve-using-Gaussian-elimination%28%7B%7B4,7,21,73,4%7D,%7B7,21,73,273,1%7D,%7B21,73,273,1057,-1%7D,%7B73,273,1057,4161,-5%7D%7D%29) , 04/12/2019 08:12

### Nội dung hướng dẫn :

1) Tìm hiểu đường cong Bézier: Định điều khiển, bậc, phương trình, vectơ nút đồng nhất, tính chất, ứng dụng.

2) Triển khai ví dụ tính bằng tay cụ thể tọa độ điểm  $C(u)$  (với giá trị tham số  $u$  tự chọn trong khoảng  $[0, 1]$ ) của đường cong Bézier bậc  $p = 3$ , có  $n + 1 = 4$  đỉnh  $P_0(10, 0)$ ,  $P_1(20, 30)$ ,  $P_2(50, 40)$ ,  $P_3(80, 0)$ ; Biểu diễn hình học của đường cong và khung điều khiển được mô tả như hình sau.



a) Tính theo cách sử dụng phương trình

b) Tính theo cách sử dụng ma trận

3) Xây dựng thuật toán. Đánh giá độ phức tạp

4) Lập trình triển khai thuật toán

5) Đánh giá kết quả

**Đề tài 507:** Nội suy đường cong B-spline đồng nhất

*Mô tả:* Đường cong B-spline đồng nhất (*uniform*) là một đường cong tham số thường được sử dụng trong đồ họa máy tính, tính toán nội suy..... Một đường cong B-spline đồng nhất được xác định bằng tập hợp  $(n + 1)$  đỉnh điều khiển  $P_0(x_0, y_0, z_0)$  đến  $P_n(x_n, y_n, z_n)$ , với  $p$  là bậc của đường cong. Phương trình đường cong B-spline đồng nhất bậc  $p$ :

$$C(u) = \sum_{i=0}^n N_{i,p}(u) \cdot P_i$$

Giả sử  $u_i = i$ , vector nút đồng nhất  $U = [0, 1, 2, \dots, n + p + 1]$ .

Các hàm cơ sở B-spline  $N_{i,p}(u)$  thứ  $i$  bậc  $p$  được xác định trong các khoảng giá trị của vector nút  $U$  như sau :

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, & i \leq u \leq i+1 \\ 0, & \text{khác} \end{cases} \\ N_{i,k}(u) = \frac{u-i}{k} N_{i,k-1}(u) + \frac{i+k+1-u}{k} N_{i+1,k-1}(u-1) \end{cases}$$

với  $i = 0 \dots n + p - k, k = 1 \dots p$ .

*Input:* File dữ liệu *data.txt* chứa  $m$  tập điểm  $D_i(x_i, y_i, z_i)$  mà ta sẽ xây dựng một đường cong B-spline đồng nhất nội suy đi qua các điểm  $D_i$  này.

*Output:* Tập  $(n + 1)$  đỉnh điều khiển  $P_0(x_0, y_0, z_0)$  đến  $P_n(x_n, y_n, z_n)$  của một đường cong B-spline đồng nhất bậc  $p$  nội suy.

*Yêu cầu:* Sử dụng các cấu trúc dữ liệu (mảng, danh sách, ...) để tính toán xử lý.

*Tài liệu tham khảo:*

[1] Tài liệu/slide về đường cong tham số do GV cung cấp.

[2] Advanced Graphics and HCI - B-Splines, Dr Neil Dodgson, University of Cambridge Computer Laboratory,  
<https://www.cl.cam.ac.uk/teaching/1999/AGraphHCI/SMAG/node4.html>, 23/10/2019.

[3] Uniform Cubic B-Spline Curves, Howard J. Hamilton, University of Regina,  
<http://www2.cs.uregina.ca/~anima/408/Notes/Interpolation/UniformBSpline.htm> ,  
23/10/2019.

[4] <http://nurbscalculator.in/>, 21/11/2019.

[5] Đỗ Thị Tuyết Hoa, Bài giảng môn Phương pháp tính, 2007

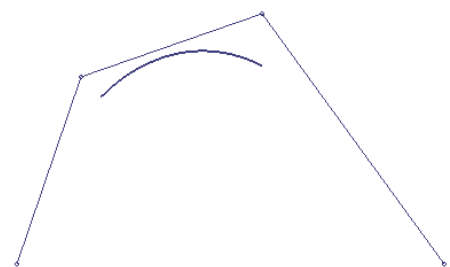
[6] Duy Thúc Trần dịch, Giải hệ phương trình tuyến tính,

[https://matrixcalc.org/vi/slu.html?fbclid=IwAR1rsZgsHHY2bMdfQp0ccXNLNqV2r0kkyB-HFqB\\_Q11JuAFCE-dnKC\\_AlC8#solve-using-Gaussian-elimination%28%7B%7B4,7,21,73,4%7D,%7B7,21,73,273,1%7D,%7B21,73,273,1057,-1%7D,%7B73,273,1057,4161,-5%7D%7D%29](https://matrixcalc.org/vi/slu.html?fbclid=IwAR1rsZgsHHY2bMdfQp0ccXNLNqV2r0kkyB-HFqB_Q11JuAFCE-dnKC_AlC8#solve-using-Gaussian-elimination%28%7B%7B4,7,21,73,4%7D,%7B7,21,73,273,1%7D,%7B21,73,273,1057,-1%7D,%7B73,273,1057,4161,-5%7D%7D%29) , 04/12/2019 08:12

### Nội dung hướng dẫn :

1) Tìm hiểu đường cong B-spline đồng nhất: Định điều khiển, bậc, phương trình, vector nút đồng nhất, tính chất, ứng dụng.

2) Triển khai ví dụ tính bằng tay cụ thể tọa độ điểm  $C(u)$  (với giá trị tham số  $u$  tự chọn trong khoảng  $[u_0, u_{n+p}] = [0, \dots, n+p+1]$  của đường cong B-spline đồng nhất bậc  $p = 3$ , có  $n + 1 = 4$  đỉnh  $P_0(10, 0)$ ,  $P_1(20, 30)$ ,  $P_2(50, 40)$ ,  $P_3(80, 0)$ ; vectơ nút  $U = \{0, 1, 2, 3, 4, 5, 6, 7\}$ . Biểu diễn hình học của đường cong và khung điều khiển được mô tả như hình sau.



a) Tính theo cách sử dụng phương trình

3) Xây dựng thuật toán. Đánh giá độ phức tạp

4) Lập trình triển khai thuật toán

5) Đánh giá kết quả

**Đề tài 508:** Giải hệ phương trình tuyến tính dựa vào phân rã LU

Mô tả: Cho hệ phương trình tuyến tính tổng quát  $A.X = B$ . Ta tiến hành phân rã  $A = L.U$ . Trong đó,  $L$  là ma trận tam giác dưới và  $U$  là ma trận tam giác trên. Khi đó,

$$A.X = B \Leftrightarrow L.U.X = B \Leftrightarrow \begin{cases} L.Y = B \\ U.X = Y \end{cases}$$

Quá trình chuyển hoá ma trận  $A$  ban đầu thành tích hai ma trận tam giác  $L.U$  dựa vào phép khử Gauss được thực hiện bằng các phép nhân ma trận. Thuật toán này được gọi là thuật toán Crout. Quá trình Crout bao gồm nhiều bước hồi quy. Nếu ma trận  $A$  có cấp  $n \times n$  ta cần  $n - 1$  bước. Thuật toán được thể hiện cụ thể như sau:

$$\begin{aligned}
A_n &= \left( \begin{array}{c|ccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right) \\
&= \begin{pmatrix} a_{11} & \mathbf{w}^T \\ \mathbf{v} & \mathbf{H} \end{pmatrix} \frac{Crout}{1} \begin{pmatrix} 1 & 0 \\ \mathbf{v}/a_{11} & \mathbf{I}_{n-1} \end{pmatrix} \begin{pmatrix} a_{11} & \mathbf{w}^T \\ 0 & \underbrace{\mathbf{H} - \mathbf{v}\mathbf{w}^T/a_{11}}_{\mathbf{A}_{n-1}} \end{pmatrix} = \mathbf{L}_n \mathbf{U}_n \\
&= \begin{pmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l_1 & \mathbf{I}_{n-1} \end{pmatrix} \begin{pmatrix} u_{11} & u_1^T \\ 0 & \mathbf{A}_{n-1} \end{pmatrix}
\end{aligned}$$

---


$$\begin{aligned}
&\frac{Crout}{2} \begin{pmatrix} 1 & 0 \\ l_1 & \mathbf{I}_{n-1} \end{pmatrix} \begin{pmatrix} u_{11} & u_1^T \\ 0 & \mathbf{L}_{n-1} \mathbf{U}_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l_1 & \mathbf{L}_{n-1} \end{pmatrix} \begin{pmatrix} u_{11} & u_1^T \\ 0 & \mathbf{U}_{n-1} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & & & \\ \dots & \dots & \mathbf{I}_{n-2} & & \\ l_{n1} & l_{n2} & & & \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ 0 & 0 & & \\ \dots & \dots & \mathbf{A}_{n-2} & \\ 0 & 0 & & \end{pmatrix} \xrightarrow{Crout \dots} \\
&\xrightarrow[Crout]{n-1} \begin{pmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & u_{nn} \end{pmatrix} = \mathbf{L} \cdot \mathbf{U}
\end{aligned}$$

**Đề tài 509:** Nội suy đường cong B-spline không đồng nhất

**Mô tả:** Đường cong B-spline không đồng nhất (*non-uniform*) là một đường cong tham số thường được sử dụng trong đồ họa máy tính, tính toán nội suy..... Một đường cong B-spline  $C(u)$  được xác định bằng tập hợp  $(n+1)$  điểm điều khiển  $P_0(x_0, y_0, z_0)$  đến  $P_n(x_n, y_n, z_n)$ , với  $p$  được gọi bậc của nó. Phương trình đường cong B-spline không đồng nhất bậc  $p$ :

$$C(u) = \sum_{i=0}^{n-1} N_{i,p}(u) \cdot P_i$$

Tập vectơ nút  $U = \{ u_0, \dots, u_{n+p} \} \forall u_i \leq u_{i+1}$  với  $u_i \in \mathbb{R}$  tập các số thực

Các hàm cơ sở B-spline  $N_{i,p}(u)$  thứ  $i$  bậc  $p$  được xác định trong các khoảng giá trị của vector nút  $U$  như sau :

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 0 & \text{trường hợp khác} \end{cases} \\ N_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u), u_i \leq u < u_{i+k+1} \end{cases}$$

*Input:* File dữ liệu *data.txt* chứa  $m$  tập điểm  $D_i(x_i, y_i, z_i)$  mà ta sẽ xây dựng một đường cong B-spline không đồng nhất bậc  $p$  nội suy đi qua các điểm  $D_i$  này.

*Output:* Tập  $(n + 1)$  đỉnh điều khiển  $P_0(x_0, y_0, z_0)$  đến  $P_n(x_n, y_n, z_n)$  của một đường cong đường cong B-spline non-uniform bậc  $p$  nội suy.

*Yêu cầu:* Sử dụng các cấu trúc dữ liệu (mảng, danh sách, ...) để tính toán xử lý.

*Tài liệu tham khảo:*

[1] Tài liệu/slide về đường cong tham số do GV cung cấp.

[2] Advanced Graphics and HCI - B-Splines, Dr Neil Dodgson, University of Cambridge Computer Laboratory,  
<https://www.cl.cam.ac.uk/teaching/1999/AGraphHCI/SMAG/node4.html>, 23/10/2019.

[3] Uniform Cubic B-Spline Curves, Howard J. Hamilton, University of Regina,  
<http://www2.cs.uregina.ca/~anima/408/Notes/Interpolation/UniformBSpline.htm>,  
 23/10/2019.

[4] <http://nurbscalculator.in/>, 21/11/2019.

[5] Đỗ Thị Tuyết Hoa, Bài giảng môn Phương pháp tính, 2007

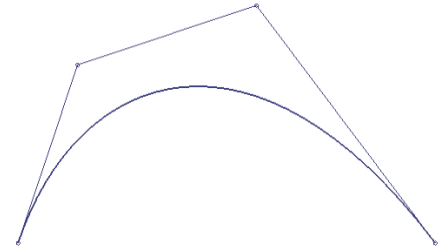
[6] Duy Thúc Trần dịch, Giải hệ phương trình tuyến tính,  
[https://matrixcalc.org/vi/slu.html?fbclid=IwAR1rsZgsHHY2bMdfQp0ccXNLNqV2r0kjbB-HFqB\\_Q11JuAFCE-dnKC\\_AlC8#solve-using-Gaussian-elimination%28%7B%7B4,7,21,73,4%7D,%7B7,21,73,273,1%7D,%7B21,73,273,1057,-1%7D,%7B73,273,1057,4161,-5%7D%7D%29](https://matrixcalc.org/vi/slu.html?fbclid=IwAR1rsZgsHHY2bMdfQp0ccXNLNqV2r0kjbB-HFqB_Q11JuAFCE-dnKC_AlC8#solve-using-Gaussian-elimination%28%7B%7B4,7,21,73,4%7D,%7B7,21,73,273,1%7D,%7B21,73,273,1057,-1%7D,%7B73,273,1057,4161,-5%7D%7D%29), 04/12/2019 08:12

### **Nội dung hướng dẫn :**

1) Tìm hiểu đường cong B-spline không đồng nhất: Định điều khiển, bậc, phương trình, vector nút đồng nhất, tính chất, ứng dụng.



2) Triển khai ví dụ tính bằng tay cụ thể tọa độ điểm  $C(u)$  (với giá trị tham số  $u$  tự chọn trong khoảng  $[u_0, u_{n+p}]$ ) của đường cong B-spline bậc  $p = 2$ , có  $n + 1 = 4$  đỉnh  $P_0(10, 0)$ ,  $P_1(20, 30)$ ,  $P_2(50, 40)$ ,  $P_3(80, 0)$ ; vectơ nút  $U = \{0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0\}$ . Biểu diễn hình học của đường cong và khung điều khiển được mô tả như hình sau.



- a) Tính theo cách sử dụng phương trình
- 3) Xây dựng thuật toán. Đánh giá độ phức tạp
- 4) Lập trình triển khai thuật toán
- 5) Đánh giá kết quả

**Đề tài 510:** Xấp xỉ đường cong B-spline không đồng nhất sử dụng phương pháp bình phương tối thiểu

*Mô tả:* Đường cong B-spline không đồng nhất (*non-uniform*) là một đường cong tham số thường được sử dụng trong đồ họa máy tính, tính toán nội suy..... Một đường cong B-spline được xác định bằng tập hợp  $(n+1)$  điểm điều khiển  $P_0(x_0, y_0, z_0)$  đến  $P_n(x_n, y_n, z_n)$ , với  $p$  được gọi bậc của nó. Phương trình biểu diễn đường cong B-spline không đồng nhất bậc  $p$ :

$$C(u) = \sum_{i=0}^{n-1} N_{i,p}(u) \cdot P_i$$

Tập vectơ nút  $U = \{u_0, \dots, u_{n+p}\} \forall u_i \leq u_{i+1}$  với  $u_i \in \mathbb{R}$  tập các số thực.

Các hàm cơ sở B-spline  $N_{i,p}(u)$  thứ  $i$  bậc  $p$  được xác định trong các khoảng giá trị của vectơ nút  $U$  như sau :

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 0 & \text{trường hợp khác} \end{cases} \\ N_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u), u_i \leq u < u_{i+k+1} \end{cases}$$

*Input:* File dữ liệu *data.txt* chứa  $m$  tập điểm  $D_i(x_i, y_i, z_i)$  mà ta sẽ xây dựng một đường cong B-spline không đồng nhất bậc  $p$  nội suy đi qua các điểm  $D_i$  này.

*Output:* Tập  $(n + 1)$  đỉnh điều khiển  $P_0(x_0, y_0, z_0)$  đến  $P_n(x_n, y_n, z_n)$  của một đường cong đường cong B-spline non-uniform bậc  $p$  nội suy.

*Yêu cầu:* Sử dụng các cấu trúc dữ liệu (mảng, danh sách, ...) để tính toán xử lý.

*Tài liệu tham khảo:*

[1] Tài liệu/slide về đường cong tham số do GV cung cấp.

[2] Advanced Graphics and HCI - B-Splines, Dr Neil Dodgson, University of Cambridge Computer Laboratory,  
<https://www.cl.cam.ac.uk/teaching/1999/AGraphHCI/SMAG/node4.html>, 23/10/2019.

[3] Uniform Cubic B-Spline Curves, Howard J. Hamilton, University of Regina,  
<http://www2.cs.uregina.ca/~anima/408/Notes/Interpolation/UniformBSpline.htm> ,  
23/10/2019.

[4] <http://nurbscalculator.in/>, 21/11/2019.

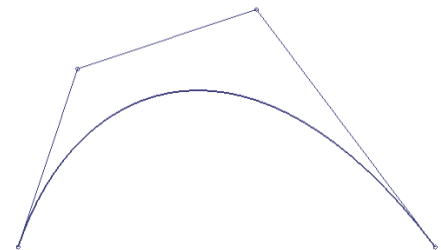
[5] Đỗ Thị Tuyết Hoa, Bài giảng môn Phương pháp tính, 2007

[6] Duy Thúc Trần dịch, Giải hệ phương trình tuyến tính,  
[https://matrixcalc.org/vi/slu.html?fbclid=IwAR1rsZgsHHY2bMdfQp0ccXNLNqV2r0kkyB-HFqB\\_Q11JuAFCE-dnKC\\_AlC8#solve-using-Gaussian-elimination%28%7B%7B4,7,21,73,4%7D,%7B7,21,73,273,1%7D,%7B21,73,273,1057,-1%7D,%7B73,273,1057,4161,-5%7D%7D%29](https://matrixcalc.org/vi/slu.html?fbclid=IwAR1rsZgsHHY2bMdfQp0ccXNLNqV2r0kkyB-HFqB_Q11JuAFCE-dnKC_AlC8#solve-using-Gaussian-elimination%28%7B%7B4,7,21,73,4%7D,%7B7,21,73,273,1%7D,%7B21,73,273,1057,-1%7D,%7B73,273,1057,4161,-5%7D%7D%29) , 04/12/2019 08:12

### Nội dung hướng dẫn :

1) Tìm hiểu đường cong B-spline không đồng nhất: Định điều khiển, bậc, phương trình, vector nút đồng nhất, tính chất, ứng dụng.

2) Triển khai ví dụ tính bằng tay cụ thể tọa độ điểm  $C(u)$  (với giá trị tham số  $u$  tự chọn trong khoảng  $[u_0, u_{n+p}]$ ) của đường cong B-spline đồng nhất bậc  $p = 2$ , có  $n + 1 = 4$  đỉnh  $P_0(10, 0)$ ,  $P_1(20, 30)$ ,  $P_2(50, 40)$ ,  $P_3(80, 0)$ ; vector nút  $U = \{0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0\}$ . Biểu diễn hình học của đường cong và khung điều khiển được mô tả như hình sau.



a) Tính theo cách sử dụng phương trình

3) Xây dựng thuật toán. Đánh giá độ phức tạp

4) Lập trình triển khai thuật toán

## 5) Đánh giá kết quả

### **Đề 601:**

Xây dựng thư viện có các thao tác của một danh sách liên kết. Từ đó viết chương trình mô phỏng hoạt động của danh sách liên kết.

**Đề 602:** Xây dựng thư viện có các thao tác của một cây nhị phân tìm kiếm. Từ đó viết chương trình mô phỏng hoạt động của cây nhị phân tìm kiếm.

**Đề 603:** Xây dựng thư viện có các thao tác của một danh sách liên kết vòng. Từ đó viết chương trình mô phỏng hoạt động của một danh sách liên kết vòng.

**Đề 604:** Cho một hàng sỏi có 3 loại (màu đỏ, màu trắng và màu vàng) xếp lộn xộn. Tìm giải pháp để xếp các sỏi theo thứ tự đỏ, vàng, trắng. Không sử dụng mảng phụ, mô phỏng bằng trực quan.

**Đề 605:** Cho một mảng có kích thước  $(2n+1) \times (2n+1)$  với  $n \geq 1$ ,  $n$  nguyên. Tìm giải thuật và viết chương trình để điền các số từ 1 đến  $(2n+1) \times (2n+1)$  vào mảng, sao cho mỗi số xuất hiện 1 lần và tổng các số trên cùng 1 hàng = tổng các số trên cùng một cột = tổng các số trên 1 đường chéo.

**Đề 606:** Trong bài toán Josephus, một nhóm có  $m$  binh sĩ bị kẻ thù bao vây và một binh sĩ được chọn để đi cầu cứu. Việc chọn thực hiện như sau: Một số nguyên  $n$  và một binh sĩ được chọn một cách ngẫu nhiên. Các binh sĩ được sắp theo vòng tròn, và họ đếm bắt đầu từ binh sĩ được chọn ngẫu nhiên. Khi đạt đến  $n$ , binh sĩ tương ứng này được lấy ra khỏi vòng và việc đếm lại bắt đầu từ binh sĩ tiếp theo. Quá trình này cứ tiếp tục cho đến khi chỉ còn một binh sĩ. Đó là người sẽ được chọn để đi cầu cứu. Cần cài đặt cách chọn này và tìm ra binh sĩ sẽ được chọn

Đầu vào:  $m, n$  ( $n > 1$ )

Đầu ra: Binh sĩ đc chọn

**Đề 607:** Hai cây nhị phân được coi là tương tự nhau nếu nó giống hệt nhau về hình dạng cây. Xây dựng hệ thống chương trình kiểm tra xem hai cây  $T1$  và  $T2$  có phải là tương tự nhau không ?

**Đề 608:** Cho số nguyên dương  $a$  có  $n$  chữ số, số nguyên dương  $b$  có  $m$  chữ số ở hệ thập phân, Tính giá trị các biểu thức  $(a+b)$ ,  $|a-b|$ ,  $a*b$ . Giả sử  $a, b$  có số chữ số vượt kích thước cho phép của NNLT.

**Đề 609:** Xây dựng dịch vụ làm mai qua máy tính: Nhập họ tên, giới tính, các sở thích, rồi tìm một hay nhiều người khác trong tệp chứa các bản ghi gồm những mục tin nói trên có nhiều sở thích chung nhất và khác giới.

**Đề 610:** Tam giác Pascal:

Ví dụ: một phần của tam giác này :

```
1
1 1
1 2 1
1 3 3 1
.....
```

Lập chương trình và thuật toán để:

- a) In ra dòng thứ n của bảng này ( chú ý khi n khá lớn !)
- b) In ra tam giác Pascal có n hàng

**Đề 701:**

**Xây dựng ứng dụng thực hiện giao dịch trên máy ATM.**

**Mục tiêu:**

- Sử dụng các mảng song song để lưu dữ liệu.
- Viết các hàm để xây dựng các chức năng.
- Đọc và Ghi dữ liệu vào file.
- Nhiều vấn đề cần giải quyết với các biến, vòng lặp và ra quyết định.

**Yêu cầu:**

Viết chương trình C/C++ để thực hiện giao dịch trên máy ATM.

**Mô tả chung về đề tài:**

- Thực hiện giao dịch trên máy ATM. Mỗi ngày, máy ATM được bắt đầu bởi nhân viên IT của ngân hàng. Tại thời điểm này, máy ATM đọc danh sách các account number (số tài khoản), PINs (Personal Id Numbers), và account balances (số tiền hiện có trong tài khoản) từ 1 file.

- Sau khi máy bắt đầu, các khách hàng sử dụng ATM để thực hiện giao dịch gồm: Gửi tiền; Rút tiền, xem tài khoản.
- Mỗi giao dịch của khách hàng, ATM yêu cầu xác nhận account number và số PIN, sau đó có thể thực hiện giao dịch cho mỗi khách hàng. Mỗi giao dịch thành công, ATM lưu lại danh sách các giao dịch bao gồm: account number và số tiền. Cập nhật tài khoản hiện tại của account (balance).
- Lưu ý máy ATM giới hạn số lần giao dịch của mỗi khách hàng trên một ngày và sẽ từ chối giao dịch nếu khách hàng giao dịch vượt quá số lần qui định.
- Kết thúc một ngày, nhân viên IT của ngân hàng sẽ nhập vào mã bảo mật, máy ATM sẽ shutdown và ghi danh sách giao dịch vào 1 file.

## **Đề 702: Xây dựng ứng dụng đặt món ăn và thanh toán đơn hàng.**

### **Mục tiêu:**

- Sử dụng các mảng song song để lưu dữ liệu quan hệ.
- Sử dụng các mảng một chiều để lưu các món ăn: tên món, đơn giá.
- Sử dụng các thuật toán thao tác với mảng.
- Xuất ra file.
- Viết các hàm để xây dựng các chức năng.

### **Yêu cầu:**

Viết chương trình bằng C/C++ để giải quyết bài toán.

### **Mô tả chung về đề tài:**

- Bắt đầu một ngày, thu ngân nhập vào một mã giao dịch.
- Chương trình in ra Menu các món ăn của nhà hàng. Cho phép khách hàng nhập vào các món ăn cần đặt với số lượng tương ứng. Chương trình sẽ tính tổng, giảm giá nếu có và in ra hóa đơn.
- Khách hàng được phép nhập tối đa 5 món và nhập vào 0 0 khi không chọn thêm món. Vì vậy, khi khách hàng đã chọn đủ 5 món thì tiến trình chọn món cũng kết thúc ngay lập tức và in ra Hóa đơn mà không cần khách hàng phải nhập vào 0 0.
- Khách hàng sẽ được giảm giá 25% nếu Tổng đơn hàng  $\geq 2$  triệu.
- Kết thúc một ngày thu ngân sẽ nhập vào mã giao dịch trên. Chương trình thống kê in ra toàn bộ các đơn hàng đã có trong ngày và ghi ra file.

### **Đề 703: Xây dựng ứng dụng quản lý danh sách sinh viên.**

#### **Mục tiêu:**

- Sử dụng các mảng song song để lưu dữ liệu quan hệ.
- Sử dụng các mảng một chiều.
- Sử dụng các thuật toán thao tác với mảng.
- Sử dụng các hàm xử lý chuỗi.
- Viết các hàm để xây dựng các chức năng.

#### **Yêu cầu:**

Viết chương trình bằng C/C++ để giải quyết bài toán.

#### **Mô tả chung về đề tài:**

- Chương trình cho phép quản lý khoản 100 sinh viên/mỗi lớp. Gồm menu sau:
  - 1-Thêm sinh viên.
  - 2-Sắp xếp danh sách
  - 3-Xóa sinh viên.
  - 4-Tìm sinh viên.
  - 5-Cấp Mã sinh viên
  - 6-Cấp email
  - 7-In ra danh sách.
  - 8-Thoát.
- Mỗi một lớp sẽ được lưu trên 1 file.
- Chức năng Thêm sinh viên: Nhập Họ lót, tên, ngày sinh, giới tính, địa chỉ;
- Chức năng Cấp Mã sinh viên chương trình tự động cấp sau khi sắp xếp danh sách theo thứ tự Alphabe của Tên và Họ lót. Nếu chưa sắp xếp chương trình sẽ thông báo lỗi.
- Chức năng địa chỉ email chương trình tự động cấp sau khi cấp Mã sinh viên. Nếu chưa cấp mã chương trình sẽ thông báo lỗi.

### **Đề 704: Xây dựng ứng dụng quản lý điểm của sinh viên theo từng môn học.**

**Mục tiêu:**

- Sử dụng các mảng song song để lưu dữ liệu quan hệ.
- Sử dụng các mảng một chiều.
- Sử dụng các thuật toán thao tác với mảng.
- Sử dụng các hàm xử lý chuỗi.
- Viết các hàm để xây dựng các chức năng.

**Yêu cầu:**

Viết chương trình bằng C/C++ để giải quyết bài toán.

**Mô tả chung về đề tài:**

- Chương trình cho phép quản lý khoản 40 sinh viên/mỗi môn. Gồm menu sau:
  - 1-Nhập điểm.
  - 2-Xem điểm
  - 3-Sửa điểm.
  - 4-Sắp xếp
  - 5-In ra danh sách.
  - 6-Thoát.
- Mỗi một môn học sẽ được lưu trên 1 file.
- Chức năng Nhập điểm: Nhập vào các cột điểm: Lab1, Lab2, Progress Test 1, Progress Test 2, Presentation, Final Test.
- Chức năng Xem điểm: Nhập vào Mã sinh viên cần xem. Hiện thị ra danh sách các điểm của một sinh viên cần xem. Nếu không tìm thấy thông báo Không tìm thấy sinh viên cần xem.
- Chức năng Sửa điểm: Nhập vào Mã sinh viên cột điểm cần sửa. Nếu không tìm thấy Mã sinh viên thì thông báo Không tìm thấy.
- Chức năng sắp xếp: Tính Điểm trung bình, quy đổi sang điểm chữ (A, B,...) rồi sắp xếp theo thứ tự giảm dần của điểm trung bình.

## **Đề 705: Mô phỏng bài toán tô màu bản đồ bằng tô màu các đỉnh của đồ thị.**

### **Mục tiêu:**

- Sử dụng lý thuyết đồ thị.
- Thuật toán và cấu trúc dữ liệu.
- Viết các hàm để xây dựng các chức năng.

### **Yêu cầu:**

Viết chương trình bằng C/C++ để giải quyết bài toán.

### **Mô tả chung về đề tài:**

- Đối với bất kỳ mặt phẳng nào được chia thành các vùng phân biệt, chẳng hạn như bản đồ hành chính của một quốc gia, chỉ cần dùng tối đa bốn màu để phân biệt các vùng lân cận với nhau. Hai vùng được coi là lân cận nếu như chúng có chung nhau một đoạn đường biên, không tính chung nhau một điểm.
- Để giải bài toán: Xây dựng các đồ thị tương ứng với các bản đồ và đưa giả thuyết 4 màu về giả thuyết số màu của đồ thị phẳng.
- Với mỗi bản đồ, ta có thể định nghĩa một đồ thị:
  - **Tập đỉnh:** gồm các đỉnh mà mỗi đỉnh tương ứng với một vùng trên bản đồ;
  - **Tập cạnh:** mỗi cạnh tương ứng với 2 vùng kề nhau có chung một biên giới.Cách tô màu vùng của bản đồ tương ứng với cách tô màu đỉnh của đồ thị, sao cho hai đỉnh kề nhau không cùng một màu được sử dụng ít nhất.

### **Thuật toán:**

- Input: đồ thị  $G = (V, E)$ .
- Output: đồ thị  $G = (V, E)$  có các đỉnh đã được tô màu.

**Bước 1:** Tính giá trị bậc của các đỉnh trong  $V$ . Lập danh sách  $V' = [v_1, v_2, \dots, v_n]$  là các đỉnh của đồ thị được sắp xếp theo thứ tự bậc giảm dần:  $d(v_1) > d(v_2) > \dots > d(v_n)$ . Ban đầu tất cả các đỉnh trong  $V$  ( $V'$ ) đều chưa được tô màu.  
Gán  $i = 1$ ;

**Bước 2:** Tô màu  $i$  cho đỉnh đầu tiên trong danh sách  $V'$ . Duyệt lần lượt các đỉnh khác trong  $V'$  (nếu có) và chỉ tô màu  $i$  cho các đỉnh không kề đỉnh đã có màu  $i$ .



**Bước 3:** Kiểm tra nếu tất cả các đỉnh trong  $V$  đã được tô màu thì thuật toán kết thúc, đồ thị đã sử dụng  $i$  màu để tô. Ngược lại, nếu vẫn còn đỉnh chưa được tô thì chuyển sang bước 4.

**Bước 4:** Loại khỏi danh sách  $V'$  các đỉnh đã tô màu. Sắp xếp lại các đỉnh trong  $V'$  theo thứ tự bậc giảm dần.

Gán  $i = i + 1$

Quay lại bước 2.

### Đề 801:

**Bộ môn mạng và truyền thông của Khoa Công nghệ thông tin, trường Đại học Bách Khoa cần chuyển chuyển hết  $n$  gói tin trên một mạng có  $m$  kênh truyền. Biết chi phí để chuyển  $i$  gói tin trên kênh  $j$  là  $C(i,j)$ .**

Yêu cầu: Viết chương trình bằng ngôn ngữ C, cho biết một phương án chuyển với chi phí thấp nhất.

I. Đầu vào: file văn bản messages.inp

a. + Dòng đầu tiên: 2 số  $n$  và  $m$

b. + Dòng thứ  $i$  trong số  $n$  dòng tiếp theo: dãy  $m$  số nguyên dương  $b_1, b_2, \dots, b_m$  trong đó  $b_j$  là chi phí chuyển  $i$  gói tin trên kênh  $j$ .

II. Đầu ra: file văn bản messages.out

+ Dòng đầu tiên: tổng chi phí thấp nhất theo phương án tìm được

+ Dòng thứ  $j$  trong số  $m$  dòng tiếp theo: Số lượng gói tin chuyển trên kênh  $j$

messages.inp				messages.out
5	4			2
31	10	1	1	0
1	31	12	13	4
4	10	31	1	1
6	1	20	19	0

10    5    10    10

Ý nghĩa: với  $n=5$  gói tin,  $m=4$  kênh và chi phí  $c(i,j)$  cho trước, trong đó  $i$  là chỉ số dòng (số gói tin),  $j$  là chỉ số cột (kênh) thì cách chuyển sau đây sẽ cho chi phí thấp nhất là 2

Kênh	Số gói tin	Chi phí
1	0	0
2	4	1
3	1	1
4	0	0

### III. Thuật toán: Có thể dùng quy hoạch động

Gọi  $s(i,j)$  là chi phí thấp nhất của phương án chuyển hết  $i$  gói tin trên mạng gồm  $j$  kênh đầu tiên  $1, 2, \dots, j$ . Ta xét kênh thứ  $j$  và thứ chuyển lần lượt  $k=0,1,\dots,i$  gói tin theo kênh này. Ta thấy, nếu chuyển  $k$  gói tin theo kênh  $j$  thì chi phí cho kênh  $j$  này sẽ là  $c(i, j)$ , còn lại  $i-k$  gói tin phải chuyển trên  $j-1$  kênh đầu tiên với chi phí là  $s(i-k, j-1)$ , vậy phương án này cho chi phí là  $c(k,j) + s(i-k, j-1)$ ,  $k=0,1,2,\dots,i$ .

Ta có  $s(i,j)=\min \{c(k,j) + s(i-k, j-1) | k = 0,1, \dots, i\}$

Để cài đặt, ta có thể dùng 1 mảng 1 chiều  $p$  và  $m$  bước lặp. Tại bước lặp thứ  $j$  ta có  $p[i]=s(i,j)$  là chi phí thấp nhất khi chuyển hết  $i$  gói tin trên mạng với kênh  $j$  đầu tiên. Vậy :

$$p[i] = \min\{c(k,j) + p(i-k) | k = i, i-1, \dots, 0\}$$

$i=n, \dots, 1$ .

Chú ý rằng để khởi ghi đè lên giá trị còn phải dùng để xử lý bước sau, với mỗi kênh  $j$  ta cần duyệt  $i$  theo chiều ngược từ  $n$  đến  $1$ .

**Đề 802: Độ cao của một số tự nhiên là tổng các chữ số của các số đó. Với mỗi cặp số tự nhiên  $n$  và  $h$  cho trước, hãy liệt kê các số nguyên tố không vượt quá  $n$  và có độ cao  $h$ ,  $10 \leq n \leq 1000000$ ;  $1 \leq h \leq 54$ .**

Yêu cầu: Viết chương trình bằng ngôn ngữ C (có sử dụng file ) để thực hiện yêu cầu trên

- I. Đầu vào:  $n, h$
- II. Đầu ra : mỗi dòng 1 số
- III. Thuật toán: liệt kê các số nguyên tố có độ cao  $h$  trong khoảng  $1..n$ 
  - Dùng hàm để xác định số nguyên tố trong khoảng  $1..n$  và đánh dấu vào mảng  $p$ :  $p[i]=0$  khi và chỉ khi  $i$  là số nguyên tố
  - Duyệt lại các số nguyên tố  $i$  trong danh sách  $p$ , tính độ cao của số  $i$  nếu Độ cao  $(i) = h$  thì ghi nhận.
  - Kết thúc.

**Đề 803: Viết chương trình bằng ngôn ngữ C biến đổi biểu thức trung tố sang biểu thức hậu tố và tính kết quả của biểu thức đó.**

- I. Đầu vào: Giả sử ta có biểu thức  $(1+5) * (8-(4-1))$
- II. Đầu ra: Tính toán giá trị của biểu thức này.
- III. Giải thích quá trình biến đổi và cách tính biểu thức

*Trong những năm đầu 1950, nhà logic học Ba Lan -Jan Łukasiewicz đã phát hiện rằng các dấu ngoặc là không cần thiết trong ký pháp hậu tố, còn được gọi là **ký pháp nghịch đảo Ba lan**( Reverse Polish Notation-RPN)*

*Để minh họa cách đánh giá của biểu thức hậu tố (RPN), ta xét biểu thức:*

*1 5 + 8 4 1 - - \* thì tương ứng biểu thức trung tố là  $(1+5) * (8-(4-1))$ . Biểu thức này được đọc từ trái sang phải cho đến tìm ra một toán tử. Tại thời điểm đó, hai toán hạng cuối cùng được đọc được kết hợp với toán tử này. Trong ví dụ này, toán tử đầu tiên được đọc là +, các toán hạng của nó là 1 và 5, như được chỉ ra bằng dấu gạch dưới đây:*

1 5 + 8 4 1 - - \*

*Thay biểu thức con này bằng giá trị 6 cho ta biểu thức hậu tố rút gọn (RPN):*

$6\ 8\ 4\ 1\ -\ -\ *$

Bắt đầu đọc lại từ trái sang phải , toán tử tiếp theo là – và chúng ta xác định được hai toán hạng của nó:

$6\ 8\ \underline{4\ 1}\ -\ -\ *$

Áp dụng toán tử này cho ra:

$6\ 8\ 3\ -\ *$

Toán tử tiếp theo được đọc là - và hai toán hạng là 8 và 3

$6\ \underline{8\ 3}\ -\ *$

Tính hiệu này ta được

$\underline{6\ 5}\ *$

Và ta nhận được giá trị 30 cho biểu thức này.

#### IV. Thuật toán chuyển từ dạng trung tố sang dạng hậu tố

1. Khởi tạo một ngăn xếp rỗng của các toán tử
2. While ( không xảy ra lỗi và chưa đạt đến kết thúc của biểu thức trung tố )  
ta làm các bước sau:
  - a) Đọc Token (hằng số, biến số , toán tử số học, các dấu ngoặc trái và ngoặc phải, hàm,..) tiếp theo trong biểu thức trung tố.
  - b) Nếu phần tử (Token) là
    - i. một dấu ngoặc trái : đẩy nó vào ngăn xếp
    - ii. một dấu ngoặc phải : Lấy ra và hiển thị các phần tử của ngăn xếp cho đến khi dấu ngoặc trái được đọc. (nếu ngăn xếp rỗng thì xảy ra lỗi)
    - iii. một toán tử : nếu ngăn xếp rỗng hay Token được ưu tiên hơn phần tử đỉnh ngăn xếp, đẩy Token vào ngăn xếp.  
Nếu khác, hiển thị phần tử đỉnh của ngăn xếp, tập lại việc so sánh Token với phần tử ở đỉnh ngăn xếp.  
Chú thích: dấu ngoặc trái được xem là có độ ưu tiên thấp hơn các toán tử

iv. một toán hạng: hiển thị nó

3. Khi đạt đến kết thúc của biểu thức trung tố, lấy ra và hiển thị các phần tử của ngăn xếp cho đến khi ngăn xếp rỗng.

**Đề 804: Ta định nghĩa từ là 1 dãy gồm tối đa 10 chữ cái Latinh và trong một văn bản thì mỗi từ được phân cách với các từ kế nó ít nhất 1 ký tự không phải là chữ cái. Ngoài ra, đối với từ không phân biệt giữa chữ thường và chữ in.**

Yêu cầu: Viết chương trình bằng ngôn ngữ C thực hiện các công việc sau:

- Đọc file 01 văn bản gồm các dòng có độ dài tối đa 80, văn bản này được kết thúc bởi một dòng rỗng.
  - Đếm và in ra tần suất (số lần xuất hiện) của các từ trong văn bản. Nếu số lượng từ khác nhau trong văn bản nhiều hơn 100 thì chỉ tính tần suất của 100 từ khác nhau xuất hiện đầu tiên.
  - In ra bảng tần suất nói trên theo thứ tự ABC.
- I. Đầu vào: 01 file văn bản gồm các dòng có độ dài tối đa 80.
- II. Đầu ra: 01 file bảng tần suất nói trên theo thứ tự ABC
- III. Thuật toán: Sử dụng cấu trúc dữ liệu dạng cây
- + Thực hiện tìm kiếm nút có giá trị x trong cây. Nếu tìm thấy thì trả về vị trí của nút đó, ngược lại chèn nút vào cây nhị phân tìm kiếm.
- + Ta có thể sử dụng giải thuật tìm kiếm và thêm vào cây để sắp xếp thứ tự một dãy: Tạo ra cây nhị phân tìm kiếm, rồi duyệt theo thứ tự LNR

**Đề 805: Vẽ đồ thị nhưng dưới cấu trúc dữ liệu cây nhị phân (gọi là cây hàm) để thực hiện phân tích một xâu nhập từ file ở dạng biểu thức xây dựng thành cây.**

Yêu cầu: Viết chương trình bằng ngôn ngữ C vẽ cây nhị phân này.

- I. Đầu vào: 01 file văn bản chứa biểu thức toán học.
- II. Đầu ra: 01 file chứa kết quả của cây hàm
- III. Thuật toán: Sử dụng cấu trúc dữ liệu dạng cây

## ĐỀ 806. BÀI TOÁN TRÊN ĐỒ THỊ

Viết chương trình phát sinh ngẫu nhiên ma trận trọng số  $A_{n \times n} = (a_{ij})_{n \times n}$  của đồ thị vô hướng liên thông  $G$  gồm  $n$  đỉnh ( $a_{ij} = a_{ji} \forall i, j = 1..n$ ).

- Kiểm tra đồ thị  $G$  có phải là đồ thị Euler hay không ?
- Nhập hai đỉnh  $x, y$  và dùng thuật toán Dijkstra để tìm đường đi ngắn nhất từ  $x$  đến  $y$ .
- Dùng thuật toán Prim để tìm cây phủ nhỏ nhất của đồ thị  $G$ .

**Yêu cầu:** Viết chương trình bằng ngôn ngữ C thực hiện các công việc trên

1. **Đầu vào:** Phát sinh ngẫu nhiên ma trận trọng số  $A_{n \times n} = (a_{ij})_{n \times n}$

2. **Đầu ra:** Thực hiện các yêu cầu a, b, c của đề tài.

3. **Thuật toán**

- Tạo ma trận: Tạo ngẫu nhiên ma trận của đồ thị  $G$ .
- Chuyển ma trận về dạng có  $a[i][j]$  và các cạnh không có đường đi bằng  $\infty$ .
- Kiểm tra đồ thị Euler: tất cả các đỉnh của đồ thị phải có bậc chẵn.
- Thuật toán Dijkstra:
  - + Đầu vào: đồ thị có hướng  $G=(V,E)$  với  $n$  đỉnh,  $x \in V$  là đỉnh xuất phát,  $y$  là đỉnh đích,  $a[u,v]$ ,  $u,v \in V$  là ma trận trọng số.

+ Đầu ra: khoảng cách từ  $x \rightarrow y$ .

+ Truoc[v] ghi nhận đỉnh đi trước v trong đường đi ngắn nhất từ  $x \rightarrow v$

+ Bước 1: Khởi tạo:  $d[v] = a[x,v]$ ;  $\text{Truoc}[v] = x$ ;

$d[x] = 0$ ;  $T = V \setminus \{x\}$ ,  $T$  là tập các đỉnh có nhãn tạm thời.

+ Bước 2: Lặp

While  $T \neq \Phi$ :

- Tìm đỉnh  $u$  thỏa mãn  $d[u] = \min \{d[z], z \in T\}$ ;

-  $T = T \setminus \{u\}$  //Cố định nhãn của đỉnh  $u$ .

- for  $v \in T$ : //Gán nhãn lại cho các đỉnh trong  $T$

if( $d[v] > d[u] + a[u,v]$ ) {

$d[v] = d[u] + a[u,v]$ ;

$\text{Truoc}[v] = u$ ;

}

- Cây phủ Prim:

**Bước 1:** Xuất phát từ đỉnh  $k$  bất kỳ (thông thường chọn đỉnh đầu tiên) chọn một cạnh có trọng số nhỏ nhất liên kết với đỉnh  $k$  ( $\min \{A[k][j]\}_{j=1..n}$ ) ta đánh dấu 2 đỉnh đi qua cạnh đó và số cạnh tìm được là 1. Chuyển sang bước 2.

**Bước 2:** Tìm cạnh nhỏ nhất của đồ thị với điều kiện cạnh tìm được phải có 1 đỉnh chưa đánh dấu và 1 đỉnh đã đánh dấu ( $\min \{A[i][j]\}_{j=1..n, i=1..n}$  sao cho  $i$  đánh dấu và  $j$  chưa đánh dấu) để tránh trường hợp tạo thành chu trình. Ta tăng số

cạnh tìm được lên 1 và chuyển sang bước 3.

**Bước 3:** Nếu số cạnh tìm được bằng  $n-1$  thì kết thúc, ngược lại quay về bước 2.

Ta tổ chức mảng 1 chiều  $D$  để đánh dấu. Nếu  $D[i]=1$  đỉnh  $i$  được đánh dấu và  $D[i]=0$  nếu  $i$  chưa được đánh dấu.

- Tìm  $\min\{A[1][j]\}_{j=1..n}$ . Sau đó gán  $D[1]=D[j]=1$  (đánh dấu 2 đỉnh 1,j) và cho số cạnh tìm được bằng 1 ( $Dem=1$ ).

- Tìm  $\min\{A[i][j]\}_{j=1..n, i=1..n}$  với điều kiện  $D[i]=1$  và  $D[j]=0$ . Sau đó gán  $D[j]=1$  (đánh dấu đỉnh  $j$  vừa tìm được) và tăng số cạnh lên 1 ( $Dem++$ ).

- Nếu  $Dem = n-1$  thì kết thúc.

## **Đề 807. CÂY PHỦ VÀ THUẬT TOÁN TÌM CÂY PHỦ THEO CHIỀU NGANG**

**Yêu cầu:**

- 1) Trình bày khái niệm cây, cây phủ, định lý tương đương
- 2) Trình bày thuật toán tìm cây phủ theo chiều ngang
- 3) Thiết kế cấu trúc dữ liệu và giải thuật tìm cây phủ theo chiều ngang
- 4) Cài đặt chương trình thuật toán tìm cây phủ theo chiều ngang bằng ngôn ngữ C
- 5) Kết quả DEMO: 10 bộ dữ liệu test chương trình

## **Đề 808. CÂY PHỦ VÀ THUẬT TOÁN TÌM CÂY PHỦ THEO CHIỀU SÂU**

**Yêu cầu:**

- 1) Trình bày khái niệm cây, cây phủ, định lý tương đương
- 2) Trình bày thuật toán tìm cây phủ theo chiều sâu
- 3) Thiết kế cấu trúc dữ liệu và giải thuật tìm cây phủ theo chiều sâu
- 4) Cài đặt chương trình thuật toán tìm cây phủ theo chiều sâu bằng ngôn ngữ C
- 5) Kết quả DEMO: 10 bộ dữ liệu test chương trình

## **Đề 808. CÂY PHỦ NHỎ NHẤT VÀ THUẬT TOÁN PRIM**

- 1) Trình bày khái niệm cây, cây phủ, cây phủ nhỏ nhất
- 2) Trình bày thuật toán Prim tìm cây phủ nhỏ nhất
- 3) Thiết kế cấu trúc dữ liệu và giải thuật Prim tìm cây phủ nhỏ nhất
- 4) Cài đặt chương trình thuật toán Prim tìm cây phủ nhỏ nhất bằng ngôn ngữ C
- 5) Kết quả DEMO: 10 bộ dữ liệu test chương trình

#### **Đề 809. CÂY PHỦ NHỎ NHẤT VÀ THUẬT TOÁN KRUSKAL**

- 1) Trình bày khái niệm cây, cây phủ, cây phủ nhỏ nhất
- 2) Trình bày thuật toán KRUSKAL tìm cây phủ nhỏ nhất
- 3) Thiết kế cấu trúc dữ liệu và giải thuật KRUSKAL tìm cây phủ nhỏ nhất
- 4) Cài đặt chương trình thuật toán KRUSKAL tìm cây phủ nhỏ nhất bằng ngôn ngữ C
- 5) Kết quả DEMO: 10 bộ dữ liệu test chương trình

#### **Đề 810. BÀI TOÁN NGƯỜI DU LỊCH**

- 1) Trình bày thuật toán nhánh cận gồm thủ tục rút gọn, thủ tục phân nhánh, thủ tục chọn cạnh phân nhánh, chọn hai cạnh cuối cùng
- 2) Thiết kế cấu trúc dữ liệu và giải thuật nhánh cận này để giải bài toán người du lịch.
- 3) Cài đặt chương trình thuật toán nhánh cận này để giải bài toán người du lịch bằng ngôn ngữ C

Kết quả DEMO: 10 bộ dữ liệu test chương trình



### Đề 901:

Tính toán trên đa thức

Cho đa thức bậc  $n$ ,  $p(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$

- Tính  $q1 = p(1) + p(2) + \dots + p(m)$
- Tính  $q2 = p(x_1) + p(x_2) + \dots + p(x_m)$

*Khai báo dữ liệu:*

- Danh sách đặc  $a$  (mảng  $a$ ) gồm  $n+1$  hệ số của đa thức  $P(x)$ .
- Danh sách đặc  $x$  (mảng  $x$ ) gồm  $m$  phần tử của vec tơ  $x$

*Yêu cầu:*

- Đọc  $n, m$ , các hệ số đa thức & các phần tử vec tơ  $x$  từ file \*.IN
- Ghi kết quả  $q1, q2$  ra file \*.OUT và ra màn hình

### Đề 902: Tính toán trên đa thức

Cho 2 đa thức:  $p(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$

$$q(y) = b_0y^m + b_1y^{m-1} + \dots + b_{m-1}y + b_m$$

- Tính  $res1 = p(z) + q(z)$
- Tính  $res2 = q(z) + q(t)$
- Tính  $res3 = p(z) + q(t)$

*Khai báo dữ liệu:*

- Danh sách đặc  $a$  (mảng  $a$ ) gồm  $n+1$  hệ số của đa thức  $p(x)$
- Danh sách đặc  $b$  (mảng  $b$ ) gồm  $m+1$  hệ số của đa thức  $q(x)$

*Yêu cầu:*

- Đọc  $n, m, z, t$  và các hệ số của 2 đa thức từ file \*.IN
- Ghi kết quả  $res1, res2, res3$  ra file \*.OUT và màn hình

### Đề 903: Tìm nghiệm phương trình $p(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n = 0$ bằng phương pháp dây cung và phương pháp chia đôi.

*Khai báo dữ liệu:* Danh sách đặc  $A$  (mảng  $A$ ) gồm  $n+1$  hệ số của đa thức  $p(x)$

*Yêu cầu:*

- Đọc  $n$  và các hệ số từ file \*.INP hoặc tạo ngẫu nhiên
- Cho phép thay đổi  $a, b$  & kiểm tra ràng buộc  $(a, b)$  là khoảng nghiệm
- Ghi kết quả là nghiệm gần đúng ra file \*.OUT và ra màn hình

### Đề 904: Giải hệ phương trình đại số tuyến tính ( $n$ phương trình, $n$ ẩn) bằng phương pháp Crame, phương pháp Gauss

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận  $A$  cấp  $n, n+1$ .

*Yêu cầu:*

- Thực hiện biến đổi ma trận A về dạng tam giác trên, tìm nghiệm theo quá trình ngược bậc đa thức và các hệ số đa thức
- Định nghĩa hàm tính định thức ma trận vuông, tìm nghiệm dựa vào hàm đã định nghĩa
- Đọc  $n$ , ma trận A từ file \*.INP
- Ghi kết quả  $n$  số  $x_1, x_2, \dots, x_n$  là nghiệm của hệ phương trình ra file \*.OUT
- Ghi ra màn hình: ma trận A đọc từ file, ma trận A qua các bước biến đổi, vector nghiệm  $(x_1, x_2, \dots, x_n)$  của hệ phương trình hình

**Đề 905:** Giải hệ phương trình đại số tuyến tính ( $n$  phương trình,  $n$  ẩn) bằng phương pháp **Crame**, phương pháp **Gauss Jordan**

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận A cấp  $n, n+1$ .

*Yêu cầu:*

- Thực hiện biến đổi ma trận A về dạng tam giác trên, tìm nghiệm theo quá trình ngược bậc đa thức và các hệ số đa thức
- Định nghĩa hàm tính định thức ma trận vuông, tìm nghiệm dựa vào hàm đã định nghĩa
- Đọc  $n$ , ma trận A từ file \*.INP
- Ghi kết quả  $n$  số  $x_1, x_2, \dots, x_n$  là nghiệm của hệ phương trình ra file \*.OUT
- Ghi ra màn hình: ma trận A đọc từ file, ma trận A qua các bước biến đổi, vector nghiệm  $(x_1, x_2, \dots, x_n)$  của hệ phương trình

**Đề 906:** Giải hệ phương trình đại số tuyến tính ( $n$  phương trình,  $n$  ẩn) bằng phương pháp **Gauss\_Sediel (tự sửa sai)**

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận A cấp  $n, n+1$

*Yêu cầu:*

- Đọc  $n$ , ma trận A từ file \*.INP
- Nhập vec tơ nghiệm ban đầu & cho phép thay đổi vec tơ này
- Ghi kết quả  $n$  số  $x_1, x_2, \dots, x_n$  từ bước đầu (vec tơ nghiệm sai) cho đến bước cuối (là vec tơ nghiệm gần đúng của hệ phương trình)

**Đề 907:** Giải hệ phương trình đại số tuyến tính ( $n$  phương trình,  $n$  ẩn) bằng phương pháp **giảm dư**

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận A cấp  $n, n+1$

*Yêu cầu:*

- Đọc  $n$ , ma trận  $A$  từ file \*.INP
- Nhập vec tơ nghiệm ban đầu & cho phép thay đổi vec tơ này
- Ghi kết quả  $n$  số  $x_1, x_2, \dots, x_n$  từ bước đầu (gây ra các số dư) cho đến bước cuối (các số dư tiến về 0, và là vec tơ nghiệm gần đúng của hệ phương trình)

**Đề 908:** Cho hàm  $f(x)$  nhận giá trị  $y_i$  tại các điểm tương ứng  $x_i$  ( $i=0, 1, 2, \dots, n$ ) và giá trị  $c$ . Viết chương trình tính và in ra bảng nội suy Ayken (theo 2 dạng: Ayken 1, Ayken 2)

*Khai báo dữ liệu:*

- Danh sách đặc  $x$  (mảng  $x$ ) gồm  $n+1$  phần tử
- Danh sách đặc  $y$  (mảng  $y$ ) gồm  $n+1$  phần tử

*Yêu cầu:*

- Đọc  $n$ , các phần tử của vec tơ  $x$ , vec tơ  $y$  từ file \*.INP
- Cho phép thay đổi giá trị  $c \in (x_0, x_n)$
- Ghi kết quả là bảng nội suy Ayken và giá trị gần đúng  $f(c)$  ra màn hình

**Đề 909:** Cho hàm  $f(x)$  nhận giá trị  $y_i$  tại các điểm tương ứng  $x_i$  ( $i=1, 2, \dots, n$ ). Xây dựng biểu thức dạng phi tuyến tính  $f(x)=ae^{bx}$  bằng phương pháp bình phương tối thiểu

*Khai báo dữ liệu:*

- Danh sách đặc  $x$  (mảng  $x$ ) gồm  $n$  phần tử
- Danh sách đặc  $y$  (mảng  $y$ ) gồm  $n$  phần tử

*Yêu cầu:*

- Đọc  $n$ , các phần tử của vec tơ  $x$ , vec tơ  $y$  từ file \*.INP
- Ghi kết quả là hệ số  $a, b$  của biểu thức hàm  $f(x)$  ra màn hình

**Đề 910:** Cho hàm  $f(x)$  nhận giá trị  $y_i$  tại các điểm tương ứng  $x_i$  ( $i=1, 2, \dots, n$ ). Xây dựng biểu thức dạng phi tuyến tính  $f(x)=ax^b$  bằng phương pháp bình phương tối thiểu.

*Khai báo dữ liệu:*

- Danh sách đặc  $x$  (mảng  $x$ ) gồm  $n$  phần tử.
- Danh sách đặc  $y$  (mảng  $y$ ) gồm  $n$  phần tử.

*Yêu cầu:*

- Đọc  $n$ , các phần tử của vec tơ  $x$ , vec tơ  $y$  từ file \*.INP
- Ghi kết quả là hệ số  $a, b$  của biểu thức hàm  $f(x)$  ra màn hình

**Đề 911:** Cho hàm đa thức  $p(x)$  bậc  $n$  và các hàm siêu việt  $g(x)$ ,  $h(x)$ . Tính gần đúng tích phân xác định của các hàm  $p(x)$ ,  $g(x)$ ,  $h(x)$  trên  $[a, b]$  theo các công thức: Hình thang, Parabol, Newton Cotet.

*Khai báo dữ liệu:*

- Danh sách đặc  $a$  (mảng  $a$ ) gồm  $n + 1$  hệ số của đa thức.
- Mảng 2 chiều lưu trữ các hệ số  $p_n^i$

*Yêu cầu:*

- Nhập bậc đa thức và các hệ số đa thức
- Khai báo hàm  $g(x)$  và  $h(x)$
- Nhập số đoạn cần chia
- Nhập cận tính tích phân
- Đọc các hệ số  $p_n^i$  tương ứng từ file \*.INP
- Ghi kết quả là bảng tính gần đúng tích phân xác định của các hàm trên (theo 3 công thức) ra màn hình

**Đề tài 912:** Tìm giá trị riêng của ma trận vuông (cấp  $n$ ) bằng phương pháp Đanhilepski

*Khai báo dữ liệu:* Mảng 2 chiều cho ma trận  $A$  cấp  $n$ .

*Yêu cầu:*

- Đọc  $n$ , ma trận  $A$  từ file \*.INP
- Thực hiện biến đổi ma trận  $A$  về ma trận  $P$  (có dạng Phorebemit)
- Tìm nghiệm phương trình đại số với các hệ số lấy từ ma trận  $P$ .
- Ghi kết quả là ma trận  $P$ , giá trị riêng của ma trận  $A$  ra file \*.OUT và ra màn hình.

## **Đề A01:**

### **Học từ vựng tiếng anh thông qua trò chơi (2 sinh viên)**

Xây dựng trò chơi đồ chữ đơn giản để giúp người chơi củng cố từ vựng trong Tiếng Anh

**Yêu cầu:**

Về lý thuyết: Nắm vững cơ sở lý thuyết về cấu trúc dữ liệu, đặc biệt là cấu trúc bảng băm và các kỹ thuật lưu trữ ngoài. Vận dụng các lý thuyết này vào cài đặt tự điển.

Về lập trình:

*Các chức năng chính của chương trình :*

- **Thêm từ vựng:** cho phép thêm từ vựng mới vào trong từ điển và lưu từ điển vào tập tin. Với mỗi từ trong từ điển cần lưu trữ các thông tin về loại từ (danh từ, động từ,...),

phiên âm quốc tế và các nghĩa có thể có của từ.

- **Hiển thị & cập nhật từ điển:** Hiển thị danh sách các từ có trong từ điển để kiểm tra. Sau khi hiển thị, người dùng có thể thực hiện các thao tác sau:

+ Hiển thị thông tin chi tiết của một từ vựng.

+ Sửa thông tin chi tiết của một từ vựng.

+ Xóa một từ vựng nào đó.

- **Chơi trò chơi đố chữ:** Chương trình chọn ngẫu nhiên một từ vựng nào đó, xóa một cách ngẫu nhiên một ký tự trong từ và yêu cầu người chơi nhập vào ký tự bị thiếu. Nếu người chơi nhập vào một ký tự làm thành một từ hợp lệ (nghĩa là có trong từ điển) thì chương trình hiển thị thông tin về từ đó, người chơi sẽ được tính điểm. Còn nếu người chơi nhập vào ký tự không làm thành từ hợp lệ, thì chương trình sẽ hiển thị thông báo lỗi: ‘Không có từ này trong tiếng Anh’

Ví dụ: Chương trình hiển thị từ **c\_t**:

- Nếu người chơi nhập vào chữ **u** thì sẽ tạo thành từ **cut**, khi đó chương trình sẽ hiển thị thông tin về từ **cut** và tính điểm cho người chơi.

- Hoặc nếu người chơi nhập vào chữ **a** thì sẽ tạo thành từ **cat**, khi đó chương trình sẽ hiển thị thông tin về từ **cat** và tính điểm cho người chơi.

- Hoặc nếu người chơi nhập vào chữ **i** thì sẽ tạo thành từ **cit**, thì chương trình sẽ hiển thị thông báo lỗi: ‘Không có từ này trong tiếng Anh’.

**Thuật toán gợi ý. Cây nhị phân hoặc Bảng băm.**

## **Đề A02: Danh bạ điện thoại (2 sinh viên)**

Danh bạ điện thoại cố định là một danh sách tuyến tính, mỗi phần tử của nó ứng với một đơn vị thuê bao gồm 03 (ba) trường:

- Tên đơn vị hoặc chủ đơn vị thuê bao.
- Địa chỉ
- Số điện thoại (không bao gồm mã vùng).

Một đơn vị Tổng đài muốn quản lý tất cả thuê bao của tất cả các tỉnh (thành phố) nên muốn tập hợp dữ liệu từ 01 tập tin có nhiều trường có cấu trúc như sau:

(xxxx)	<i>Tên tỉnh 1</i>		
	<i>Tên đơn vị 1</i>	<i>Địa chỉ 1</i>	<i>Số điện thoại 1</i>
	<i>Tên đơn vị 2</i>	<i>Địa chỉ 2</i>	<i>Số điện thoại 1</i>
....			
(xxxx)	<i>Tên tỉnh 2</i>		
	<i>Tên đơn vị 1</i>	<i>Địa chỉ 1</i>	<i>Số điện thoại 1</i>
	<i>Tên đơn vị 2</i>	<i>Địa chỉ 2</i>	<i>Số điện thoại 2</i>
....			

Chọn cấu trúc dữ liệu để lưu trữ Danh bạ trên và viết chương trình đọc dữ liệu từ tập tin lưu trữ trên máy tính. Yêu cầu thực hiện các công việc sau:

1. Tìm và thông báo nếu có 2 số điện thoại trùng nhau trong một tỉnh và yêu cầu người dùng xóa 1 số nếu trùng.
2. Liệt kê danh bạ điện thoại từng tỉnh khi có yêu cầu.
3. Xây dựng chức năng tìm kiếm và hiển thị thông tin theo Số điện thoại hoặc tên đơn vị.

Dữ liệu vào: Tập chứa danh sách danh bạ các tỉnh.

Dữ liệu ra: Hiển thị các thao tác theo menu chọn và lưu trữ sang tập tin nếu có yêu cầu.

**Phương pháp gợi ý: Sử dụng danh sách liên kết, ngăn xếp, hàng đợi.....**

### **ĐỀ A03: Sơ đồ mạng (2 sinh viên)**

Trường Đại học Bách khoa bao gồm nhiều khu giảng đường khác nhau. Tổ quản trị mạng cần xây dựng một sơ đồ mạng để kết nối tất cả các khu với nhau. Các khu có thể kết nối được với 1 số khu khác hoặc không. Trong 1 khu các phòng có thể kết nối được với nhau hoặc không. Kèm theo là độ dài chi phí dây mạng cần kết nối.

Ví dụ:

#### **Ký hiệu**

Khu	Khu có thể kết nối
A	C (150), E (100), F (120)
B	C (90), F (50)
C	A (150), B (90), E (70), F (80)

...

#### **Ký hiệu**

Phòng	Phòng có thể kết nối
A101	A102 (5), A103 (9), A201 (12)
A102	A202 (12), A135 (15)
C105	C205 (12)

....

Chọn cấu trúc dữ liệu để lưu trữ sơ đồ kết nối của các Khu, Phòng.

1. Đọc dữ liệu từ tập tin lưu trữ các Khu, Phòng.
2. Đưa ra Sơ đồ kết nối các Khu với chi phí thấp nhất

Dữ liệu vào: Tập tin chứa bảng mô tả kết nối và chi phí giữa các khu, phòng.

Dữ liệu ra: Tập tin chứa các lưu trữ mạng kết nối các Khu, Phòng với chi phí thấp nhất.

**Phương pháp gợi ý: Sử dụng cây khung nhỏ nhất**

### **ĐỀ A04: Sắp xếp lịch học tập**

Một sinh viên CNTT nếu muốn đạt một cấp độ nào đó trong quá trình học thì phải hoàn thành một số môn học theo quy định được mô tả như dưới đây:

<i>Số hiệu môn học</i>	<i>Tên môn học</i>	<i>Môn cần học trước</i>
<i>M1</i>	<i>Nhập môn tin học</i>	<i>Không</i>
<i>M2</i>	<i>Giải tích số</i>	<i>M4</i>
<i>M3</i>	<i>Cấu trúc dữ liệu và giải thuật</i>	<i>M1,M4</i>
<i>M4</i>	<i>Giải tích</i>	<i>Không</i>
<i>M5</i>	<i>Lập trình Assembly</i>	<i>M1</i>
<i>.....</i>	<i>.....</i>	<i>.....</i>

Chọn cấu trúc dữ liệu để lưu trữ Bảng tiến độ học tập trên và viết chương trình đọc dữ liệu từ tập tin lưu trữ trên máy tính. Yêu cầu thực hiện các công việc sau:

1. Kiểm tra tính hợp lý của dữ liệu. Ví dụ

<i>Số hiệu môn học</i>	<i>Tên môn học</i>	<i>Môn cần học trước</i>
<i>M1</i>	<i>Nhập môn tin học</i>	<i>M2</i>
<i>M2</i>	<i>Giải tích số</i>	<i>M1</i>
<i>Thì dữ liệu không phù hợp</i>		

2. Xây dựng lại cách lưu trữ dữ liệu theo dạng cây quan hệ.
3. Xuất ra tập tin các trình tự học tập của sinh viên thỏa mãn yêu cầu  $x$  (*số môn tối thiểu*  $\leq x \leq$  *số môn tối đa*).

Dữ liệu vào: Tập tin chứa bảng mô tả môn chương trình học.

Dữ liệu ra: Hiển thị các thao tác theo menu chọn và lưu trữ sang tập tin nếu có yêu cầu.

**Phương pháp gợi ý: Sử dụng danh sách liên kết, sắp xếp tôpô....**

#### **Đề A05: Bài toán cái túi (1 sinh viên)**

Trong siêu thị có  $n$  đồ vật ( $n \leq 1000$ ), đồ vật thứ  $i$  có trọng lượng là  $W[i] \leq 1000$  và giá trị  $V[i] \leq 1000$ . Một tên trộm đột nhập vào siêu thị, tên trộm mang theo một cái túi có thể mang được tối đa trọng lượng  $M$  ( $M \leq 1000$ ). Hỏi tên trộm sẽ lấy đi những đồ vật nào để được tổng giá trị lớn nhất.

Giải quyết bài toán trong các trường hợp sau:



- Mỗi vật chỉ được chọn một lần.
- Mỗi vật được chọn nhiều lần (không hạn chế số lần)

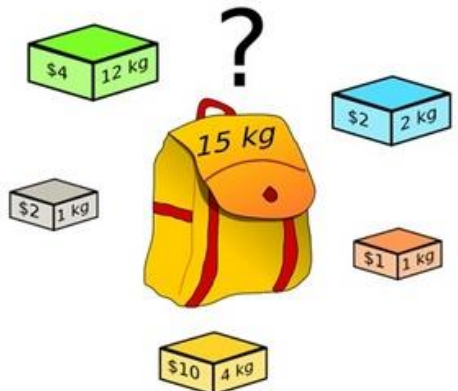
**InputData:** file văn bản Bag.inp

- Dòng 1: n, M cách nhau ít nhất một dấu cách
- n dòng tiếp theo: Mỗi dòng gồm 2 số  $W_i, V_i$ , là chi phí và giá trị đồ vật thứ  $i$ .

**OutputData:** file văn bản bag.out: Ghi giá trị lớn nhất tên trộm có thể lấy

**Example**

Input	Output
5 15 12 4 2 2 1 1 1 2 4 10	15



**Đề A06:** Minh họa chương trình quản lý sách đơn giản trong thư viện. Sử dụng cấu trúc dữ liệu danh sách liên kết đơn để cài đặt danh sách chứa nội dung các cuốn sách.

A. Thông tin liên quan đến một cuốn sách gồm: Mã số sách, Tên sách, Tên tác giả, Nhà xuất bản, Năm xuất bản, Trạng thái sách: {TRUE: chưa mượn/ FALSE: đã mượn}

B. Các thao tác trên danh sách này:

1. Khởi tạo danh sách

a. Khởi tạo danh mục sách rỗng (chưa có sách)

b. Đọc từ file: nhập vào tên file đã lưu danh mục sách ở lần làm việc trước đó.

2. Thêm một cuốn sách vào danh sách

a. Thêm vào đầu danh sách: InsertFirst

- b. Thêm vào sau một cuốn sách nào đó: InsertAfter
- c. Thêm vào cuối của danh sách: InsertLast

### 3. Xoá một cuốn sách khỏi danh sách theo

- a. Mã số
- b. Tên sách
- c. Tên tác giả: (nếu tác giả có nhiều sách thì xoá hết)
- d. Xoá cuốn sách ở đầu danh sách
- e. Xoá cuốn sách ở sau cuốn sách có mã số nào đó
- f. Xoá cuốn sách ở cuối danh sách

### 4. Tìm kiếm sách theo: lưu ý kết quả có bao nhiêu thì liệt kê ra hết, không phải chỉ liệt kê kết quả đầu tiên

- a. Tên sách
- b. Tên tác giả.
- c. Tên nhà xuất bản

### 5. Chức năng xem danh sách

- a. Xem toàn bộ danh mục sách trong danh sách
- b. Xem những cuốn sách đang cho mượn
- c. Xem những cuốn sách chưa cho mượn
- d. Xem danh sách theo thứ tự:
  - i. Theo vần alphabet của tên sách
  - ii. Theo vần alphabet của tên tác giả
  - iii. Theo vần alphabet của tên nhà xuất bản
  - iv. Sách được xuất bản mới nhất (theo năm)

### 6. Chức năng mượn/trả sách

- a. Mượn sách: liệt kê những cuốn sách chưa cho mượn, cho user chọn một cuốn sách ⇒ rồi cập nhật lại trạng thái cho mượn của sách.

b.Trả sách: nhập vào mã số sách được trả  $\Rightarrow$  cập nhật lại trạng thái đã trả sách cho cuốn sách đó.

7. Chức năng chỉnh sửa nội dung của sách: cho phép chọn các thông tin của sách để sửa và sau đó cập nhật lại.

8. Chức năng lưu file: nhập vào một tên file rồi lưu toàn bộ trạng thái hiện tại của danh mục sách vào đó.

**Yêu cầu: Sinh viên phải thực hiện theo các yêu cầu sau:**

1.Cài đặt cấu trúc dữ liệu Book theo mô tả như phần A

2. Cài đặt cấu trúc BookNode là phần tử của danh sách liên kết chứa danh mục sách

3. Cài đặt toàn bộ các chức năng mô tả trong phần B từ 1  $\rightarrow$  8: thể hiện các chức năng này theo *menu* chọn.

Ngoài ra sinh viên có thể bổ sung những chức năng mở rộng tùy ý. Tất cả các chức năng nâng cao này đều được đánh giá cao!

## **Đề A07: Quản lý gia phả**

Mô tả: Cài đặt cây gia phả và lưu vào tập tin, để hỗ trợ cho quá trình tìm kiếm thông tin.

**Yêu cầu của đề tài :**

- Nắm vững cơ sở lý thuyết về cấu trúc dữ liệu, các giải thuật duyệt cây. Vận dụng vào cài đặt cây gia phả.

- Giao diện thân thiện người dùng.

*Các chức năng chính của chương trình :*

- **Nhập cây gia phả**: của một dòng họ và lưu vào tập tin. Mỗi dòng họ là một tập tin. Chương trình cũng có thể dựng lại cây gia phả của dòng họ từ tập tin đã lưu. Mỗi người trong dòng họ cần lưu các thông tin như: họ tên, ngày sinh, giới tính, nghề nghiệp, ngày mất, họ tên chồng/vợ, số con (nếu là người nữ), v.v...

- **Hiển thị cây gia phả**: của dòng họ lên màn hình để kiểm tra.

Ví dụ: Có thể hiển thị cây gia phả của một dòng họ Nguyễn như sau:

N. Văn Phước

N. Thị Lộc (nút lá, vì là nữ)

N. Văn Thọ

N. Thị Hạnh

N. Văn Phúc

N. Văn Phú

N. Văn Quý

N. Thị Cát (nút lá, vì là nữ)

N. Văn Tường

...

- ***Tìm mối quan hệ***: Cho phép nhập vào họ tên của hai người nào đó, chương trình sẽ tìm xem họ có mối quan hệ thân thuộc gì với nhau không. Nếu có, thì cho biết mối quan hệ đó là gì (cha, mẹ, vợ, chồng, con trai, con gái, ông, bà nội, ông, bà hay cô, bác, v.v...)

Ví dụ: Với cây gia phả minh họa như trên, thì khi nhập vào N. Văn Quý và N. Thị Hạnh thì chương trình sẽ cho biết Quý là cháu trai của Hạnh, gọi Hạnh là cô Hai.

- ***Tìm người theo mối quan hệ***: Cho phép nhập vào họ tên X của một người nào đó, và mối quan hệ cần tìm. Chương trình sẽ hiển thị tên của người thân của X với mối quan hệ đã cho.

Ví dụ: Với cây gia phả minh họa như trên, thì khi nhập vào tên N. Văn Quý và mối quan hệ ‘bác’, thì chương trình sẽ hiển thị tên của các bác của Quý: N. Văn Phúc.

### **Đề A08: Lập lịch ưu tiên đúng hạn**

Có  $n$  công việc đánh số từ 1 đến  $n$  và một máy để thực hiện chúng. Biết:

- $p_j$  là thời gian cần thiết để hoàn thành công việc  $j$ .
- $d_i$  là thời hạn hoàn thành công việc  $i$ .

Mỗi công việc cần được thực hiện liên tục từ lúc bắt đầu cho tới khi kết thúc, không cho phép ngắt quãng. Khoảng thời gian thực hiện hai công việc bất kỳ chỉ được có nhiều nhất 1 điểm chung. Giả sử  $C_i$  là thời điểm hoàn thành trễ hạn, còn nếu  $C_i < d_i$  thì ta nói công việc  $i$  được hoàn thành đúng hạn.

**Yêu cầu:** Tìm trình tự thực hiện các công việc sao cho số công việc được hoàn thành đúng hạn là lớn nhất.

**Dữ liệu:** Vào từ file văn bản LICH.D.INP:

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $0 < n < 100$ )
- Dòng thứ 2 chứa  $n$  số nguyên dương  $p_1, p_2, \dots, p_n$ .
- Dòng thứ 3 chứa  $n$  số nguyên dương  $d_1, d_2, \dots, d_n$ .

**Kết quả:** Ghi ra file văn bản LICH.D.OUT.

Dòng đầu tiên ghi số lượng công việc được hoàn thành đúng hạn theo trình tự thu được.

Dòng tiếp theo ghi trình tự thực hiện các công việc đã cho.

### **Đề A09: Ong tìm mật**

Một chú ong đi hút mật trong rừng, lượng mật thu nhận được ước lượng sẵn trên đường đi từ đỉnh đồi A đến đỉnh đồi B. Nhiệm vụ chú ong được giao là từ tổ (xuất phát) phải đi đến hết tất cả các đỉnh đồi mỗi đỉnh 1 lần và quay về tổ (vị trí) với lượng mật thu được là nhiều nhất. Dữ liệu được mô tả như sau

Ví dụ:

```
D      //D là đỉnh chứa tổ của Ong
A      B      100  //Lượng mật thu nhận khi đi từ đỉnh A đến đỉnh D hoặc
ngược lại
A      D      200
.....
```

Chọn cấu trúc dữ liệu để mô tả dữ liệu trên.

**Dữ liệu vào:** Tập tin chứa bảng mô tả dữ liệu.

**Phương pháp gợi ý:** Sử dụng cây khung nhỏ nhất

**Dữ liệu ra:** Tập tin chứa hành trình tối ưu của Chú ong.

**A10:** Viết chương trình minh họa các giải thuật tìm kiếm và sắp xếp trên mảng có kích thước  $n$  phần tử nguyên. Chương trình được mô tả với các yêu cầu như sau:

- Cài đặt hàm tìm kiếm:
  - Tìm kiếm tuần tự (tuyến tính) cho mảng bất kỳ
  - Tìm kiếm nhị phân cho mảng dữ liệu được sắp tăng
- Cài đặt các hàm sắp xếp (tăng) theo phương pháp:
  - Chọn trực tiếp
  - Chèn trực tiếp
  - Nổi bọt
  - Đổi chỗ trực tiếp
  - Shell Sort
  - Quick Sort

Các hàm sắp xếp phải minh họa trực quan: tại mỗi bước hoán vị  $a[i]$  và  $a[j]$

### **Đề B01:**

**Xây dựng chương trình tìm đường đi ngắn nhất trên đồ thị có hướng có trọng số.**

Mô tả:

Dữ liệu vào: Tập chứa đỉnh xuất phát, đỉnh kết thúc và ma trận trọng số của đồ thị

Phương pháp gợi ý : Dijkstra và Floyd.

Dữ liệu ra: Tập chứa khoảng cách ngắn nhất và cách đi từ đỉnh xuất phát đến đỉnh kết thúc.

Tài liệu tham khảo chính: Chuyên đề lý thuyết đồ thị của Lê Minh Hoàng

**Đề B02: Xây dựng chương trình tìm đường đi ngắn nhất trên đồ thị có hướng có trọng số.**

Mô tả:

Dữ liệu vào: Tập chứa đỉnh xuất phát, đỉnh kết thúc và ma trận trọng số của đồ thị

Phương pháp gợi ý: Bellman-Ford và Floyd.

Dữ liệu ra: Tập chứa khoảng cách ngắn nhất và cách đi từ đỉnh xuất phát đến đỉnh kết thúc.

Tài liệu tham khảo chính: Chuyên đề lý thuyết đồ thị của Lê Minh Hoàng

**Đề B03: Xây dựng chương trình giải bài toán quy hoạch tuyến tính.**

Mô tả:

Dữ liệu vào: Tập chứa ma trận biểu diễn bài toán tối ưu tuyến tính.

Phương pháp gợi ý: Simplex Method và Two phases Simplex Method.

Dữ liệu ra: Tập chứa giá trị tối ưu và chứa tập giá trị các biến đạt giá trị tối ưu.

Tài liệu tham khảo chính: Giáo trình quy hoạch tuyến tính của Võ Văn Tuấn Dũng

**Đề B04: Xây dựng chương trình giải bài toán quy hoạch tuyến tính.**

Mô tả:

Dữ liệu vào: Tập chứa ma trận biểu diễn bài toán tối ưu tuyến tính.

Phương pháp gợi ý: Simplex Method và M-method.

Dữ liệu ra: Tập chứa giá trị tối ưu và chứa tập giá trị các biến đạt giá trị tối ưu.

Tài liệu tham khảo chính: Giáo trình quy hoạch tuyến tính của Võ Văn Tuấn Dũng

**Đề B05: Xây dựng chương trình dự đoán kết quả đánh giá của người dùng.**

Mô tả:

Dữ liệu vào: ma trận  $X$  kích cỡ  $n \times m$  ( $X$  nhận các giá trị từ 0 đến 5)

Phương pháp gợi ý: Latent Factor, matrix factorization.

Dữ liệu ra: Tập chứa ma trận  $Y$  kích cỡ  $n \times m$  và sao cho  $Y \cong X$ .

Tài liệu tham khảo chính: Introduction to Algorithmic Marketing, Lya Katsov, 2018

**Đề B06: Xây dựng chương trình dự đoán kết quả đánh giá của người dùng với trường hợp dữ liệu thưa.**

Mô tả:

Dữ liệu vào: ma trận  $X$  kích cỡ  $n \times m$  ( $X$  nhận giá trị từ 0 đến 5, với giá trị 0 chiếm  $\frac{2}{3}$  tổng số phần tử)

Phương pháp gợi ý: Hybrid method.

Dữ liệu ra: Tập chứa ma trận  $Y$  kích cỡ  $n \times m$  sao cho  $Y \cong X$ .

Tài liệu tham khảo chính: Introduction to Algorithmic Marketing, Lya Katsov, 2018

**Đề B07: Xây dựng chương trình tìm luồng cực đại trên mạng  $G$ .**

Mô tả:

Dữ liệu vào: Tập chứa đỉnh phát, đỉnh thu và ma trận trọng số của đồ thị  $G$ .

Phương pháp gợi ý: Ford–Fulkerson

Dữ liệu ra: Tập chứa ma trận biểu diễn luồng cực đại của mạng  $G$ .

Tài liệu tham khảo chính: Chuyên đề lý thuyết đồ thị của Lê Minh Hoàng

**Đề B08: Xây dựng cây khung bé nhất của đồ thị vô hướng  $G$ .**

Mô tả:

Dữ liệu vào: Tập chứa ma trận trọng số của đồ thị  $G$ .

Phương pháp gợi ý: Prim và Kruskal

Dữ liệu ra: Tập chứa các cạnh của cây khung của đồ thị  $G$  và trọng số của cây

Tài liệu tham khảo chính: Chuyên đề lý thuyết đồ thị của Lê Minh Hoàng

**Đề B09: Xây dựng hành trình người du lịch của đồ thị liên thông  $G$ .**

Mô tả:

Dữ liệu vào: Tập chứa ma trận trọng số của đồ thị  $G$ .



Phương pháp gợi ý: Nhân cận

Dữ liệu ra: Tập chứa hành trình người du lịch và tổng chi phí của hành trình

Tài liệu tham khảo chính: Tập bài giảng toán rời rạc của Nguyễn Văn Hiệu.

### **Đề B10: Xây dựng véc tơ riêng của ma trận vuông A .**

Mô tả:

Dữ liệu vào: Tập chứa ma trận vuông kích thước  $15 \times 15$  .

Phương pháp gợi ý: Chuẩn hóa ma trận, phương pháp véc tơ riêng

Dữ liệu ra: Tập chứa vector riêng của ma trận đầu vào.

Tài liệu tham khảo chính: Tập bài giảng cơ sở hệ thống thông tin của Nguyễn Văn Hiệu.

### **Đề C01: Đánh BOSS**

Bạn đang chơi một trò chơi điện tử trong đó bạn phải làm nhiệm vụ là đánh bại BOSS cuối để chiến thắng trò chơi. Bạn có thể thay phiên sử dụng  $N$  loại vũ khí khác nhau có trong túi biết rằng mỗi món vũ khí có lượng sát thương riêng. BOSS có lượng máu là  $D$ , khi bạn sử dụng vũ khí có tính sát thương  $P$  tấn công BOSS thì máu của BOSS sẽ giảm đi một lượng là  $P$  (nếu  $D \leq P$  thì  $D = 0$  bạn tiêu diệt được BOSS) và sau đó tính sát thương của món vũ khí đó sẽ giảm đi một nửa (lấy phần nguyên).

Bạn có thể thay đổi vũ khí tùy ý ở mỗi lần tấn công, tuy nhiên để tăng độ khó của trò chơi thì bạn chỉ được tấn công nhiều nhất là  $K$  lần. Nếu không thể tiêu diệt BOSS sau  $K$  lần thì bạn sẽ thua.

Input:

- Dòng đầu chứa 3 số nguyên dương:  $N(1 \leq n \leq 10^5), D(1 \leq D \leq 10^{17}), K(1 \leq K \leq 10^5)$
- Dòng tiếp theo chứa  $N$  số nguyên dương là độ sát thương của  $N$  món vũ khí bạn có trong túi

Output: In ra số lượng đợt tấn công ít nhất mà bạn có thể tiêu diệt được BOSS (nhỏ hơn hoặc bằng  $K$ ), ngược lại in ra -1 nếu bạn không thể hạ BOSS sau nhiều nhất  $K$  lần tấn công.

Ví dụ:

Input 1

Output 1

3 30 7

4

10 5 13

Giải thích:

Có 3 món vũ khí trong túi (10, 5, 13)

Có thể tấn công như sau:

Lần 1, chọn vũ khí đầu tiên, BOSS còn 20 máu và độ sát thương của vũ khí còn lại trong túi là (5, 5, 13)

Lần 2, chọn vũ khí thứ ba, BOSS còn 7 máu và tập vũ khí là (5, 5, 6)

Lần 3, chọn vũ khí đầu tiên, BOSS còn 2 máu và tập vũ khí là (2, 5, 6)

Lần 4, chọn vũ khí thứ ba, BOSS bị tiêu diệt do  $D < 6$  và tập vũ khí là (2, 5, 3)

Như vậy BOSS bị tiêu diệt sau 4 lần.

Gợi ý: Có thể sử dụng hàng đợi ưu tiên

## Đề C02: Về nhà

Hòa có nhiều em gái mưa trên trường. Một ngày anh ta gặp rắc rối với hội chị em nên quyết định chạy trốn các em gái mưa. Trong lúc khẩn cấp, anh ta đã lấy cắp một chiếc xe bus và chạy về nhà. Trong khi đang lái xe thì anh ta nhận ra cần phải đổ xăng cho xe bus mà chiếc xe này cũng cũ rồi nên nó chạy 1km thì hết 1 lít xăng. Quãng đường từ trường về nhà dài  $K$  km và sẽ đi ngang qua  $N$  trạm đổ xăng. Nhưng vì vội nên Hòa muốn số lần dừng đổ xăng ít nhất có thể. Hãy giúp Hòa tìm ra số lần dừng đổ xăng ít nhất để có thể về đến nhà. Giả sử ban đầu khi lấy xe thì xe đã có sẵn  $P$  lít xăng và sức chứa của thùng xăng là vô hạn.

Input:

- Dòng đầu chứa  $T (1 \leq T \leq 10)$  là số test case
- Với mỗi test case sẽ có:

- Dòng đầu chứa 2 số nguyên dương  $K(1 \leq K \leq 10^6)$  và  $P(1 \leq P \leq 10^6)$
- Dòng thứ 2 chứa số nguyên dương  $N(1 \leq N \leq 10^4)$  là số lượng trạm đổ xăng
- Sau đó là N dòng, mỗi dòng chứa 2 số nguyên  $D(1 \leq D \leq 10^6)$  và  $A(1 \leq A \leq 10^6)$  lần lượt thể hiện khoảng cách của trạm đổ xăng so với nhà của Hòa và lượng xăng có thể đổ ở trạm này.

Output: Với mỗi test case in ra số lần dừng lại ít nhất đổ xăng để về đến nhà, ngược lại in ra -1 nếu anh ta không thể về nhà được.

Ví dụ:

Input 1	Output 1
1	3
6	
30 7	
25 10	
23 1	
15 8	
11 3	
10 7	
5 1	

Giải thích:

Hòa phải đi 30km với 7 lít xăng, anh ta sẽ dừng ở những trạm sau:

25km đổ xăng thì còn 12 lít

15km đổ xăng thì còn 10 lít

10km đổ xăng thì còn 12 lít

Gợi ý: Có thể sử dụng Max Heap

### Đề C03: Cầu hôn hay không?

Son rất muốn cầu hôn bạn gái mình nhưng anh vẫn có chút lo sợ cô ấy sẽ từ chối. Vì những suy nghĩ của mình, anh ấy đã rủ một người bạn của anh ấy cùng chơi một trò chơi may rủi như sau. Anh ta có một danh sách ra  $N$  số may mắn. Son sẽ chọn ra 2 số  $a$  và  $b$  ( $a \leq b$ ) và bạn anh ta chọn 2 số  $c$  và  $d$  ( $c \leq d$ ), trong đó  $a, b, c$  và  $d$  không cần phải thuộc danh sách các số may mắn. Son sẽ tạo ra 2 danh sách số may mắn, một danh sách chứa các số may mắn trong đoạn từ  $a$  đến  $b$  và danh sách thứ hai chứa các số may mắn trong đoạn từ  $c$  đến  $d$ . Sau đó Son đếm số phần tử giống nhau của 2 danh sách, giả sử có giá trị  $m$ . Nếu giá trị  $m \geq k$  với  $k$  cho trước thì Son sẽ tỏ tình với bạn gái, ngược lại thì anh ta sẽ đợi một dịp khác.

Vì trò chơi này hơi may rủi nên Son không hài lòng với duy nhất một kết quả. Do đó anh ấy thực hiện vài lần cho các giá trị  $a, b, c, d$  khác nhau và bạn anh ta có khi lại sửa giá trị các số trong danh sách các số may mắn của Son.

0 i y –  $A[i]=y$

1 l r – Đếm số lượng số chẵn từ vị trí thứ  $l$  đến vị trí thứ  $r$

2 l r – Đếm số lượng số lẻ từ vị trí thứ  $l$  đến vị trí thứ  $r$

#### Input:

- Dòng đầu là ba số nguyên  $N, Q, k$  ( $1 \leq N, Q, k \leq 10^5$ ) lần lượt là số lượng các số may mắn, số lượng truy vấn và số  $k$  cho trước.

- Dòng thứ 2 gồm  $N$  số may mắn ( $1 \leq \text{số may mắn} \leq 10^9$ )

- Tiếp theo  $Q$  dòng chứa một trong 2 loại truy vấn sau

0 x y – Thay đổi số may mắn thứ  $x$  bằng giá trị  $y$ ,  $1 \leq x \leq N, 1 \leq y \leq 10^9$ .

1 a b c d – In ra câu trả lời cho giá trị  $a, b, c, d$  tương ứng  $1 \leq a, b, c, d \leq 10^9$

#### Output:

- Với mỗi truy vấn loại 1, in ra Propose nếu  $m \geq k$  và Do not propose nếu ngược lại.

**Ví dụ:**

Input	Output
5 3 3	Do not propose
1 2 3 4 5	Propose
1 2 3 3 7	
0 5 6	
1 1 6 3 7	

Giải thích:

Truy vấn 1. 2 danh sách là (2,3) và (3,4,5). Số phần tử giống nhau là 1 (số 3)  $\Rightarrow m=1 < k=3$

Truy vấn 2. Cập nhật danh sách số may mắn 1 2 3 4 6

Truy vấn 3. 2 danh sách là (1,2,3,4,6) và (3,4,6). Số phần tử giống nhau là 3 (số 3,4 và 6)  $\Rightarrow m=k=3$

Gợi ý: Sử dụng Binary Indexed Tree

**Đề C04: Truy vấn chuỗi nhị phân**

Cho một chuỗi nhị phân có độ dài  $N$  và 2 loại truy vấn sau:

0 x y – In ra giá trị theo hệ thập phân của chuỗi con từ vị trí x đến y mod 3.

1 x – Nếu kí tự ở vị trí thứ x có giá trị 0 thì đổi thành 1

**Input:**

- Dòng đầu là số nguyên  $N (1 \leq N \leq 10^5)$  là độ dài của chuỗi.
- Dòng thứ 2 chứa chuỗi nhị phân độ dài  $N$

- Dòng thứ 3 chứa số nguyên  $Q (1 \leq Q \leq 10^5)$  là số lượng truy vấn
- Tiếp theo  $Q$  dòng chứa một trong 2 loại truy vấn,  $0 \leq x \leq y < N$ .

**Output:**

- Với mỗi truy vấn loại 0 in ra giá trị theo yêu cầu mod 3

**Ví dụ:**

Input	Output
5	2
10010	1
6	2
0 2 4	1
0 2 3	
1 1	
0 0 4	
1 1	
0 0 3	

**Giải thích:**

Truy vấn 1. Chuỗi nhị phân 010 có giá trị 2 và  $2\%3=2$

Truy vấn 2. Chuỗi nhị phân 01 có giá trị 1 và  $1\%3=1$

Truy vấn 3. Đổi giá trị tại vị trí 1 từ 0 thành 1. Ta được 11010

Truy vấn 4. Chuỗi nhị phân 11010 có giá trị là 26 và  $26\%3=2$

Truy vấn 5. Vị trí 1 đã có giá trị 1 nên không thay đổi. Chuỗi vẫn là 11010

Truy vấn 6. Chuỗi nhị phân 1101 có giá trị  $13$  và  $13\%3=1$

Gợi ý: Sử dụng Binary Indexed Tree

### Đề C05: Bạn muốn hẹn hò?

Chương trình Bạn muốn hẹn hò đã trở nên quá phổ biến trên truyền hình, và ở bách khoa cũng đã có chương trình tương tự như vậy với quy mô nhỏ hơn. Trong đợt này có  $X$  bạn nam và  $Y$  bạn nữ đăng ký tham gia. Ban tổ chức có thông tin của tất cả người tham gia, kể cả trang cá nhân Facebook nên có thể biết được hai sinh viên có phải là bạn của nhau trên Facebook hay không. Và ban tổ chức không muốn ghép cặp hai sinh viên ở trong cùng một nhóm bạn với nhau. Cho  $q_1$  mỗi quan hệ bạn bè giữa các bạn sinh viên nam với nhau,  $q_2$  mỗi quan hệ bạn bè giữa các bạn nữ với nhau và  $q_3$  mỗi quan hệ bạn bè giữa nam và nữ. Ban tổ chức muốn tính xem tổng cộng có bao nhiêu cặp sinh viên nam nữ có thể ghép cặp được với nhau.

**Note:** 2 sinh viên được xem là trong cùng một group nếu 2 sinh viên này là bạn bè của nhau trên Facebook hoặc kết nối với nhau qua một chuỗi bạn chung.

#### Input:

- Dòng đầu là số 2 nguyên  $X, Y (1 \leq X, Y \leq 10^6)$  là số lượng sinh viên nam và số lượng sinh viên nữ tham gia chương trình
- Dòng tiếp theo chứa  $q_1$ , theo sau là  $q_1$  dòng, mỗi dòng gồm 2 số  $A$  và  $B$  thể hiện bạn nam  $A$  và bạn nam  $B$  là bạn của nhau trên Facebook ( $A \neq B$ )
- Dòng tiếp theo chứa  $q_2$ , theo sau là  $q_2$  dòng, mỗi dòng gồm 2 số  $A$  và  $B$  thể hiện bạn nữ  $A$  và bạn nữ  $B$  là bạn của nhau trên Facebook ( $A \neq B$ )
- Dòng tiếp theo chứa  $q_3$ , theo sau là  $q_3$  dòng, mỗi dòng gồm 2 số  $A$  và  $B$  thể hiện bạn nam  $A$  và bạn nữ  $B$  là bạn của nhau trên Facebook

#### Output:

- In kết quả theo yêu cầu đề bài.

#### Ví dụ:

Input	Output
4 5	15
1	
1 3	
2	
1 4	
1 5	
2	
1 2	
4 1	

Giải thích:

Có 4 nhóm:

Nhóm 1 – Nam 1, Nam 3, Nữ 2

Nhóm 2 – Nam 4, Nữ 1, Nữ 4, Nữ 5

Nhóm 3 – Nam 2

Nhóm 4 – Nữ 3

Hai sinh viên nam trong Nhóm 1 có thể được ghép cặp với 4 bạn nữ ở các nhóm còn lại nên có 8 khả năng

Một sinh viên nam trong Nhóm 2 có thể ghép cặp với 2 bạn nữ ở các nhóm còn lại nên có 2 khả năng

Một sinh viên trong Nhóm 3 có thể ghép cặp với 5 bạn sinh viên nữ trong các nhóm còn lại nên có 5 khả năng



Gợi ý: Sử dụng Disjoint Data Structures

The End.