

Q1. Đọc đoạn văn bản sau liên quan đến việc thực hiện một lệnh, rồi trả lời câu hỏi con.

Một máy tính có dung lượng nhớ chính là 65.536 từ, mỗi từ gồm 16 bits. Nó có bốn thanh ghi vụn nắn (từ 0 đến 3) và một thanh ghi chương trình (“PR”). Khuôn dạng các từ lệnh (chiều dài 2 từ) như sau:

OP	R	X	I	D	adr
8 bit	2 bit	2 bit	1 bit	3 bit	16 bit

OP: Biểu diễn mã lệnh - gồm 8 bits.

Trong ví dụ này, ba mã lệnh sau được sử dụng.

20₁₆: Đặt địa chỉ hiệu dụng vào trong thanh ghi vụn nắn được biểu diễn bởi R.

21₁₆: Đặt nội dung của từ được chỉ ra bởi địa chỉ hiệu dụng vào trong thanh ghi vụn nắn được biểu diễn bởi R.

FF₁₆: Kết thúc thực hiện

R: Biểu diễn số hiệu của thanh ghi vụn nắn (từ 0 đến 3) - hai bit

X: Biểu diễn số hiệu của thanh ghi chỉ số (từ 1 đến 3) - hai bit. Thanh ghi vụn nắn tại số hiệu này được sử dụng làm thanh ghi chỉ số. Tuy nhiên nếu X có giá trị “0”, không có sự biến đổi nào theo thanh ghi chỉ số được thực hiện

I: Có giá trị “1” biểu diễn bằng một bit trong trường hợp địa gián tiếp, trong trường hợp khác có giá trị “0”

D: Ba bit mở rộng, luôn có giá trị “0”

adr: Biểu diễn địa chỉ bằng 16 bits

Địa chỉ hiệu dụng của từ lệnh được tính toán theo bảng sau

Bảng quan hệ giữa X,I, và địa chỉ hiệu dụng

X	I	Địa chỉ hiệu dụng
0	0	Adr
1 đến 3	0	Adr+(X)
0	1	(adr)
1 đến 3	1	(adr+(X))

Chú ý (): Ngỏc đơn biểu diễn thông tin lưu trong thanh nghi hoặc địa chỉ nằm trong ngoặc

Câu hỏi con

Từ nhóm câu trả lời sau, chọn các câu trả lời đúng để điền vào các chỗ trống từ đến trong đoạn văn bản sau.

Khi Nội dung của các thanh ghi vận năng và nội dung bộ nhớ chính có giá trị nêu trong hình vẽ VD: giá trị 0100_{16} đặt trong PR và chương trình được thực hiện.

Trong ví dụ này, lệnh đặt tại địa chỉ 0100_{16} (mã lệnh lưu ở địa chỉ 0100) thực hiện đặt nội dung 0113_{16} của địa chỉ $011B_{16}$ (gạch dưới) vào thanh ghi vận năng 0.

Sau khi kết thúc thực hiện, được đặt vào thanh ghi vận năng 0, được đặt vào thanh ghi vận năng 1, được đặt vào thanh ghi vận năng 2, và được đặt vào thanh ghi vận năng 3.

Thanh ghi vận năng 0: 0003 1: 0000 2: 0000 3: 0000

Thanh ghi chương trình: PR: 0100

Bộ nhớ chính

Địa chỉ	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	<u>0113</u>	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

Hình: Các giá trị trong các thanh ghi vận năng, PR và bộ nhớ chính

Nhóm câu trả lời:

a) 0000_{16}

b) 0001_{16}

c) 0002_{16}

d) 0003_{16}

e) 0004_{16}

f) 0005_{16}

g) 0006_{16}

h) 0113_{16}

i) 0115_{16}

Phần lời giải

Câu hỏi con	Đáp án đúng
a	h) 0113
b	e) 0004
c	f) 0005
d	d) 0003

Các bước thực hiện có thể tóm tắt như sau:

1. Tính địa chỉ hiệu dụng dựa trên Bảng 1: Địa chỉ hiệu dụng được tính dựa theo giá trị của X và I. Hai giá trị này được tính dựa trên từ lệnh như trong mô tả của đề bài.
2. Thực hiện phép toán dựa vào mã lệnh: Trong bài này chỉ có 2 mã lệnh là 20H và 21 và cách tính địa chỉ hiệu dụng (Theo Bảng 1 Quan hệ giữa X, I và địa chỉ hiệu dụng).
3. Thanh ghi đích được xác định trong R.
4. Chuyển nội dung trực tiếp hoặc gián tiếp vào thanh ghi R

Các lệnh bắt đầu thực hiện từ ô 0100 như sau:

Mã lệnh	R 2bit	X 2bit	I 1bit	D 3bit	adr	Địa chỉ hiệu dụng (Theo Bảng 1)	Hành động thực hiện (Theo mô tả và theo Hình 1)	Đáp án đúng cho các ô A đến D
2100 011B	0 R0	0	0	0	011B	adr	(011B) → R0 hay 0113 → R0	ô A: h) 0113
20C0 0003	11R3	0	0	0	0003	adr	0003 → R3	ô D: d) 0003
2170 0111	01R1	11	0	0	0111	adr + (X)	Tính địa chỉ hiệu dụng: $0111 + (X) = 0111$ $+(R3) = (0111 + 3 = 0114$ Thực hiện phép toán: (0114) → R1 hay 0004 → R1	ô B: e) 0004
21B8 011A	10 R2	11	1	0	011A	(adr + (X))	Tính địa chỉ hiệu dụng: $(011A + (X)) =$ $(011A + (R3)) = (011A$ $+ 3) = (011D) = 0115$ Phép toán thực hiện: (0115) → R2 0005 → R2	ô C: f) 0005

Tại ô nhớ 0108, chương trình gặp lệnh FFH và dừng.

Q2: Hãy đọc đoạn văn bản sau liên quan đến việc sử dụng các biểu thức, rồi trả lời các Câu hỏi con từ 1- 3.

Có một sổ cái (ledger) dùng để quản lý thông tin của tệp văn bản và đồ hoạ. Như trong ví dụ trong bảng 1. Thông tệp bao gồm bốn tham số (tên tệp, kiểu mã, mã phiên bản, ngày tạo), và mỗi mục là một xâu kí tự. Việc tìm kiếm được thực hiện trên mỗi mục trong sổ cái. Điều kiện tìm kiếm được xác định như các biểu thức.

Bảng 1: Ví dụ của sổ cái

Tên tệp	Kiểu mã	Phiên bản mã	Ngày tạo
LOGO-T01	JPG	V/R100	2002-01-15
TITLE-A1	GIF	V/R100	2002-01-15
REP-JP01	HTML	V/R203	2002-01-22
OPINION3	TXT	V/R103	2003-02-05

- (1) Các xâu kí tự của các mục bao gồm các chữ cái in hoa, số, gạch dưới, gạch nghiêng được xác định bởi tập kí tự mã trao đổi thông tin ASCII.
- (2) Bảng 2 dưới đây thể hiện các siêu kí tự sử dụng trong biểu thức của các điều kiện tìm kiếm. Siêu kí tự (meta-character) là những kí tự dùng để thể hiện cú pháp của một biểu thức.

Bảng 2: Các siêu kí tự dùng trong các biểu thức

Siêu kí tự	ý nghĩa	Biểu thức ví dụ	Giải thích ví dụ
.	Thể hiện một kí tự bất kỳ	M..N	Xâu kí tự gồm 4 kí tự, bắt đầu với “M” và kết thúc “N”
(Xâu kí tự)	Xâu kí tự được bao bởi các ngoặc đơn trái và phải, được coi như mẫu(pattern)	(MO)	Mẫu kí tự gồm xâu kí tự “MO”
+	Thể hiện một hoặc lặp lại nhiều lần của một kí tự hoặc mẫu đứng trước.	ABX+	AB và một hoặc lặp lại của nhiều lần của X (ví dụ: ABX, ABXX, ABXXX)
		(MO)+	Một hoặc lặp lại nhiều lần của MO. Ví dụ: MO, MOMO
*	Thể hiện 0 hoặc lặp lại nhiều lần của kí tự hoặc mẫu đứng trước.	ABX*	AB và 0 hoặc lại nhiều lần của X. Ví dụ: AB, ABX, ABXX
?	Chỉ ra kí tự hoặc mẫu đứng trước xuất hiện 0 hoặc 1 lần	ABCD?	ABC hoặc ABCD

\	Kí tự tiếp theo không phải là siêu kí tự mà chỉ là kí tự thường.	AB*	Xâu kí tự AB*
	Thể hiện lựa chọn kí tự hoặc mẫu.	A B (AB) (CDE)	A hoặc B mẫu “AB” hoặc “CDE”
[m-n]	Thể hiện lựa chọn một kí tự bất kỳ trong nhóm các kí tự liên tiếp từ “m” đến “n”	[3-5] [W-Z]	Kí tự bất kỳ (3,4,5) trong các nhóm kí tự liên tiếp Kí tự bất kỳ (W,X,Y,Z) trong nhóm các kí tự liên tiếp.

Câu hỏi con 1

Các mã phiên bản của tệp được ghi trong sổ cái theo cú pháp sau đây.

Mã phiên bản được bắt đầu với 3 kí tự “V|R”, tiếp theo sau là một chữ số (từ 1 đến 9) thể hiện số hiệu của phiên bản và kết thúc với 2 chữ số (00 đến 99) thể hiện số hiệu con của phiên bản. Ví dụ, nếu số hiệu phiên bản là 1 và số hiệu con là 03, thì mã phiên bản là “V|R103”.

Trong nhóm câu trả lời sau đây, hãy chọn câu trả lời **không đúng** cho việc tìm kiếm mã phiên bản bằng cách dùng các biểu thức chuẩn tắc.

Nhóm các câu trả lời:

- Sử dụng “01” để lấy ra các tệp có mã con là 01.
- Sử dụng “R..1” để lấy ra các tệp có mã con là 01.
- Sử dụng “R[1-3]” để lấy ra các tệp có số hiệu phiên bản là 3 hoặc nhỏ hơn.
- Sử dụng “302” để lấy ra tệp có số hiệu phiên bản và số hiệu con tương ứng là 3 và 02.
- Sử dụng “V|R302” để lấy ra tệp có số hiệu phiên bản và số hiệu con tương ứng là 3 và 02.

Câu hỏi con 2

Các mã kiểu (các xâu kí tự có 3 hoặc 4 kí tự) chỉ ra tệp được ghi trong sổ cái. Có 8 kiểu các nhau như sau:

GIF	HTM	HTML	JPG	JPEG	JPN	MPEG	TXT
-----	-----	------	-----	------	-----	------	-----

Các tệp có mã kiểu JPG hoặc JPEG cần được lấy ra. Trong nhóm các câu trả lời sau, hãy chọn câu trả lời là biểu thức chính xác được dùng để tìm kiếm các mã kiểu này

Nhóm các câu trả lời :

- | | | |
|----------|----------|----------|
| a) JPEG? | b) JPEG* | c) JPE?G |
| d) JP+G | e) JP?G | f) JP.G |

Câu hỏi con 3

Các ngày tạo tệp được ghi lại trong sổ cái. Ngày tạo là một xâu kí tự chứa 4 chữ số (từ 0001 đến 9999) thể hiện năm, tiếp theo sau là một số có hai chữ số (1 đến 12) thể hiện tháng, tiếp theo nữa là số có hai chữ số (01 đến 31) thể hiện ngày, với dấu gạch ngang để nối các số này lại với nhau. Ví dụ: Ngày 20 tháng 3 năm 2003 được viết “2003-03-20”. Việc tìm kiếm ngày tạo tệp được thực hiện bằng cách dùng biểu thức dưới đây. Trong nhóm các câu trả lời sau, chọn ra tất cả các ngày được lấy ra bằng cách tìm kiếm sử dụng biểu thức chuẩn tắc:

..(0(1|2)\-)+.1

Nhóm các câu trả lời

- | | | |
|---------------|---------------|---------------|
| a) 2001-02-10 | b) 2001-12-11 | c) 2002-02-21 |
| d) 2002-11-10 | e) 2002-12-01 | f) 2003-01-10 |

Phần lời giải

Câu hỏi con	Đáp án đúng
1	b) Sử dụng “R..1” để lấy ra các tệp có mã con là 01
2	c) JPE?G
3	c) 2002-02-21 và d) 2002-11-10

Câu hỏi con 1

Nhóm câu trả lời cho Câu hỏi con 1 như sau, phân tích từng đáp án để chọn ra đáp án sai

- a) *Sử dụng “01” để lấy ra các tệp có mã con là 01.* → Đáp án đúng, vì mã phiên bản cần tìm là bất kì mã nào chứa 2 kí tự liền nhau “01”.
- b) *Sử dụng “R.. 1” để lấy ra các tệp con có mã là 01.* → sai, vì “..” sẽ chắc chắn lấy ra tất cả các số từ 0 cho đến 9; ngoài ra trước số 1 chứa dấu cách do vậy sẽ không có mã con 01 nào được lấy. → đáp án
- c) *Sử dụng “R[1-3]” để lấy ra các tệp có số hiệu phiên bản là 3 hoặc nhỏ hơn* → đúng, vì xâu cần tìm sẽ bắt đầu bởi chữ cái R sau đó là 1 kí tự được lấy trong khoảng từ 1 đến 3
- d) *Sử dụng “302” để lấy ra tệp có số hiệu phiên bản và số hiệu con tương ứng là 3 và 02* → đúng, vì xâu cần tìm là bất kỳ trong đó có chứa 3 kí tự liền nhau “302”
- e) *Sử dụng “V/R302” để lấy ra tệp có số hiệu phiên bản và số hiệu con tương ứng là 3 và 02* → đúng, vì xâu cần tìm là bất kỳ trong đó có chứa 6 kí tự liền nhau “V/R302”

Câu hỏi con 2

Nhóm câu trả lời cho Câu hỏi con 2 như sau, phân tích từng đáp án để chọn ra đáp án đúng

- a) $JPEG?$ \rightarrow sai, cho kết quả tìm kiếm là: JPE hoặc JPEG vì "?" cho biết kí tự đứng trước (G) xuất hiện 0 hoặc 1 lần.
- b) $JPEG^*$ \rightarrow sai, cho kết quả tìm kiếm là: JPE, JPEG, JPEGG, JPEGGG, ... vì * cho biết kí tự đứng trước (G) xuất hiện 0 hoặc nhiều lần
- c) $JPE?G$ \rightarrow đúng, cho kết quả tìm kiếm là: JPG hoặc JPEG vì ? cho biết kí tự đứng trước (E) xuất hiện 0 hoặc 1 lần
- d) $JP+G$ \rightarrow sai, cho kết quả tìm kiếm là: JPG, JPPG, JPPPG, ... vì + cho biết kí tự đứng trước (P) xuất hiện 1 hoặc lặp lại nhiều lần
- e) $JP?G$ \rightarrow sai, cho kết quả tìm kiếm là: JP hoặc JPG vì ? cho biết kí tự đứng trước (P) xuất hiện 0 hoặc 1 lần
- f) $JP.G$ \rightarrow sai, cho kết quả tìm kiếm gồm 4 kí tự, bắt đầu là JP và kết thúc là G

Câu hỏi con 3

.. $(0(1|2)\backslash-)+.1$ có thể hiểu:

- 1) .. chắc chắn sẽ có 2 kí tự đứng trước, theo sau sẽ là
- 2) $(0(1|2)\backslash-)+$ thể hiện 1 hoặc lặp lại nhiều lần xâu đứng trước dấu + trong đó
 - 0: chắc chắn sẽ có 1 số 0
 - $(1|2)\backslash-$: đi sau 0 sẽ là 1 hoặc 2 và sau đó là dấu gạch nối – (vì sau kí tự \ thì kí tự tiếp theo không phải là siêu kí tự)

như vậy tại 2) có thể có các kết quả: 01- 01-; 01-02-; 02-01-; 02-02-

- 3) .1 thể hiện trước số 1 sẽ là 1 số bất kỳ

Như vậy, ghép các khoản mục 1), 2) và 3), ta có:

Biểu thức chuẩn tắc	..	$(0(1 2)\backslash-)+$.	1	Kết quả
Đáp án c	20	02-02-	2	1	2002-02-21
Đáp án d	20	02-	1	1	2002-11

- Q3.** Hãy đọc đoạn văn bản sau liên quan đến điều khiển truy cập LAN, rồi trả lời câu hỏi con.
- CSMA/CD (Carrier Sense Multiple Access/ Collision Detect) là hệ thống điều khiển truy cập được sử dụng trong các mạng LAN hình sao sử dụng cáp đồng trục (coaxial cable), và trong các mạng LAN hình sao sử dụng hai cáp 2 dây xoắn và hubs.
- Với CSMA/CD, thủ tục sau được sử dụng trong truyền khung (frame transmission) để nhận biết tín hiệu mang và phát hiện xung đột giữa các thiết bị được nối tới đường truyền.

[Mô tả thủ tục truyền khung]

Bước 0: Chờ cho đến khi một khung được truyền. Khi có một khung được truyền, nhảy tới bước 1

Bước 1: Nếu tín hiệu sóng mang của thiết bị khác trong mạng được nhận biết trên đường truyền, nhảy đến bước 2, nếu không nhảy đến bước 3.

Bước 2: Khi chu kỳ thời gian xác định ngẫu nhiên đã trôi qua, quay lại bước 1.

Bước 3: Bắt đầu truyền khung và nhảy tới bước 4.

Bước 4: Nếu không có xung đột khi truyền khung, việc truyền coi như thành công. Quay lại bước 0. Nếu phát hiện xung đột trong quá trình truyền khung, nhảy tới bước 5. Việc kiểm tra này được thực hiện nhằm xác định xem có xung đột trong quá trình truyền khung hay không, bởi vì có khả năng là việc bắt đầu truyền khung từ một thiết bị khác trong mạng đã không được phát hiện trong bước 1 do độ trễ lan truyền tín hiệu trên đường truyền.

Bước 5: Chuyển từ khung đang được truyền sang tín hiệu thông báo cho các thiết bị khác sự xuất hiện xung đột. Sau khi truyền tín hiệu này trong khoảng thời gian xác định, quay lại bước 2. Tín hiệu này cho phép các thiết bị khác trong mạng biết rằng xung đột đã xảy ra.

Câu hỏi con

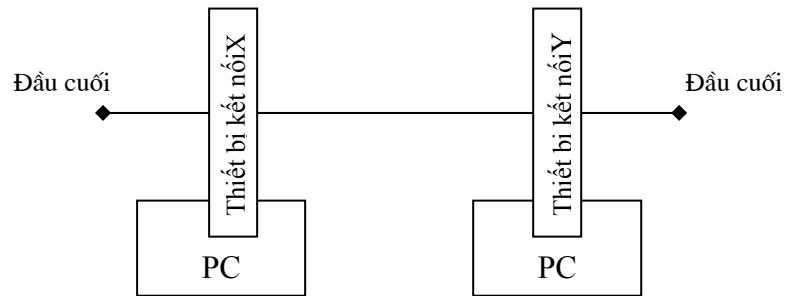
Từ nhóm các câu trả lời dưới đây, chọn các câu trả lời đúng để điền vào các chỗ trống từ đến trong đoạn văn bản sau.

Có một mạng LAN được nối như trong sơ đồ sau. Khoảng cách đường truyền giữa thiết bị kết nối X và thiết bị kết nối Y là 230m. Tốc độ lan truyền tín hiệu trên đường truyền là 230 m trong một micro giây.

Việc truyền khung từ thiết bị X được bắt đầu. Trước khi trôi qua micro giây từ lúc bắt đầu truyền, một khung đang được truyền từ thiết bị Y. Khi điều này xảy ra, thiết bị Y chuyển qua bước 1 và nhảy tới . Nhờ đó, thiết bị Y nhận biết được xung đột.

Khoảng thời gian cần tối đa cho thiết bị X để nhận biết xung đột là xấp xỉ micro giây sau khi bắt đầu truyền khung từ thiết bị X phụ thuộc vào thời gian chênh lệch so với thời điểm

bắt đầu truyền từ thiết bị Y, là thiết bị bắt đầu truyền sau thiết bị X. Nếu việc truyền từ thiết bị X kết thúc trước khi khoảng thời gian này trôi qua, thì thiết bị X sẽ đi từ **d** đến **e** và sẽ không thể phát hiện ra xung đột. Trong trường hợp này, giả thiết rằng các thiết bị kết nối có thể gắn vào bất kỳ vị trí nào ở khoảng giữa các thiết bị đầu cuối lắp tại hai đầu, thì rõ ràng với phương pháp nhận biết này, thời gian cần có để truyền một khung đơn (single frame) ít nhất cũng phải bằng thời gian tín hiệu đi một vòng giữa các thiết bị đầu cuối trên đường truyền.



Kết nối mạng Lan Connections bằng phương pháp CSMA/CD

Nhóm các câu trả lời cho a và c

- | | | |
|-----------|--------|---------|
| a) 0.0043 | b) 0.2 | c) 1 |
| d) 2 | e) 10 | f) 4300 |

Nhóm các câu trả lời cho b, d, e

- | | | |
|-----------|-----------|-----------|
| a) Bước 0 | b) Bước 1 | c) Bước 2 |
| d) Bước 3 | e) Bước 4 | f) Bước 5 |

Phần lời giải

Câu hỏi con	Đáp án đúng
A	c) 1
B	d) Bước 3
C	d) 2
D	e) Bước 4
E	a) Bước 0

Độ dài từ X đến Y là 230 m, và thời gian truyền là $1\mu s$ (vì $rate = 230m/\mu s$).

Trong vòng $1\mu s$, khung truyền chưa tới Y, nên một khung đang được truyền từ thiết bị Y (Y chưa cảm nhận được sóng mang của khung dữ liệu truyền từ X). Nếu điều này xảy thì Y chuyển qua Bước 1 và nhảy đến bước 3. Đáp án đúng “d) Bước 3” cho ô B.

Khoảng thời gian cần có tối đa cho thiết bị X để nhận biết xung đột là xấp xỉ “**d) 2**” micro giây sau khi bắt đầu truyền khung từ thiết bị X phụ thuộc vào thời gian chênh lệch so với thời điểm bắt đầu truyền từ thiết bị Y, là thiết bị bắt đầu truyền sau thiết bị X. Đáp án d) đúng cho ô C.

- Xấp xỉ 1 micro giây để một khung truyền từ X đến Y
- Xấp xỉ 1 micro giây nữa để một khung truyền từ Y đến X.

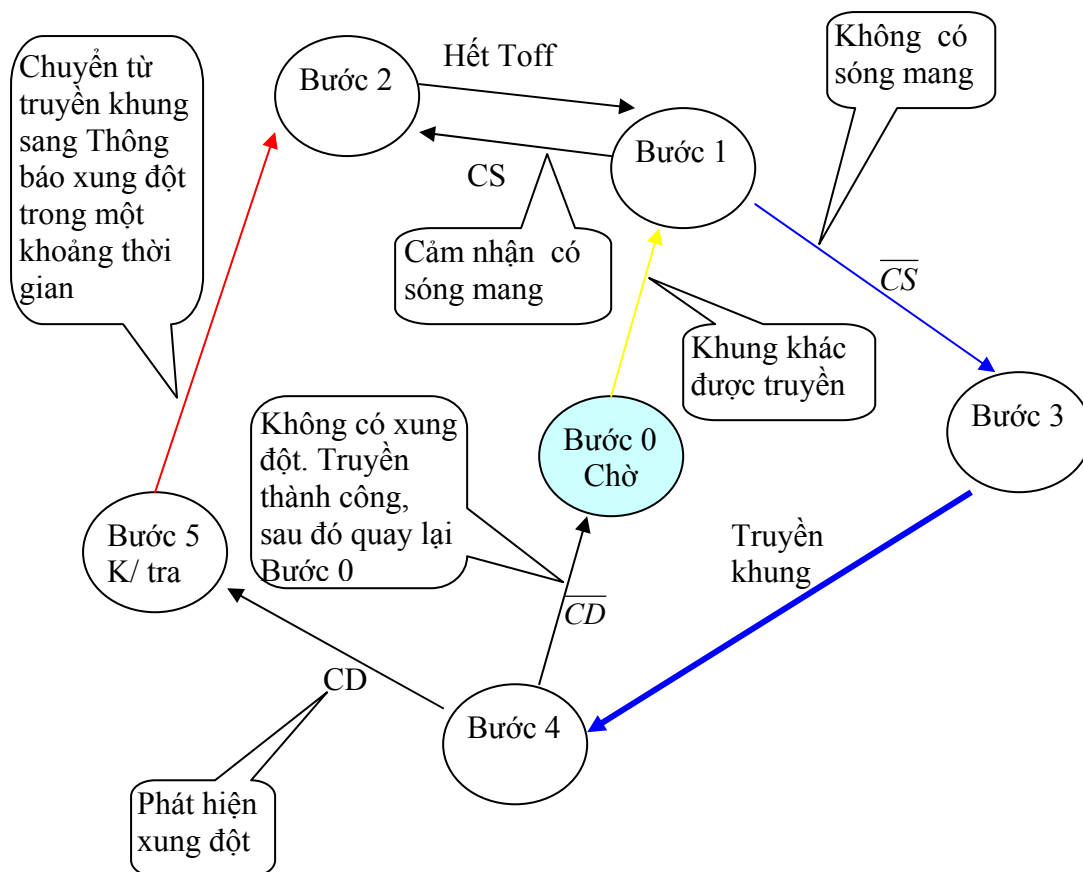
Do vậy trong vòng xấp xỉ 2 micro giây kể từ khi X truyền, và nó nhận biết sóng mang (phát hiện xung đột).

Nếu việc truyền từ X kết thúc trước khoảng thời gian này (2 micro giây), thì việc truyền hoàn thành thì X sẽ đi từ Bước 4 (ô **D là Bước 4**) đến Bước 0 (ô **E là Bước 0**) chờ để truyền tiếp tục.

♣ Trong các mạng máy tính khác nhau, **binary exponential backoff** or **truncated binary exponential backoff** liên quan tới thuật toán đặt cách quãng việc truyền lại được lặp lại cùng một khối dữ liệu. Những thí dụ về truyền lại các gói tin (packets) trong mạng Đa truy nhập cảm nhận sóng mang/Tránh xung đột (carrier sense multiple access with collision avoidance (CSMA/CA)) và Đa truy nhập cảm nhận sóng mang/Phát hiện xung đột (carrier sense multiple access with collision detection (CSMA/CD)), trong đó thuật toán này là một phần của p/p truy nhập kênh được sử dụng để gửi dữ liệu trên mạng này. Trong mạng Ethernet thuật toán được sử dụng để lập lịch biểu truyền lại sau một xung đột mà truyền lại bị trễ đi một khoảng thời gian được tính theo khe thời gian (time slot) và số lần thử truyền lại. Sau i xung đột, một số ngẫu nhiên khe thời gian giữa 0 và $2^i - 1$ được chọn. Như vậy, đối với xung đột đầu tiên, mỗi máy phát có khi phải đợi 0 hoặc 1 khe thời gian. Sau lần xung đột thứ hai, các máy phát có thể phải đợi 0, 1, 2, hoặc 3 khe thời gian, và tiếp tục như vậy... Do các lần thử truyền lại tăng, nên số các khả năng trễ tăng.

Thuật ngữ 'truncated' chỉ có nghĩa sau một số lần tăng; có nghĩa ngưỡng thời gian truyền lại đạt tới điểm trần và sẽ không tăng nữa. Trần được đặt là $i = 10$, do vậy trễ cực đại là 1023 khe thời gian.

Do những trễ gây cho những trạm này mà chúng cũng gửi để xảy ra xung đột, có khả năng rằng trong một mạng bận, hàng trăm người có thể “chộp được” trong một tập xung đột đơn. Do vậy sau 16 lần thử truyền lại quá trình dừng.



Biểu đồ chuyển trạng thái cho các trạng thái Bước 1- 5.

Q4. Hãy đọc mô tả chương trình, mô tả cú pháp giả ngữ, và chương trình dưới đây rồi trả lời câu hỏi con.

[Mô tả chương trình]

Calc là chương trình con sử dụng một ngăn xếp để tính các biểu thức số được biểu diễn theo kí pháp Ba Lan đảo ngược (Reverse Polish Notation).

- (1) Các biểu thức số được biểu diễn theo kí pháp Ba Lan đảo ngược được lưu giữ mỗi lần một kí tự, trong các phần tử riêng biệt $Ex[0], \dots, Ex[Lp]$ ($Lp \geq 3$) của mảng một chiều kiểu kí tự **Ex**.
- (2) Biểu thức số gồm số nguyên dương hoặc nguyên âm và một hoặc nhiều kí hiệu phép toán số học. Nếu số nguyên là dương, thì dấu cộng được thêm vào nó, trừ số nguyên đầu tiên.
- (3) Một dấu cách (trống) được đặt trước các số nguyên, trừ số nguyên đầu tiên.
- (4) Các tính toán được thực hiện bằng cách dùng các số thực. Chương trình con **Abort** () được gọi để thoát khỏi chương trình nếu một trong hai trạng thái sau xảy ra trong quá trình tính toán.
 - ① Thực hiện việc chia cho 0
 - ② Có gì đó bên ngoài ngăn xếp được tham chiếu đến.
- (5) **Calc** sử dụng trong chương trình con **Push**, chương trình này thêm các số thực vào ngăn xếp và chương trình con **Pop**, chương trình này lấy các số thực ra khỏi ngăn xếp. Các bảng 1 đến 3 dưới đây cho thấy đặc tả của các đối số đối với mỗi chương trình con. Ngoài ra, hàm **ToReal** cũng được sử dụng, hàm này chuyển đổi một số kí tự số đơn thành một số thực.
- (6) Các biến sau đây được định nghĩa như các biến toàn cục (global variable): **Stack**, mảng một chiều kiểu số thực; **MAX**, một hằng diễn tả số lượng phần tử lớn nhất trong **Stack**; và **Sp**, một biến cho biết vị trí đang được xử lý trong ngăn xếp. Giá trị ban đầu của **Sp** là "0".
- (7) Giả thiết rằng các biểu thức số được diễn tả theo cách ghi Polish đảo là đúng.

Ví dụ: Biểu thức số
trong cách ghi trung tố Biểu thức số trong kí pháp Ba Lan đảo ngược

Ví dụ 1:
(1 + 2) * (3 - 4) Ex

0	1	2	3	4	5	6	7	8	9
1	Δ	2	+	Δ	3	Δ	4	-	*

 $Lp = 9$

Ví dụ 2:
12 / (-345)

0 1 2 3 4 5 6 7
Ex

1	2	Δ	-	3	4	5	/
---	---	---	---	---	---	---	---

 Lp = 7

Ghi chú : Các hình tam giác là kí hiệu cho các dấu trống

Bảng 1. Đặc tả đối số của Calc

Tên biến	Vào/ra	Ý nghĩa
Ex[]	Vào	Mảng một chiều kiểu kí tự chứa biểu thức số được diễn tả theo cách ghi Polish đảo
Lp	Vào	Số hiệu của phần tử cuối cùng trong mảng một chiều kiểu kí tự Ex ($Lp \geq 3$)
Ret	Ra	Kết quả tính toán

Bảng 2. Đặc tả đối số của Push

Tên biến	Vào/Ra	Ý nghĩa
T	Vào	Số thực được thêm vào ngăn xếp

Bảng 3. Đặc tả đối số của Pop

Tên biến	Vào/Ra	ý nghĩa
T	Vào	Số thực được lấy ra khỏi ngăn xếp

[Mô tả cú pháp giả ngữ]

Cú pháp	Mô tả
o	Khai báo tên, kiểu, ... của các thủ tục, biến,
• Biến ← biểu thức	Gán giá trị của biểu thức vào biến
{Văn bản}	Văn bản là chú thích
<div style="display: inline-block; vertical-align: middle;"> ↑ Biểu thức điều kiện • Tiến trình 1 ─── ↓ • Tiến trình 2 </div>	Chỉ ra một quy trình tuyển chọn. Nếu biểu thức điều kiện là true, thì Tiến trình1 được thực hiện; nếu nó là False, thì Tiến trình 2 được thực hiện
<div style="display: inline-block; vertical-align: middle;"> ■ Biểu thức điều kiện • Tiến trình ─── ■ </div>	Chỉ ra vòng lặp với điều kiện kết thúc đặt lên đầu. Nếu biểu thức điều kiện là true, thì quy trình được thực hiện.

[Chương trình]

O Subprogram name: Calc(Ex[], Lp, Ret)

O Character type: Ex[]

O Integer type: Lp, Cp

O Real number type: Ret, X, Y, T

O logic type: NumF, NegF

• **Cp** \leftarrow 0

• **T** \leftarrow 0

• **NumF** \leftarrow False

• **NegF** \leftarrow False

■ **Cp** \leq Lp

↑ **Ex[Cp] = Number**

• **T** \leftarrow **T** * 10 + **ToReal** (**Ex** [**Cp**])

{ToReal () là một hàm dùng để biến đổi một chữ số đơn thành một số thực}

• **NumF** \leftarrow true

↑ **a**

↑ **NegF = True**

• **T** \leftarrow -**T**

• **NegF** \leftarrow false

• **Push** (**T**)

• **b**

↓ • **NumF** \leftarrow false

↑ **Ex** [**Cp**] = "+"

• **Pop** (**Y**)

• **Pop** (**X**)

• **T** \leftarrow **X** + **Y**

• **Push** (**T**)

• **T** \leftarrow 0

↑ **Ex**[**Cp**] = "-"

↑ **Cp** \neq Lp AND **Ex** [**Cp** + 1] = Number

• **c**

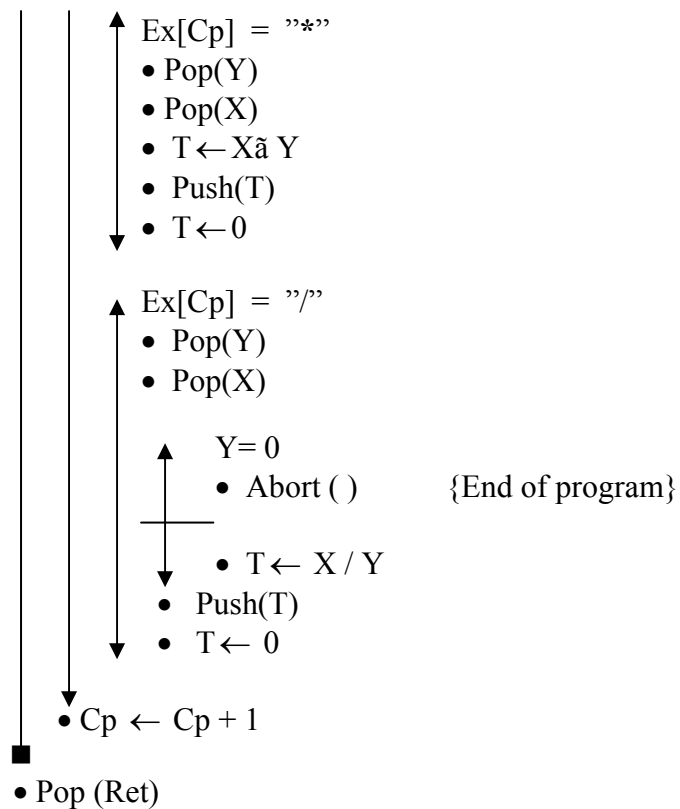
• **Pop**[**Y**]

• **Pop** [**X**]

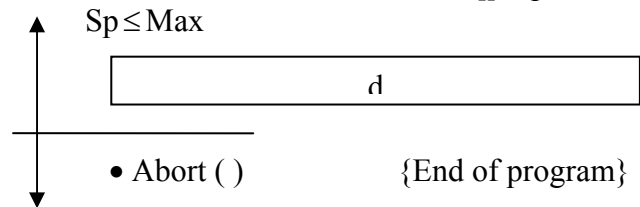
• **T** \leftarrow **X** - **Y**

• **Push**(**T**)

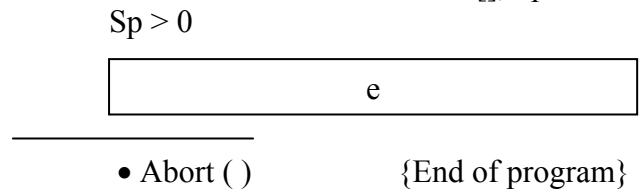
↓ • **T** \leftarrow 0



- o Subprogram name: Push (T)
- o Real number type: T
- o External reference: Stack [], Sp



- o Sub program name: Pop (T)
- o Real number type: T
- o External reference: Stack [], Sp



Câu hỏi con

Từ các nhóm câu trả lời dưới đây, hãy chọn các câu trả lời đúng để điền vào các chỗ trống từ đến trong chương trình trên.

Nhóm câu trả lời cho a:

- a) $Ex[Cp] = \text{'' ''}$
- b) $NumF = \text{false}$
- c) $NumF = \text{true}$
- d) $T = 0$
- e) $T \neq 0$

Nhóm câu trả lời cho b và c:

- a) $NegF \leftarrow \text{False}$
- b) $NegF \leftarrow \text{true}$
- c) $NumF \leftarrow \text{False}$
- d) $NumF \leftarrow \text{true}$
- e) $T \leftarrow 0$

Nhóm câu trả lời cho d và e:

- a) • $Sp \leftarrow Sp + 1$
- b) • $Sp \leftarrow Sp + 1$
• $T \leftarrow \text{Stack}[Sp]$
- c) • $Sp \leftarrow Sp - 1$
• $\text{Stack}[Sp] \leftarrow T$
- d) • $Sp \leftarrow Sp - 1$
• $\text{Stack}[Sp] \leftarrow T$
- e) • $\text{Stack}[Sp] \leftarrow T$
• $Sp \leftarrow Sp + 1$
- f) • $\text{Stack}[Sp] \leftarrow T$
• $Sp \leftarrow Sp - 1$
- g) • $T \leftarrow \text{Stack}[Sp]$
• $Sp \leftarrow Sp + 1$
- h) • $T \leftarrow \text{Stack}[Sp]$
• $Sp \leftarrow Sp - 1$

Phần lời giải

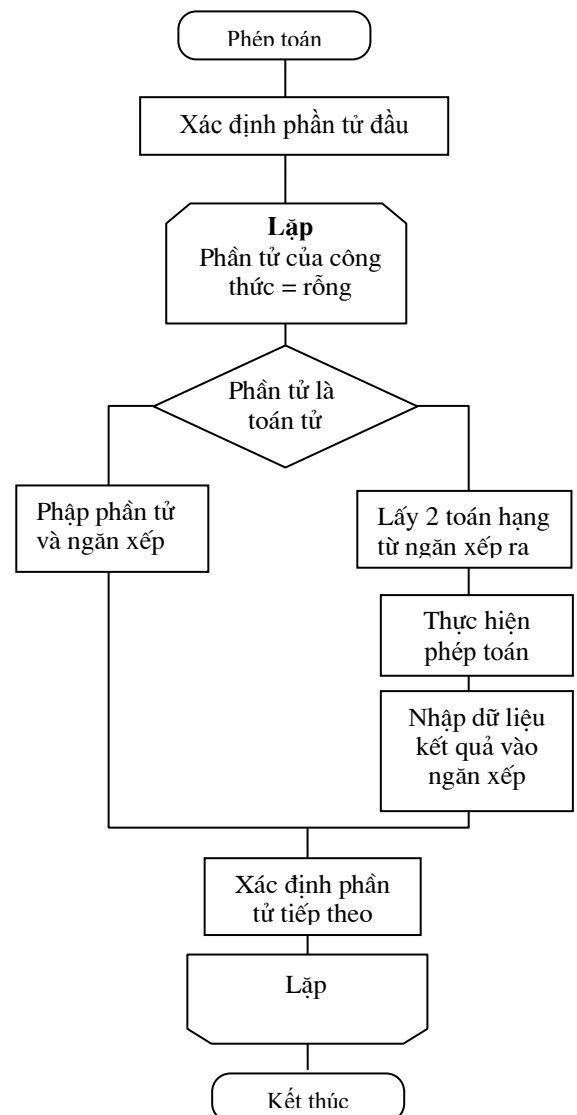
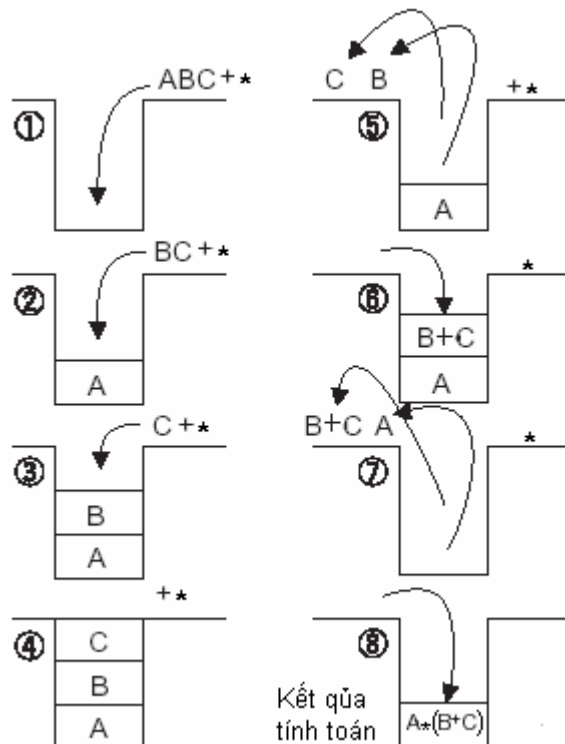
Câu hỏi con	Đáp án đúng
a	c) numF = true
b	e) $T \leftarrow 0$
c	b) negF \leftarrow true
d	e) $\text{Stack}[\text{Sp}] \leftarrow T$; $\text{Sp} \leftarrow \text{Sp} + 1$
e	d) $\text{Sp} \leftarrow \text{Sp} - 1$; $T \leftarrow \text{Stack}[\text{SP}]$

Ví dụ về các phép toán dùng kí pháp Balan đảo ngược

[Thuật toán]

- (1) Đẩy nó vào ngăn xếp, nếu là một toán hạng
- (2) Nếu là một phép toán, thực hiện phép toán xác định cho hai toán hạng được đặt ở vị trí cao nhất trong ngăn xếp, và trả lại kết quả vào ngăn xếp.

{ Thí dụ: $A * (B + C)$
 Trong kí pháp Balan ngược: $ABC + *$ }



Để không thay đổi trong phần câu hỏi, sao lại toàn bộ chương trình và chú thích vào ngay chương trình cho dễ hiểu:

[Chương trình]

O Subprogram name: Calc(Ex[], Lp, Ret)

O Character type: Ex[]

O Integer type: Lp, Cp

O Real number type: Ret, X, Y, T

O logic type: NumF, NegF

• **Cp** \leftarrow **0** // Con trỏ ngăn xếp: nạp đầy ngăn xếp là **0**, **Lp** là giá trị đáy của stack

• **T** \leftarrow **0** // Khai báo giá trị ban đầu của số thực $T = 0$

• **NumF** \leftarrow **False** // Cờ báo khi là một số

• **NegF** \leftarrow **False** // Cờ báo khi là một số âm

■ **Cp** \leq **Lp** // Khi con trỏ ngăn xếp trở tới đáy ngăn xếp, thoát khỏi vòng lặp.

↑ **Ex[Cp]** = **Number** // Nếu là một số trong mảng biểu diễn kí pháp Balan ngược

• **T** \leftarrow **T * 10 + ToReal (Ex [Cp])** // Biến đổi một số đơn trong mảng thành số thực T
 {ToReal () là một hàm dùng để biến đổi một chữ số đơn (kí tự số đơn) thành kiểu số thực}

• **NumF** \leftarrow **true**

↑ a // If numF= true, Nếu là một số

↑ **NegF** = **True** // Nếu số đó là số âm, đổi thành số dương và xóa cờ NegF

• **T** \leftarrow **-T**

↓ • **NegF** \leftarrow **false**

• **Push (T)** // đưa vào ngăn xếp là số âm thì đảo dấu, là số dương đưa ngay vào stack!!

• b // Sau khi đưa vào ngăn xếp, xóa biến T (temp): $T \leftarrow 0$.

↓ • **NumF** \leftarrow **false** // Xóa cờ NumF báo là số, để cho vòng lặp tiếp.

↑ **Ex [Cp]** = **"+"** // Nếu là phép cộng

• **Pop (Y)**

• **Pop (X)**

• **T** \leftarrow **X + Y**

• **Push (T)**

• **T** \leftarrow **0**

↑ **Ex[Cp]** = **"-"** // Nếu là phép trừ

↑ **Cp** \neq **Lp** AND **Ex [Cp + 1]** = **Number** // chưa phải số cuối, tiếp theo là một số.

• c // **negF** \leftarrow **true**

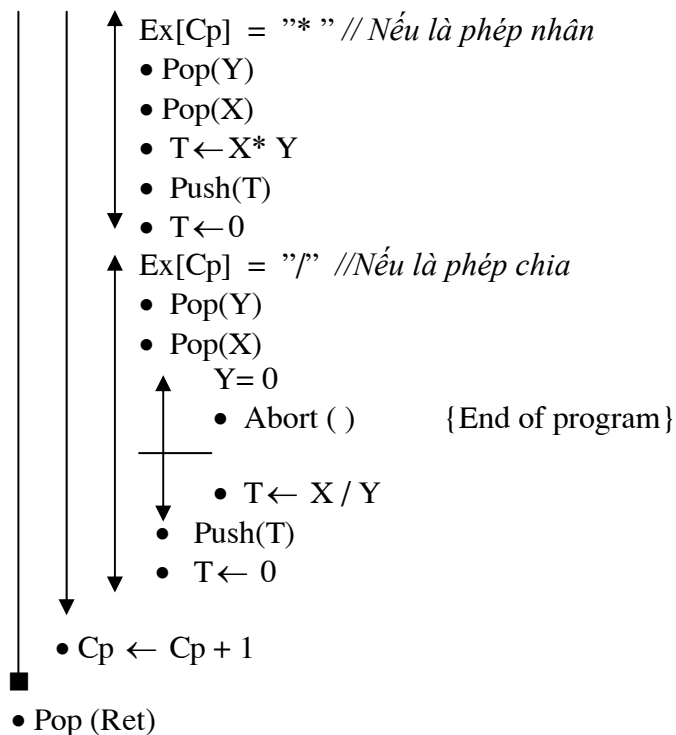
• **Pop[Y]**

• **Pop [X]**

• **T** \leftarrow **X - Y**

• **Push(T)**

• **T** \leftarrow **0**



Định nghĩa các biến như sau:

Ex[] : Mảng chứa các phần tử theo kí pháp Ba lan ngược

Cp: Con trỏ tới phần tử mảng Ex[]

Lp: Trỏ đến phần tử cuối mảng Ex[] (Xem hai ví dụ trong đề bài)

Ret, X, Y, T: các biến trung gian, số thực.

NumF: Cờ báo phần tử mảng là số đơn, biến logic

NegF: Cờ báo phần tử mảng là số âm, biến logic

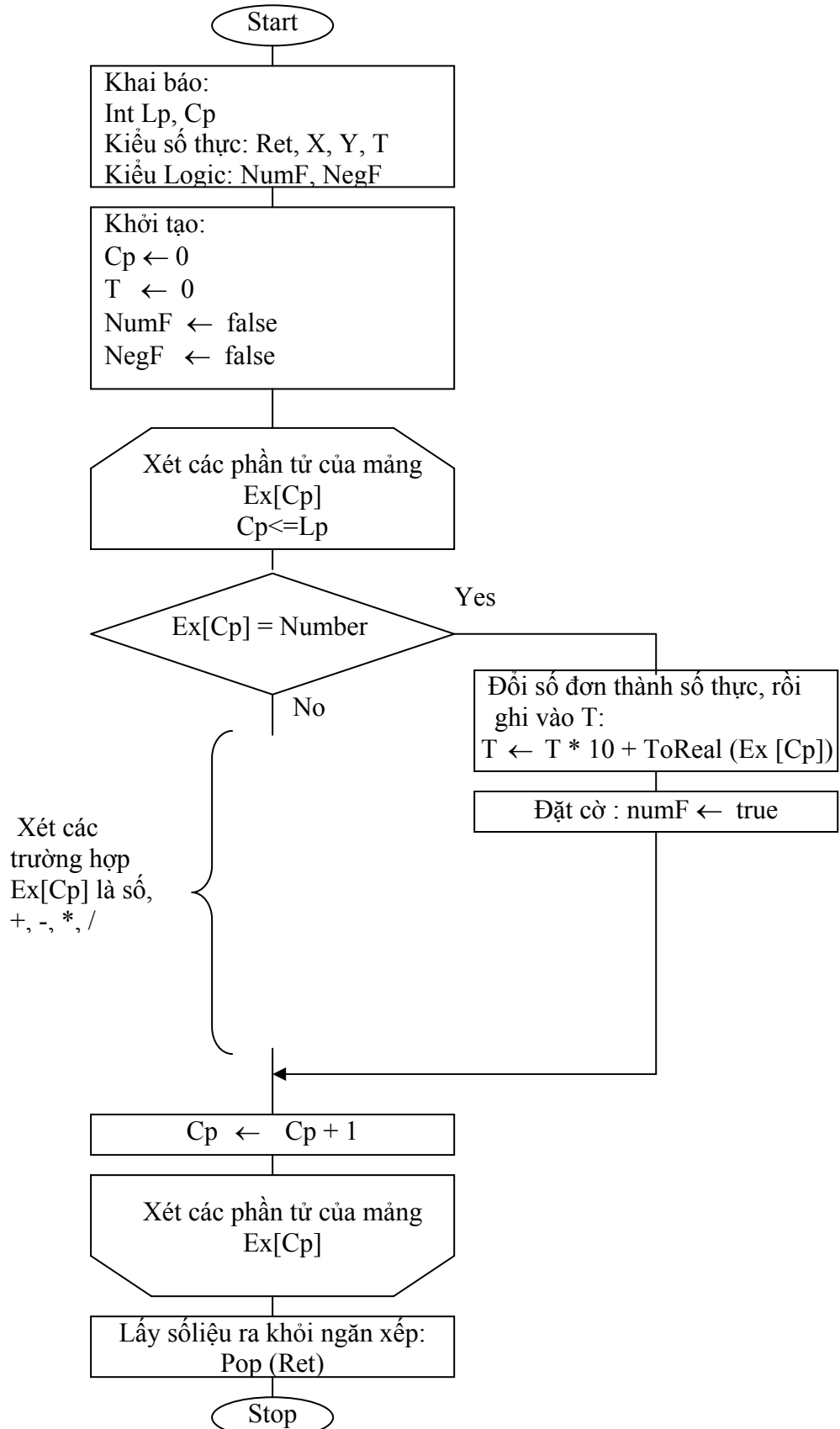
Phần Yes trong lưu đồ sau ứng với phần tử mảng Ex[] là số. Trong ví dụ của đề bài, số đầu tiên đó có thể là số 1 (trong ví dụ 1) hoặc 12 (trong ví dụ 2). Sau khi nhận hết một số (1 hoặc 12), tiếp theo gặp dấu cách (kí hiệu là Δ), chương trình thực hiện phần No của Lưu đồ. Do cờ numF được đặt true trong phần Yes, chương trình sẽ thực hiện phần chương trình trong ô trống [a] Nhiệm vụ của phần chương trình này, trong phần IF với điều kiện là ô [a], là đưa số thực T đó vào ngăn xếp, tùy theo số thực đó là dương hay âm.

- Nếu là số âm: đảo T, rồi đưa vào ngăn xếp
- Nếu là số dương, thì đưa vào ngăn xếp ngay.

Do vậy, ô trống [a] phải là: numF = true và ô trống [b] phải là $T \leftarrow 0$. Sau khi đưa số thực T vào ngăn xếp, phải xóa số thực về 0, cũng như cờ negF được xóa về false sau khi xử lí số thực là số âm (đảo số thực đó: $T \leftarrow -T$).

Lưu ý: Trong chương trình, các biến trung gian, sau khi được gán giá trị tại một bước trong vòng lặp, sau khi đã sử dụng giá trị được gán của biến đó, trước khi thực hiện bước tiếp theo, các biến đó bao giờ cũng được xóa về giá trị khởi đầu!

[Lưu đồ của Chương trình] có thể được vẽ như sau:

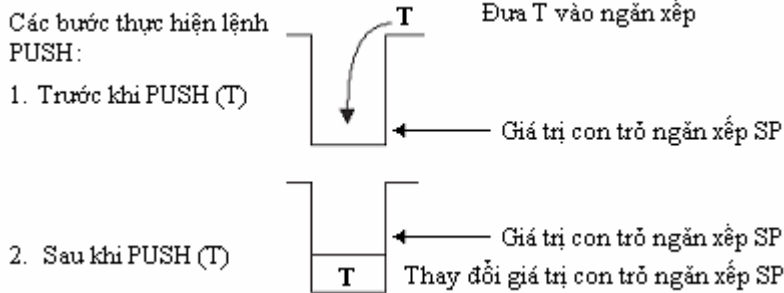


Cơ chế của LIFO được thể hiện trong hai chương trình con PUSH (T) và POP (T):

Ngăn xếp là một vùng bộ nhớ được dùng để lưu trữ các kết quả trung gian. Phép toán cất vào ngăn xếp là PUSH (T), còn phép toán lấy ra là POP (T), trong đó T là giá trị cần lưu giữ.

Con trỏ ngăn xếp SP luôn được khởi tạo là đáy của ngăn xếp (Địa chỉ cao nhất của vùng bộ nhớ)

/ Chương trình con PUSH (T) đưa số T vào ngăn xếp */*



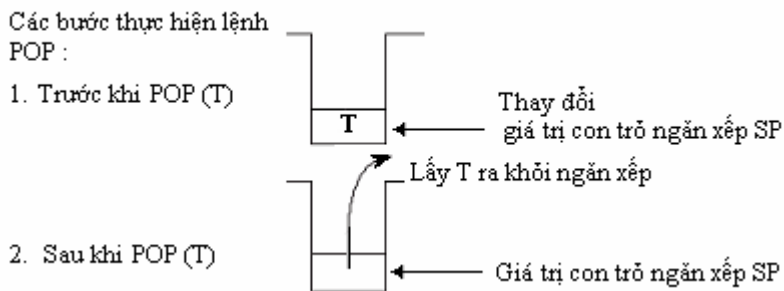
Ở đây cần lưu ý một điều là trong giả ngữ lệnh PUSH (T) được mô tả như trong đề bài, nghĩa là:

- Bước 1: Đặt giá trị T vào ngăn xếp: $Stack[Sp] \leftarrow T$
- Bước 2: Thay đổi giá trị con trỏ ngăn xếp: $Sp \leftarrow Sp + 1$

Nhưng trong thực tế, giá trị SP sau lệnh PUSH (T) bị trừ đi số byte mà là độ dài của T. Nếu T là một từ hai byte, thì $Sp \leftarrow Sp - 2$.

Tương tự, trong thực tế giá trị con trỏ ngăn xếp được cộng thêm vào sau khi bắt đầu thực hiện lệnh POP (T).

/ Chương trình con POP (T) lấy một số từ ngăn xếp vào T */*



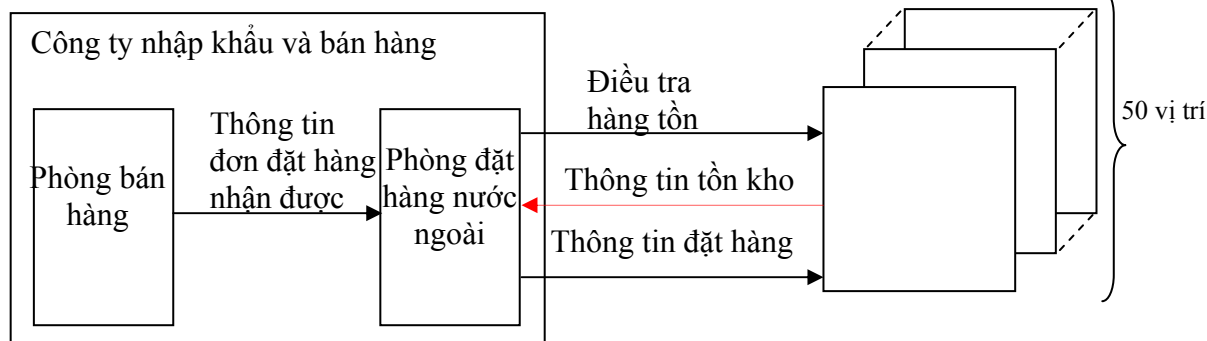
Trong giả ngữ lệnh POP (T) thực hiện theo 2 bước sau:

- Bước 1: Thay đổi giá trị SP để trỏ tới giá trị cần lấy ra: $Sp \leftarrow Sp - 1$
- Bước 2: Lấy giá trị T ra khỏi ngăn xếp: $T \leftarrow Stack[SP]$

Q5. Hãy đọc văn bản sau về thiết kế chương trình, sau đó trả lời câu hỏi con 1 và 2

Một hệ thống phân công nhập hàng đang được thiết kế cho phòng đặt hàng nước ngoài của một công ty nhập khẩu và bán hàng. Khái niệm về hệ thống như sau.

[Mô tả chung về hệ thống]



(1) Mã hàng được đặt, số lượng đặt, và ngày giao hàng mà phòng bán hàng mong muốn, được nhận trực tuyến từ phòng bán hàng. Thông tin đơn hàng có định dạng như sau:

Mã hàng	Số lượng yêu cầu	Ngày giao hàng mong muốn		
		Năm	Tháng	Ngày

(2) Dựa theo mã hàng trong thông tin đơn hàng, hệ thống phân công nhập hàng của phòng đặt hàng nước ngoài gửi mã hàng tới các điểm nhập hàng có mặt hàng đó và yêu cầu cho biết số lượng hàng còn trong kho. Có 50 điểm nhập khẩu trên toàn thế giới, và chúng kết nối trực tiếp với phòng đặt hàng nước ngoài. Các sản phẩm được tích lũy tùy thuộc vào từng điểm nhập khẩu cụ thể. Trong một số trường hợp, nhiều điểm nhập khẩu tích trữ cùng một loại hàng.

(3) Để phản hồi, các điểm nhập khẩu gửi ngay về phòng đặt hàng nước ngoài các thông tin tồn kho về mặt hàng yêu cầu. Nếu không có hàng tồn kho, thì câu trả lời được gửi lại với số lượng hàng trong kho bằng 0. Phòng đặt hàng nước ngoài ghi lại các thông tin tồn kho nhận được vào một tệp thông tin tồn kho. Thông tin tồn kho có định dạng sau.

Mã điểm nhập khẩu	Mã hàng	Số lượng trong kho
-------------------	---------	--------------------

(4) Có hai cách để giao hàng từ điểm nhập khẩu tới phòng bán hàng: bằng tàu thủy và bằng máy bay. Số ngày cần để giao hàng cho phòng bán hàng (được gọi là “số ngày để giao hàng”) từ một điểm nhập khẩu đã cho thay đổi phụ thuộc vào việc vận chuyển bằng đường biển hay đường hàng không. Số ngày để giao hàng cũng thay đổi phụ thuộc vào điểm nhập khẩu cụ thể.

Số ngày để giao hàng đối với mỗi điểm nhập khẩu được lưu trong tệp chủ về các điểm nhập khẩu. Tệp này có định dạng sau.

Mã điểm nhập khẩu	Tên điểm nhập khẩu	Số ngày để giao hàng bằng tàu thủy	Số ngày để giao hàng bằng máy bay	Thông tin khác
-------------------	--------------------	------------------------------------	-----------------------------------	----------------

- (5) Quyết định về việc có thể nhập hàng hay không được đưa ra và số lượng hàng nhập từ mỗi điểm được nhập được xác định trên cơ sở thông tin tồn kho từ các điểm nhập khẩu và số ngày để giao hàng bằng tàu thủy hay bằng máy bay từ các điểm đó. Cụ thể, trước tiên phải kiểm tra để xác định những điểm nào có hàng yêu cầu theo kiểm kê, và có thể đáp ứng được ngày giao hàng mong muốn. Tiếp theo, sẽ không đáp ứng được số lượng hàng cần nhập thì sẽ phân công theo nhiều điểm. Thủ tục xử lý để làm việc đó được phân công theo các mức ưu tiên từ ① đến ③.

① Ưu tiên cao hơn cho các điểm nhập khẩu có thể đáp ứng được thời hạn giao hàng mong muốn qua đường tàu thủy so với các điểm đáp ứng được thời hạn giao hàng qua đường hàng không.

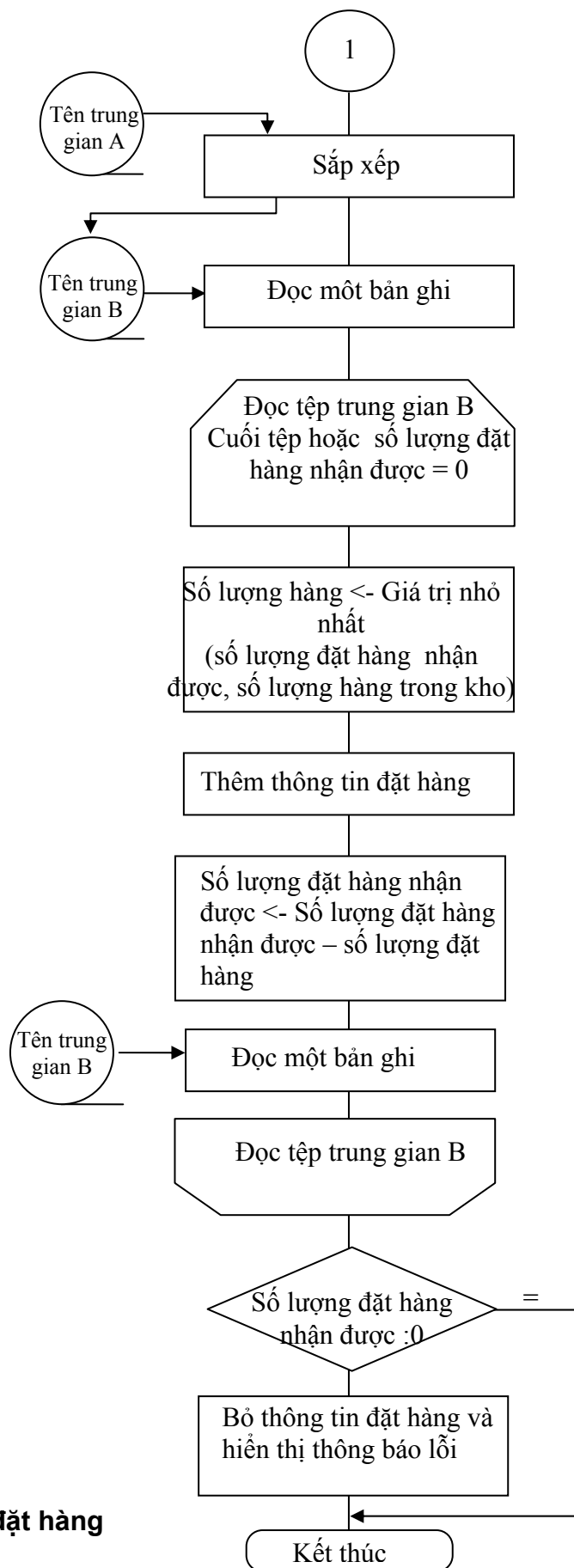
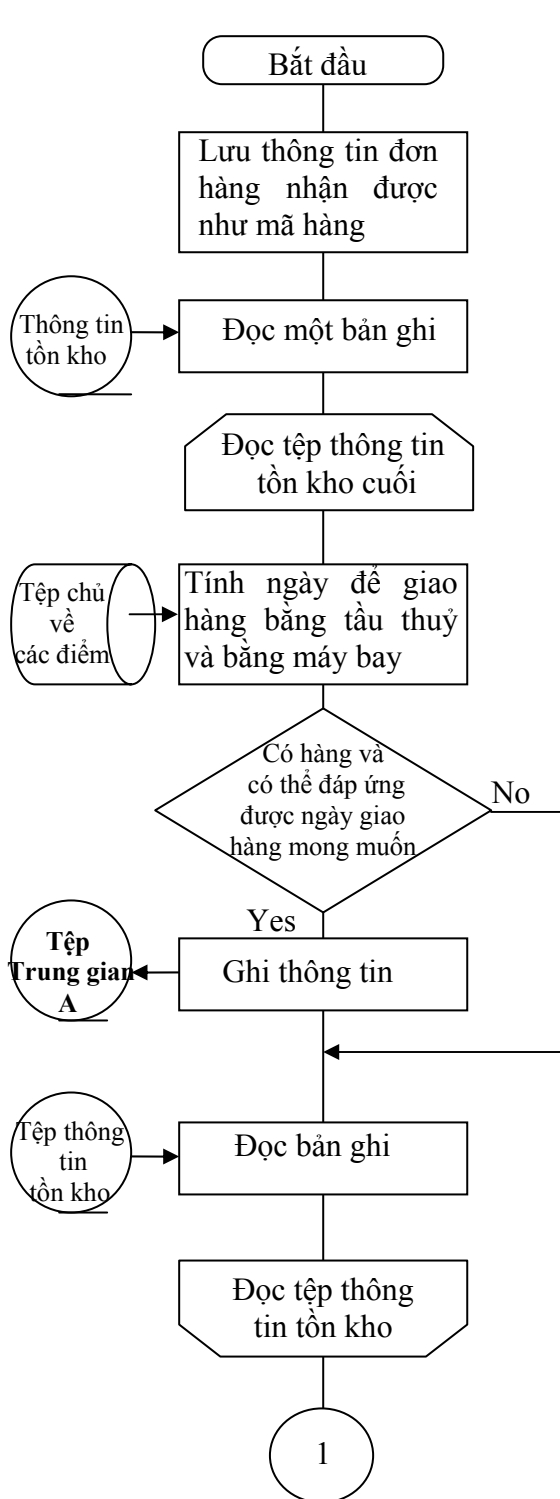
② Ưu tiên cho các điểm nhập khẩu có số lượng mặt hàng yêu cầu lớn

③ Ưu tiên cho các điểm nhập khẩu có mã điểm nhập khẩu nhỏ hơn.

Nếu không thể chấp nhận được số hàng yêu cầu vào ngày giao hàng mong muốn, một thông báo về việc đó sẽ được đưa ra.

- (6) Một khi các điểm nhập khẩu, số lượng hàng được đặt tương ứng tại các điểm đó, và phương tiện giao hàng được xác định, thông tin đặt hàng được gửi tới các điểm nhập khẩu liên quan. Thông tin đặt hàng có định dạng như sau. Trong định dạng, phương tiện giao hàng được lưu là một trong hai phương tiện tàu thủy hoặc máy bay.

Mã điểm nhập khẩu	Mã hàng	Số lượng đặt hàng	Phương tiện giao hàng
-------------------	---------	-------------------	-----------------------



Hình Thủ tục phân công các điểm đặt hàng

Câu hỏi con 1

Từ nhóm câu trả lời dưới đây, hãy chọn ba câu trả lời đúng cho các mục dữ liệu cần phải đưa vào tệp trung gian A được chỉ ra trong biểu đồ luồng ở trên.

Nhóm câu trả lời:

- a) Ngày giao hàng mong muốn
- b) Mã điểm nhập khẩu
- c) Số lượng hàng trong kho
- d) Mã hàng
- e) Phương tiện giao hàng
- f) Số ngày để giao hàng
- g) Số lượng đặt hàng

Câu hỏi con 2

Hệ thống này cần được kiểm thử. Trước tiên, hãy xem các dữ liệu kiểm thử cho thông tin đặt hàng nhận được.

Mã hàng	Số lượng đặt	Ngày giao hàng mong muốn		
		Năm	Tháng	Ngày
12345	200	2003	05	10

Giả thiết rằng, việc so sánh ngày giao hàng mong muốn với ngày nhận hàng cho thấy có 20 ngày dự trữ cho số ngày để giao hàng. Như được chỉ ra trong bảng sau, có 7 mẫu khác nhau đã được thiết lập làm dữ liệu kiểm thử cho số lượng hàng cho kho và số ngày giao hàng tại các điểm nhập khẩu có mặt hàng đó. Từ nhóm câu trả lời sau, hãy chọn các câu trả lời đúng để điền vào các ô trống từ đến trong bảng sao cho thỏa mãn các kết quả mong đợi chỉ ra ở trong đó.

Bảng chi tiết về các mẫu kiểm thử và các kết quả mong đợi

Mã thử	Mã điểm	Lượng hàng trong kho	Số ngày để giao hàng bằng máy bay	Số ngày để giao hàng bằng tàu thủy	Các kết quả mong đợi
1	01	500	10	20	Đặt 200 đơn vị hàng bằng tàu thủy tại điểm nhập khẩu có mã là 01
2	02	300	8	21	Đặt 200 đơn vị hàng bằng máy bay tại điểm nhập khẩu có mã là 02
3	03	500	21	21	Thông báo lỗi được thực thi.

4	04	100	6	19	Thông báo lỗi được thực thi.
	05	50	7	21	
5	06	70	7	20	Đặt 70 đơn vị hàng bằng tàu thủy tại điểm nhập khẩu có mã là 06; 70 đơn vị hàng bằng máy bay tại điểm nhập khẩu có mã là 07; và 60 đơn vị hàng bằng máy bay tại điểm nhập khẩu có mã là 08.
	07	70	12	30	
	08	a	8	21	
6	09	150	2	22	Đặt 150 đơn vị hàng bằng máy bay tại điểm nhập khẩu có mã là 09; và 50 đơn vị hàng bằng máy bay tại điểm nhập khẩu có mã là 10
	10	b	3	22	
	11	120	4	23	
7	12	c	20	40	Đặt 30 đơn vị hàng bằng máy bay tại điểm nhập khẩu có mã là 12; 160 đơn vị hàng bằng tàu thủy tại điểm nhập khẩu có mã là 13; và 10 đơn vị hàng bằng máy bay tại điểm nhập khẩu có mã là 14.
	13	160	10	20	
	14	30	15	25	

Nhóm câu trả lời:

- a) 20
- b) 30
- c) 50
- d) 70
- e) 100
- f) 120
- g) 160

Phần lời giải

Câu hỏi con	Đáp án đúng
1	b) Mã điểm nhập khẩu
	c) Số lượng hàng trong kho
	e) Phương tiện giao hàng
2A	d) 70
2B	f) 120
2C	b) 30

Câu hỏi con 1:

Cho đến Phần này của lưu đồ, chương trình mới trong chu trình “Đọc tệp thông tin tồn kho” mà đầu vào gồm:

Hình 1: Tệp thông tin tồn kho, gửi tới từ 50 điểm:

Mã điểm nhập khẩu	Mã hàng	Số lượng trong kho
-------------------	---------	--------------------

Tệp chủ về các điểm được dùng để phân tích xem khả năng đáp ứng của từng điểm

Mã điểm nhập khẩu	Tên điểm nhập khẩu	Số ngày để giao hàng bằng tàu thủy	Số ngày để giao hàng bằng máy bay	Thông tin khác
-------------------	--------------------	------------------------------------	-----------------------------------	----------------

Trong Tệp A cần lưu các mục dữ liệu sau:

b) Mã điểm nhập khẩu

c) Số lượng hàng trong kho

e) Phương tiện giao hàng (bằng tàu thủy hoặc máy bay)

Còn các mục dữ liệu khác:

a) Ngày giao hàng mong muốn: Theo lưu đồ điểm giao hàng đã đáp ứng được ngày giao hàng mong muốn. là thông tin của Bảng thông tin đơn hàng, được gửi tới 50 điểm từ Phòng đặt hàng nước ngoài của Công ty nhập khẩu và bán hàng, cho nên không cần thông tin này đưa vào tệp A.

d) Mã hàng: là thông tin của Bảng thông tin đơn hàng, được gửi tới 50 điểm từ Phòng đặt hàng nước ngoài của Công ty nhập khẩu và bán hàng. Mã hàng đã có sẵn trong tệp thông tin tồn kho(Hình 1 bên trên). Ở đây tệp đó được đọc vào và thay đổi để ghi vào tệp A (Xem lưu đồ).

f) Số ngày để giao hàng: Không có thông tin này vì có hai trường cụ thể hơn là Số ngày để giao hàng bằng tàu thủy hoặc Số ngày để giao hàng bằng máy bay.

g) Số lượng đặt hàng: chưa cần đến, vì chưa phân tích xong khả năng của từng điểm.

Câu hỏi con 2

Trong tất cả các trường hợp kết quả mong đợi là đặt **200** đơn vị hàng theo ưu tiên mô tả ở trên.

1. Theo kết quả mong đợi A = 70. Đáp án đúng là d).

Theo kết quả mong đợi, chỉ còn thiếu 60 đơn vị hàng với mã là 08 là đủ cho mẫu kiểm thử này. Với mẫu này số ngày nhập bằng tàu thủy chỉ mất có 21 ngày, nên ưu tiên hơn. Do vậy lượng hàng trong kho với mã điểm “08” ít nhất phải là 70 đơn vị (Theo ưu tiên (1)).

2. Theo kết quả mong đợi B = 120. Đáp án đúng là f)

Chỉ thiếu có 50 đơn vị hàng nữa nếu đặt 150 đơn vị hàng với mã là "09". Song theo mong đợi tất cả các đơn vị hàng đều nhập khẩu bằng máy bay, cho nên để đơn vị hàng có mã là "10" có ưu tiên cao hơn, thì lượng hàng trong kho với mã điểm "10" ít nhất phải là 120 (ngang với lượng hàng trong kho với mã điểm bằng "11")

3. Theo kết quả mong đợi C = 30. Đáp án đúng là b).

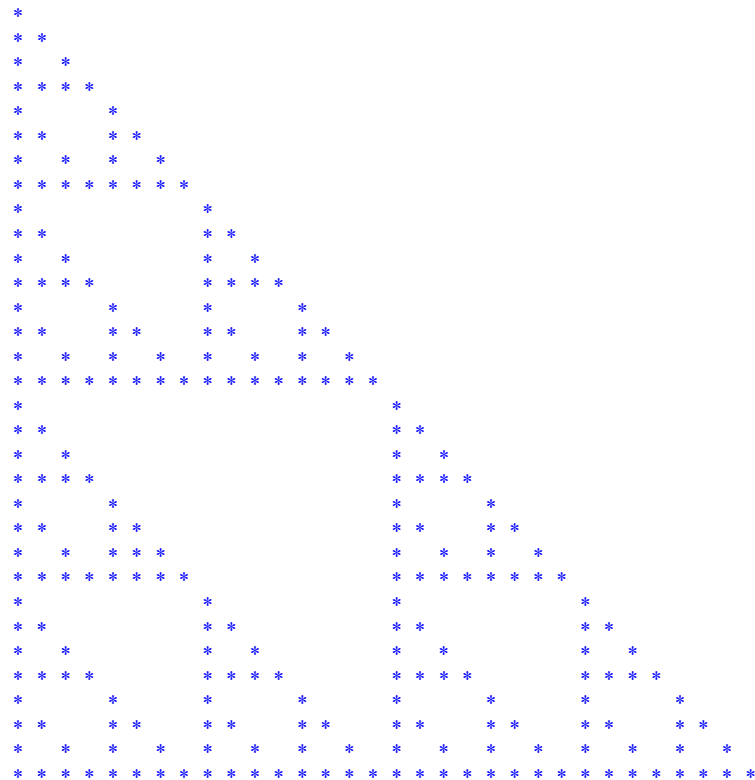
Theo kết quả mong đợi, lượng hàng trong kho với mã điểm "12" ít nhất phải là 30 đơn vị cho mẫu kiểm thử này. Theo ưu tiên thứ (2).

Q6. Hãy đọc mô tả sau về một chương trình C và đọc chương trình đó, sau đó trả lời các câu hỏi con 1 và 2.

[Mô tả chương trình]

(1) Chương trình này tạo ra đồ thị có tính cân đối bằng cách sử dụng một mảng hai chiều.

Chương trình cho ra đồ thị fractal (được gọi là Sierpinski gasket) được chỉ ra trong hình dưới đây từ các giá trị ban đầu trong dòng 0 của bảng dưới đây. Trong ví dụ này, mảng có 33 dòng và cột.



Hình: Kết quả đưa ra

Bảng Các giá trị của trạng thái được tạo ra

		Cột												
		0	1	2	3	4	5	6	7	8	9	...	32	
Các giá trị ban đầu	0	0	1	0	0	0	0	0	0	0	0	...	0	
	1	0	1	1	0	0	0	0	0	0	0	...	0	
	2	0	1	0	1	0	0	0	0	0	0	...	0	
	3	0	1	1	1	1	0	0	0	0	0	...	0	
	4	0	1	0	0	0	1	0	0	0	0	...	0	
Các giá trị trạng thái được tạo ra	5	0	1	1	0	0	1	1	0	0	0	...	0	
	6	0	1	0	1	0	1	0	1	0	0	...	0	
	7	0	1	1	1	1	1	1	1	1	0	...	0	
	8	0	1	0	0	0	0	0	0	0	1	...	0	
	9	0	1	1	0	0	0	0	0	0	1	...	0	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
	31	0	1	1	1	1	1	1	1	1	1	...	1	
	32	0	1	0	0	0	0	0	0	0	0	...	0	

(2) Các giá trị trạng thái (0 hoặc 1) được lưu trong mảng hai chiều s . Các giá trị trạng thái trong dòng 1 và các dòng sau đó được tạo ra theo các quy tắc dưới đây từ các giá trị ban đầu được lưu trong dòng 0, như ta thấy ở bảng trên.

Giá trị trạng thái của dòng i cột j được biểu diễn bằng $s[i][j]$.

Mọi giá trị trong cột 0 đều bằng 0.

Các giá trị $s[i+1][j]$ (trong đó $j \geq 1$) được tạo ra theo bảng sau, dựa trên các giá trị ban đầu của $s[i][j-1]$ and $s[i][j]$.

Các giá trị trạng thái dùng làm cơ sở		Giá trị trạng thái được tạo ra cho $s[i+1][j]$
$s[i][j-1]$	$s[i][j]$	
1	1	0
1	0	1
0	1	1
0	0	0

(3) Các giá trị của cột 0, dòng 32 trong mảng hai chiều s không phải là đầu ra.

[Program]

```
# include <stdio.h>
#define ALIVE 1
#define DEAD 0
#define Sz 33

int stschk (int, int );
main()
{
    int s[Sz] [Sz], i, j;
    for (i=0; i<Sz; i++) s[i][0] = DEAD;
    for (j=0; j<Sz; j++) s[0][j] = DEAD;
    s[0][1] = ALIVE;
    for ( i=0; i<Sz-1; i++) {
        for ( j=0; j<Sz; j++) {
             = stschk (  ,  );
            if (  == ALIVE) printf( "." );
            else printf( " " );
        }
        printf ( "\n" );
    }

    int stschk( int s1, int s2 )
    {
        if (((s1 == DEAD ) && (s2 == ALIVE )) ||
            ((s1 == ALIVE ) && (s2 == DEAD))) return ALIVE;
        else return DEAD;
    }
}
```

Câu hỏi con 1

Từ nhóm câu trả lời dưới đây, hãy chọn các câu trả lời đúng để chèn vào ô trống từ

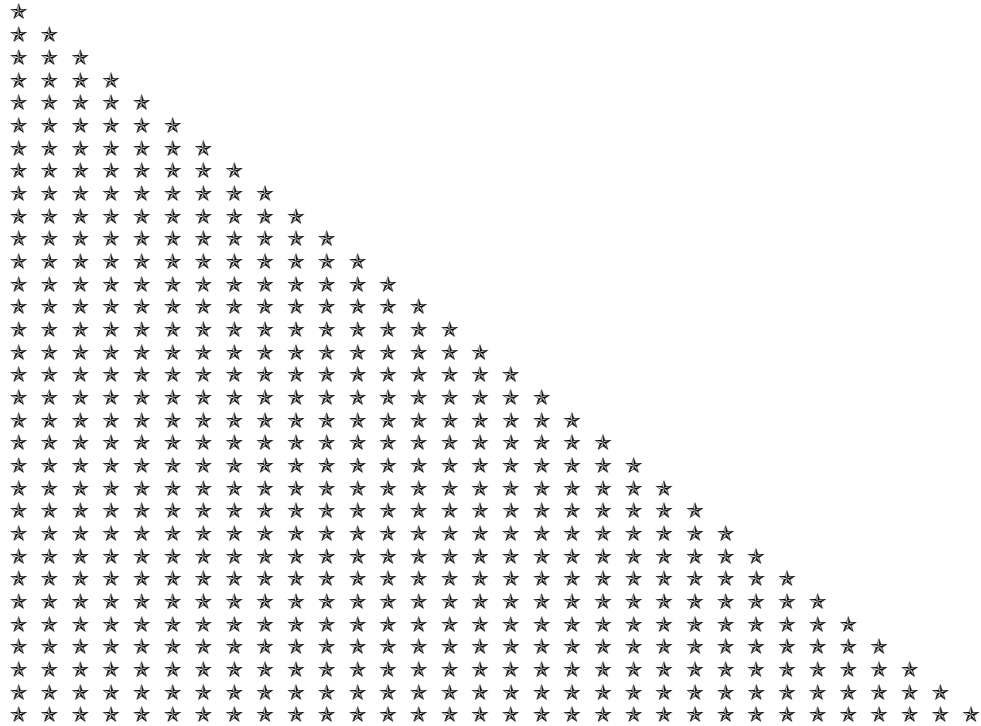
đến trong chương trình trên.

Nhóm câu trả lời:

- | | | |
|----------------|--------------|----------------|
| a) s[i-1][j-1] | b) s[i-1][j] | c) s[i-1][j+1] |
| d) s[i][j-1] | e) s[i][j] | f) s[i][j+1] |
| g) s[i+1][j-1] | h) s[s+1][j] | i) s[i+1][j+1] |

Câu hỏi con 2

Đồ thị mà ta thấy dưới đây sẽ được đưa ra nếu mệnh đề if trong hàm stschk bị thay đổi. Từ nhóm câu trả lời dưới đây, hãy chọn mệnh đề if dẫn đến kết quả này.



Nhóm câu trả lời:

- a) if ((s1 == ALIVE) || (s2 == ALIVE)) return ALIVE;
else return DEAD;
- b) if ((s1 == ALIVE) && (s2 == ALIVE)) return ALIVE;
else return DEAD;
- c) if ((s1 == DEAD) || (s2 == DEAD)) return DEAD;
else return ALIVE;
- d) if ((s1 == DEAD) && (s2 == ALIVE)) return ALIVE ;
else return DEAD;

Phần lời giải

Câu hỏi con 1:

Từ bảng trạng thái đã cho :

Các giá trị trạng thái dùng làm cơ sở		Giá trị trạng thái được tạo ra cho
$i=0..SZ-1, j=1..SZ-1$		
$s[i][j-1]$	$s[i][j]$	$s[i+1][j]$
1	1	0
1	0	1
0	1	1
0	0	0

Ta thấy rằng: Giá trị trạng thái tạo ra cho một ô mới $s[i+1][j]$ chính bằng phép XOR bit của hai ô đầu vào tương ứng là $s[i][j-1]$ và $s[i][j]$. Hàm ***int stschk(int s1, int s2)*** thực hiện phép toán này, hiển nhiên là $a = s[i+1][j]$, ta cần xác định, b và c là ô nào trong các ô: $s[i][j-1]$ và $s[i][j]$. Nhưng vì sau lệnh tính a có lệnh:

```
if (b==ALIVE) printf("*")
else printf("");
```

nên ta chỉ cần xét các cột biên ứng với $j=1$ và $j=SZ-1$ trong vòng lặp for của chương trình để xác định b (Hiển nhiên khi các cột biên đúng thì các cột còn lại phải đúng). Giả sử $b = s[i][j-1]$ thì với $j=1$, các ô trong cột $j-1$ đều bằng 0 (DEAD) (theo yêu cầu cho trước), câu lệnh print ở trên chỉ in ra các khoảng trống của cột 0, điều này trở nên vô nghĩa, trong khi với $j=SZ-1$ (cột cuối cùng) thì các dấu "*" trong cột $SZ-1$ không được đưa vào đồ thị. Vì vậy $b = s[i][j-1]$ là không đúng, Do đó $b = s[i][j]$. Thật vậy, vì $b = s[i][j]$, nên khi $j=1$ hoặc khi $j=SZ-1$ thì các dấu "*" trong các cột này đều được đưa vào đồ thị. Do đó kết quả cuối cùng là:

Câu hỏi con 1	Đáp án đúng
a	h) $s[i+1][j]$
b	e) $s[i][j]$
c	d) $s[i][j-1]$

Câu hỏi con 2

Từ đồ thị ta thấy rằng, chỉ cần một trong hai ô $b=s[i][j]=1$ hoặc $c=s[i][j-1]=1$ thì ô $a=s[i+1][j]=1$. Tất cả các ô có giá trị 1 đều được thay bằng dấu `"*"`. Như vậy, tương ứng với giá trị của các tham biến `s1` và `s2` trong hàm `stschk` sẽ phải thỏa mãn điều kiện của mệnh đề `if` như sau:

```
if ((s1 == ALIVE) || (s2 == ALIVE)) return ALIVE;

else return DEAD;
```

Do vậy, đáp án đúng là a)

Q10. Đọc mô tả sau về chương trình C và đọc chính chương trình, sau đó trả lời câu hỏi con

[Mô tả chương trình]

Một chương trình sẽ được tạo ra để xác định hệ số hiệu chỉnh nhiệt độ dùng để hiệu chỉnh nhiệt độ của một bộ cảm biến áp điện (tức thiết bị tạo ra điện áp phù hợp với cường độ áp suất sử dụng). Đặc tính đầu ra của bộ cảm ứng điện này biến đổi theo nhiệt độ xung quanh, nên giá trị đầu ra phải được hiệu chỉnh. Các kết quả cho thấy bảng dưới đây thu được thông qua việc nghiên cứu thực nghiệm về thay đổi trong giá trị đầu ra của bộ cảm biến áp điện liên quan đến nhiệt độ xung quanh. Bảng này cho thấy tỉ lệ, với các giá trị đầu ra ở 0°C bằng 1.00 (ở đây dấu chấm được hiểu là dấu phẩy thập phân).

Bảng Nhiệt độ môi trường và tỉ lệ giá trị đầu ra của bộ cảm biến

Nhiệt độ	Tỉ lệ giá trị đầu ra của bộ cảm biến
-40°C	0.20
-20°C	0.60
-10°C	0.80
0°C	1.00
10°C	1.17
30°C	1.50
50°C	1.80

Để hiệu chỉnh giá trị đầu ra của bộ cảm biến áp điện áp (tinh thể piezo) thành giá trị không phụ thuộc vào nhiệt độ xung quanh trong quá trình đo, hệ số hiệu chỉnh nhiệt độ K được xác định dựa trên dữ liệu trong bảng. Nhiệt độ được hiệu chỉnh bằng cách nhân K với giá trị đầu ra của bộ cảm biến.

(1) Bảng hiệu chỉnh nhiệt độ được tạo ra từ các dữ liệu trong bảng trên. Bảng điều chỉnh nhiệt độ được thiết lập với cấu trúc mảng.

```
typedef struct {  
    int temp;           /* Nhiệt độ */  
    double Ratio;       /* Tỉ lệ giá trị đầu ra của bộ cảm biến (giá trị đo thực tế) */  
    double Step;        /* Tăng từng 1°C */  
}CURVE;
```

(2) Hàm số **SetupCurve**, cho giá trị ban đầu vào bảng hiệu chỉnh nhiệt độ, và hàm **GetK**, xác định hệ số hiệu chỉnh nhiệt độ K, được tạo ra.

- (3) Để xác định hệ số hiệu chỉnh nhiệt độ K cho nhiệt độ **Degree**, **GetK** tìm bảng hiệu chỉnh bằng cách sử dụng phương pháp tìm nhị phân. Nhiệt độ xung quanh được giả thiết là không nhỏ hơn -40°C và không lớn hơn 50°C . Nếu không có giá trị nhiệt độ tương ứng trong bảng hiệu chỉnh nhiệt độ, thì tỉ lệ giá trị đầu ra ứng với nhiệt độ Degree được xác định bằng nội suy tuyến tính, và đảo ngược của giá trị này được trả về là K.
- (4) Chương trình chính là một chương trình kiểm thử, chương trình này xác định và hiển thị hệ số điều chỉnh nhiệt độ K từ -40°C đến 50°C được gia thêm mỗi lần 1°C , để kiểm chứng hoạt động của hai hàm số này. Dưới đây là ví dụ của việc hiển thị này.

Temperature (Nhiệt độ)	Temperature correction coefficient (Hệ số hiệu chỉnh nhiệt độ)
-40	5.00
-39	4.55
-38	4.17
-37	3.85
-36	3.57
-35	3.33
-34	3.13
-33	2.94
-32	2.78
-31	2.63
-30	2.50
-29	2.38
-28	2.27
-27	2.17
-26	2.08
-25	2.00

[Chương trình]

```
#include<stdio.h>
typedef struct{
    int Temp;      /* Nhiệt độ */
    double Ratio;  /* Tỷ số giá trị đầu ra của bộ cảm biến (giá trị đo thực tế) */
    double Step;   /* Mức tăng trên 1°C */
} CURVE;

void SetupCuver(CURVE *, int) ;
double GetK(int, CURVE *, int) ;
#define ITEMS 7

main()
{
    int Degree ;
    double k ;
    CURVE cuver[ITEMS] = {
        { -40 , 0.20 , 0.0 },
        { -20 , 0.60 , 0.0 },
        { -10 , 0.80 , 0.0 },
        { 0 , 1.00 , 0.0 },
        { 50 , 1.80 , 0.0 } };

    SetupCurve(curve , ITEMS ) ;
    Printf("Temperture Temperture correction coefficient \n");
    for (Degree=-40; Degree<=50; Degree++) {
        k = GetK(Degree, curve, ITEMS);
        printf("%3d %4.2f \n",Degree, k);
    }
}

void SetupCurve(CURVE *p , int Points)
{
    /* Khởi động bảng hiệu chỉnh nhiệt độ */
    int i;
    for (i = 0 ; i < Point -1 ; i++, a) {
        p->Step = (b) / ((p + 1)->Temp - p->Temp);
    }
}

double GetK(int Temp , CURVE *p , int Points)
{
    /* Trả về hệ số hiệu chỉnh nhiệt độ K là giá trị của hàm */
    int i,j,n;
    i=0 ; j=Points -1 ;
    if ((Temp < p->Temp) || (Temp > (p+j)->Temp))
        Return 0.0; /* trả về giá trị 0.0 nếu ngoài khoảng nhiệt độ đo */

    /* Nếu trùng với nhiệt độ của phần tử cuối cùng trong bảng hiệu chỉnh */
}
```

```

    If (Temp == (p+j)->Temp)
    Return 1.0 / (p+j)->Ratio ;

/* Tìm bảng hiệu chỉnh nhiệt độ bằng phương pháp nhị phân */
    While (1) {
        n = ;
        If ((Temp >= (p+n)->Temp) &&
            (Temp < (p+n)->Temp)) break;
        If (Temp < (p+n)->Temp) j = n;
        Else  ;
    }
    p += n ;
    return 1.0 / (p->Ratio + p->Step * ());
}

```

!!!!Sai điều kiện IF: Temp < (p+n+1) -> Temp

Câu hỏi con

Từ các nhóm câu trả lời dưới đây, hãy chọn câu trả lời đúng để điền vào chỗ trống từ đến trong chương trình sau.

Nhóm câu trả lời cho a:

- | | | |
|-----------|-----------|--------------|
| a) p -- | b) p++ | c) p->Temp++ |
| d) p += 3 | e) p += 7 | |

Nhóm câu trả lời cho b:

- | | |
|--------------------------------|--------------------------------|
| a) (p-1)->Ration-p->Ration | b) (p-1)->Ration-(p+1)->Ration |
| c) p->Ration-(p-1)->Ration | d) p->Ration-(p->Ration - 1) |
| e) (p+1)->Ration - p -> Ration | |

Nhóm câu trả lời cho c:

- | | | |
|----------------|----------------|------------|
| a) (i + j)/2 | b) (i * j) / 2 | c) (i%j)/2 |
| d) (i + j) * 2 | e) (i % j) * 2 | |

Nhóm câu trả lời cho d:

- | | | |
|----------|--------------|--------------|
| a) i = 0 | b) i = n - 1 | c) i = n + 1 |
| d) j = n | e) j = n + 1 | |

Nhóm câu trả lời cho e:

- | | | |
|-----------------|-------------------|-------------------|
| a) Temp-p->Step | b) Temp - p->Step | c) Temp + p->Step |
| d) p->Step - n | e) p->Temp - Temp | |

Phần lời giải

Câu hỏi con:

Nhóm câu trả lời cho a: Đáp án đúng: b) $p++$

Mục đích của vòng lặp for nhằm khởi tạo bảng hiệu chỉnh nhiệt độ, tức là tính giá trị của trường Step theo trường Ratio trong từng khoảng nhiệt độ. Lúc đầu con trỏ p chỉ vào đầu bảng hiệu chỉnh nhiệt độ và cũng chính là hàng đầu tiên trong bảng, sau mỗi lần tính tr-êng Step xong th× con trỏ p phải dịch chuyển sang hàng tiếp theo, tức là tăng p lên 1 đơn vị ($p++$). Nhìn vào vòng lặp ta thấy chưa có câu lệnh tăng p, *do đó, đáp án cho a phải là b).*

Nhóm câu trả lời cho b: Đáp án đúng: e) $(p+1) \rightarrow \text{Ratio} - p \rightarrow \text{Ratio}$

Rõ ràng tại nhiệt độ $p \rightarrow \text{Temp}$, giá trị hệ số điều chỉnh nhiệt độ là $p \rightarrow \text{Ratio}$. Tại nhiệt độ $(p+1) \rightarrow \text{Temp}$, giá trị tương ứng của hệ số điều chỉnh nhiệt độ là $(p+1) \rightarrow \text{Ratio}$. Do đó giá trị bước tăng của hệ số hiệu chỉnh nhiệt độ trong khoảng này là:

$$p \rightarrow \text{Step} = ((p+1) \rightarrow \text{Ratio} - p \rightarrow \text{Ratio}) / ((p+1) \rightarrow \text{Temp} - p \rightarrow \text{Temp})$$

Hiển nhiên e) là đáp án đúng.

Nhóm câu trả lời cho c: Đáp án đúng: a) $(i+j)/2$

ở đây áp dụng thuật toán tìm kiếm nhị phân để tìm bảng hiệu chỉnh nhiệt độ. Rõ ràng, giá trị của n phải chia bảng ra làm 2 khoảng (trước và sau) để xác định nhiệt độ cần tính hệ số hiệu chỉnh rơi vào khoảng nào. Vì thế câu lệnh gán giá trị cho n phải là : $n = (i+j)/2$. *Điều này tương ứng với đáp án đúng là a).*

Nhóm câu trả lời cho d: Đáp án đúng: c) $i=n+1$

Theo câu lệnh if, điều kiện đúng của nó sẽ xác định nhiệt độ cần tính hệ số hiệu chỉnh nhiệt độ K rơi vào khoảng trước, do đó phải điều chỉnh cận trên để thu hẹp khoảng tìm kiếm nhị phân bằng cách đặt $j=n$. Trong trường hợp ngược lại, nhiệt độ để tính hệ số hiệu chỉnh rơi vào khoảng sau, do đó phải thay đổi cận dưới để thu hẹp khoảng tìm kiếm nhị phân bằng cách đặt $i=n+1$. *Rõ ràng đáp án đúng chỉ có thể là c).*

Nhóm câu trả lời cho e: Đáp án đúng: b) $\text{Temp} - p \rightarrow \text{Temp}$

Với điều kiện thoát khỏi vòng lặp `while` ở phía trên và lệnh `p+=n;` cho thấy con trỏ `p` định vị ở vị trí trong bảng mà tại đó nhiệt độ của bảng nhỏ hơn hoặc bằng nhiệt độ đưa vào (`Temp`) để tính hệ số hiệu chỉnh nhiệt độ `K`. Do vậy hệ số tỉ lệ tại nhiệt độ `p->Temp` sẽ được nội suy là:

$$\text{Ratio} = p \rightarrow \text{Ratio} + p \rightarrow \text{Step} * (\text{Temp} - p \rightarrow \text{Temp})$$

và hệ số điều chỉnh nhiệt độ `K` sẽ là `1/Ratio`.

Từ đó đáp án đúng cho e là phương án b)