제11회 전국상업경진대회

비즈니스프로그래밍 문제지

소속 학교		수험 번호		성명		시험 시간	150분

수험자 유의사항

▶ 경진참가자 유의사항 ◀

- 시험 전 필요한 소프트웨어(Visual Studio, JDK, Eclipse 등)를 확인한다.
- 답안을 작성하기 전 나누어 준 USB 메모리의 불량 여부를 확인한다.
- **문제별 실행파일과 소스파일을 저장**해야 하며 만약의 사태에 대비하여 수시로 저장한다.
- 시험이 종료되면 **컴퓨터 본체의 전원을 끄지 말고** 감독관의 지시에 따라 **모니터의 전원만 종료**한 뒤 **문제지와 답안 USB 메모리를 모두 제출**한다.
- **시험 시간이 종료되기 전이라도 답안 USB 메모리 제출이 가능**하며 **동점자는 제출 시간이 빠른 학생이 우선**한다. (단, 답안 USB 메모리 제출 이후 추가 작업 불가능 함)
- **저장되지 않았거나 저장파일이 손상되었을 경우에는 채점에서 제외**한다.

▶ 작업 시 공통사항 ◀

- USB 메모리 라벨지에 **수험번호를** 정확히 **기록**한다.
- 답안 파일은 USB 메모리에 저장하여 제출한다.
- 문제별로 폴더**(예, F:₩문제1)**를 만들어 저장하고, 파일명은 **"수험번호_문제번호"**로 하여, 실행파일(exe, class)과 소스파일을 해당 문제 폴더에 함께 저장한다.
  (예) C언어를 선택한 수험번호가 '12345'번인 학생의 '문제1'의 소스파일 경로 F:₩문제1₩12345_1.c
  (드라이브명은 컴퓨터에 따라 달라질 수 있음)
- **자바언어를 선택한 응시자는 소스코드에 패키지 명을 지정하지 않는다.**
  (package 키워드 사용금지, 위반 시 0점 처리됨.)
- **특별한 지시사항이나 처리조건이 없는 경우는 프로그램의 전반적인 로직과 사용 S/W의 특성에 맞게 작성하며 최대 처리 시간은 5초가 넘을 수 없다.**

NBO 전국상업경진대회 조직위원회

문제 1 [12점]

대한민국 국민이라면 누구나 납세의 의무를 이행해야 한다. 우리나라 국세청 누리집에 게시된 2020년 종합소득세율표는 아래와 같다.

과세표준	세율	누진공제
12,000,000원 이하	6%	-
12,000,000원 초과 46,000,000원 이하	15%	1,080,000원
46,000,000원 초과 88,000,000원 이하	24%	5,220,000원
88,000,000원 초과 150,000,000원 이하	35%	14,900,000원
150,000,000원 초과 300,000,000원 이하	38%	19,400,000원
300,000,000원 초과 500,000,000원 이하	40%	25,400,000원
500,000,000원 초과	42%	35,400,000원

소득이 많을수록 높은 세율을 적용하는 것이 기본 골자인데, 표를 살펴보면 소득이라는 단어 대신 과세표준이라는 용어가 사용된 것을 알 수 있다. 그 이유는 소득 전체에 세금이 부과되는 것이 아니라 국민연금, 건강보험 등 소득공제 대상에 해당하는 금액에는 세금이 붙지 않기 때문이다. 따라서 세금을 계산하려면 일단 소득액에서 소득공제액을 제외한 과세표준을 먼저 구하고 이 과세표준에 세율을 곱하여 세금을 계산하여야 한다. (단, 구간별로 세율을 나누어 계산해야 함. 예시 참고) 이렇게 계산된 소득세액에 마지막으로 월세세액공제, 자녀세액공제 등 세액감면액을 제하면 최종 소득세액이 확정된다. (단, 최종 소득세액은 0원보다 크거나 같아야 함)

예를 들어 소득액이 6000만원이고, 소득공제액이 1000만원, 그리고 세액감면액이 100만원인 경우를 계산하자면,

1) 과세표준 = 소득액(6000만원) - 소득공제액(1000만원) = 5000만원
2) 과세표준액 5000만원을 각 세율 구간으로 분리하여 세율 적용 (누진공제를 활용하면 더 쉽게 계산할 수 있지만 여기에서는 설명하지 아니함)

0원 초과 ~ 1200만원까지의 세금은 6% 세율 = 72만원
1200만원 초과 ~ 4600만원까지의 세금은 15% 세율 = 510만원
4600만원 초과 ~ 5000만원까지의 세금은 24% 세율 = 96만원
∴ 소득세액 = 72만원 + 510만원 + 96만원 = 678만원

3) 세액공제 적용

∴ 최종 소득세액 = 소득세(678만원) - 세액감면액(100만원) = 578만원

근로자의 소득액, 소득공제액, 세액감면액의 정보를 입력받아 최종 소득세액을 구하는 프로그램을 작성하시오.

입력

(1) 소득세 계산에 필요한 세 가지 정보가 한 줄에 입력된다.

(2) 소득액, 소득공제액, 세액감면액 이렇게 3개의 숫자가 공백으로 분리되어 입력된다. 숫자의 범위는 0부터 1,000,000,000사이의 숫자이다.

출력

(1) 최종 소득세액을 계산하여 출력한다.

(2) 최종 소득세액은 0보다 작을 수 없고, 10원 미만 원 단위 금액은 절사하여 정수로 출력한다. (예를 들어, 세금이 1998원으로 계산되면 원 단위 8원은 삭제하고 1990원만 부과한다)

형식

입력 예시	출력 예시
60000000 10000000 1000000	5780000

문제 2 [12점]

우리가 흔히 사용하는 신용카드에는 16개의 고유번호가 존재한다. 주민등록번호가 사람의 고유한 정보를 담고 있듯 카드번호에도 다양한 정보가 담겨있다. 이러한 카드번호에는 신용카드 브랜드, 발급사 구분, 발급사 임의의 규칙 번호, 검증 값 등이 담겨있다. 특히 앞자리는 어떤 카드 브랜드인지 판별해 주는 번호가 담겨있다. 이러한 정보들을 통하여 카드번호의 유효성을 판단하고, 브랜드를 출력해주는 프로그램을 만들고자 한다.

검증 값을 이용해 신용카드번호가 유효한지 판별하는 방법은 다음과 같다.

1) 맨 오른쪽 숫자 '2'부터 1번째 숫자라고 가정한다면, 짝수 번째 있는 숫자에 ×2를 한다.
2) 곱해진 숫자 값 중 두 자릿수가 된 값은 십의 자리와 일의 자리를 더한 값으로 바꾼다.
3) 위의 작업이 처리된 짝수 번째 번호와 홀수 번째 번호를 모두 더한다.
4) 모두 더해진 값을 검증 값이라고 부른다. 이 검증 값이 10의 배수일 경우 카드번호가 유효하다고 판별한다.

번호	4	2	3	4	4	3	2	1	1	2	3	4	3	3	9	2	검증값
×2	8		6		8		4		2		6		6		18		
변환	8	2	6	4	8	3	4	1	2	2	6	4	6	3	9	2	70

앞자리 번호	카드 브랜드
35XX	JCB
37XX	American Express
4XXX	VISA
51XX~55XX	MasterCard
65XX	BC Global
9XXX	Local

카드번호를 입력했을 때 어떤 카드 브랜드인지, 입력된 카드번호가 유효한 카드번호인지 판별해 주는 프로그램을 작성하시오.

입력

(1) 첫 번째 줄에 카드번호를 입력받는다.
(2) 카드번호는 띄어쓰기로 16개의 숫자가 4개씩 4부분으로 구분되어 입력된다.

출력

(1) 각 부분마다 4개의 숫자(0~9)가 입력되지 않을 시 -1을 출력한다.
(2) 카드번호는 숫자가 입력되었지만 입력된 카드 브랜드 외의 번호일 경우 -2을 출력한다.
(3) 카드 브랜드 명칭은 대소문자를 구분해 출력한다.
(4) 카드 브랜드와 검증 값, 카드번호가 유효한지를 순서대로 띄어쓰기로 구분하여 출력한다.
(5) 카드번호가 유효하다면 'O'를, 유효하지 않다면 'X'를 출력한다.

형식

입력 예시	출력 예시
12 12 12 1234	-1
1566 1231 1231 5546	-2
4234 4321 1234 3392	VISA 70 O
9420 1144 2474 7246	Local 70 O

문제 3 [12점]

W(가로)×H(세로) 크기의 텍스트 화면(콘솔창 또는 터미널창이라고 생각해도 무방)에서 작동하는 슈팅 게임을 제작하려고 한다. 가장 왼쪽 위의 좌표는 (1, 1)이며, 가장 오른쪽 아래의 좌표는 (W, H)에 해당한다. 슈팅 게임에는 자기(플레이어의 기체) 이외에도 N개의 적기가 등장하게 되는데 적기의 움직임은 사전에 정해진 위치에서 이동 계획에 따라서만 움직인다. 이동 계획은 다음 네 가지 의미를 가지는 문자가 연결된 문자열로 기록된다.

<W가 5, H가 4인 게임판의 좌표 예시>

(1,1)	(2,1)	(3,1)	(4,1)	(5,1)
(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
(1,3)	(2,3)	(3,3)	(4,3)	(5,3)
(1,4)	(2,4)	(3,4)	(4,4)	(5,4)

<이동 계획 문자>

- L: 왼쪽으로 이동
- R: 오른쪽으로 이동
- U: 위로 이동
- D: 아래로 이동

적기가 W × H 크기의 공간을 벗어나는 움직임은 무시된다. 예를 들어 (1,1) 위치에서 L 또는 U를 만나면 무시된다.

게임 정보를 입력받아 적기의 최종 위치(좌표)를 순서대로 출력하는 프로그램을 작성하시오.

입력

(1) 첫 번째 줄에는 게임판의 크기 W(가로), H(세로), 등장하는 적기의 개수 N이 공백으로 구분되어 차례대로 입력된다. ($1 \le W \le 120$, $1 \le H \le 120$, $1 \le N \le 20$)

(2) 두 번째 줄부터 N개의 적기 정보가 공백으로 구분되어 입력되는데 첫 번째 문자열은 적기의 시작 좌표이고, 두 번째 문자열은 적기의 이동 계획을 의미한다.

출력

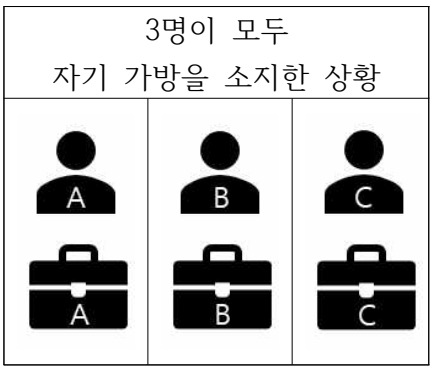
적기 정보가 입력된 순서대로 최종 위치를 계산하여 한 줄에 한 개씩 순서대로 출력한다.

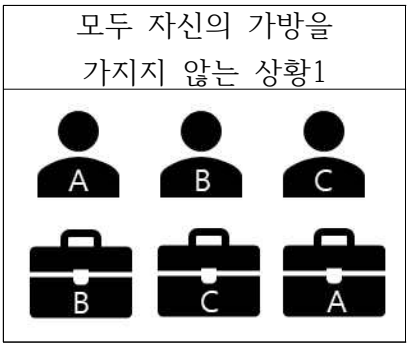
형식

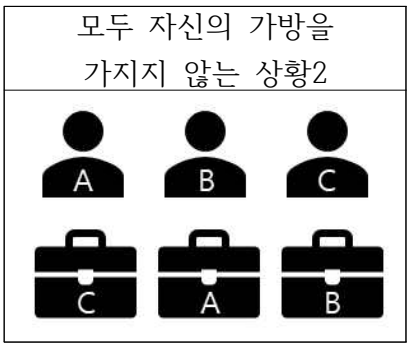
입력 예시	출력 예시
12 8 2 1,1 RRUDD 3,4 LLDDRD	3,3 2,7

문제 4 [12점]

N명의 친구가 모여있다. N명이 가지고 있는 가방을 모두 수거했다가 다시 돌려주었을 때, 아무도 본인의 가방을 받지 않는 상황의 가짓수를 구하는 프로그램을 작성하시오.







입력

입력으로 사람 수 N을 입력받는다($1 \leq N \leq 20$).

출력

(1) 아무도 본인의 가방을 받지 않는 모든 상황의 가짓수를 출력한다.
(2) 입력되는 N이 입력범위를 벗어나면 "오류"를 출력한다.

형식

입력 예시	출력 예시
2	1
3	2

문제 5 [12점]

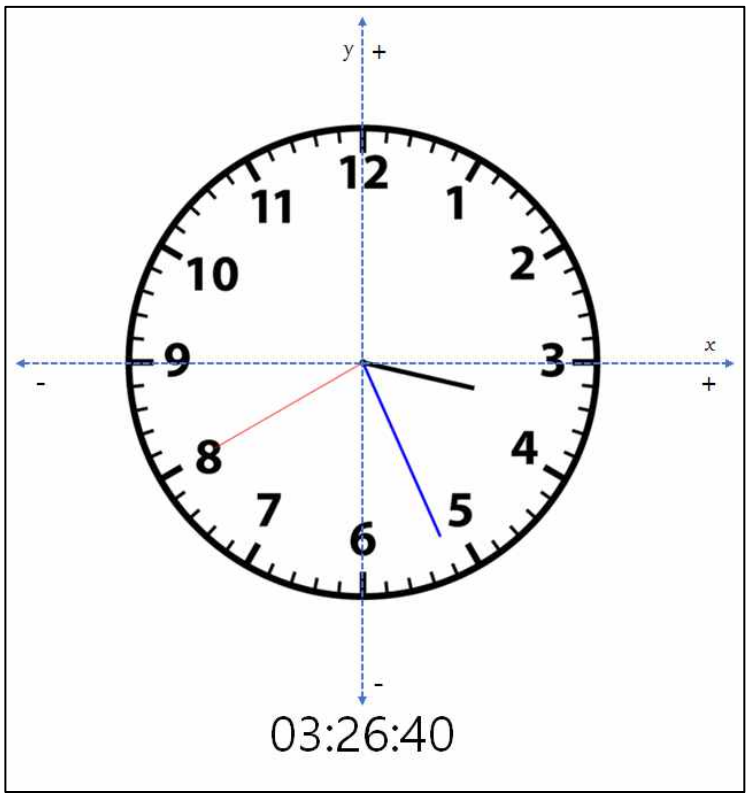
좌표평면에 아래 같이 작동하는 아날로그 시계를 그리려고 한다.

시계의 초침, 분침, 시침은 00시 00분 00초 위치에서 정렬된 상태로 시작하며 회전 방향은 왼쪽에서 오른쪽 방향이다(시계방향).

초침은 1초마다 6°씩 움직이므로 1분(60초)마다 시계판을 1회 회전하게 된다. 초침이 0초에서 시작하여 1회전을 완료하기 전까지 분침과 시침은 움직이지 않는다.

분침도 초침과 마찬가지로 1분마다 6°씩 움직이므로 1시간(60분)마다 시계판을 1회 회전하게 된다.

시침은 1분마다 0.5°씩 움직이며, 따라서 12시간마다 시계 판을 1회 회전하게 된다.



<좌표평면 위에 그린 시계 그림 예시>

초침, 분침, 시침의 시곗바늘은 직선으로 표현하며 좌표평면의 원점 (0, 0)에서 그리기 시작하여 각 바늘의 길이가 초침 180, 분침 200, 시침 120이 되도록 그려야 한다.

위와 같은 시계 그리기 조건에서 특정 시간이 입력되면 해당 시간에 해당하는 시침, 분침, 초침의 좌표를 계산하여 출력하는 프로그램을 작성하시오.

입력

(1) 첫 번째 줄에 시간이 입력된다.

(2) 시간은 시(H):분(M):초(S)의 형태로 입력되며, 시, 분, 초는 모두 정수이고, $0 \le H \le 12$, $0 \le M, S \le 59$ 의 범위를 갖는다.

출력

(1) 입력된 시간에 해당하는 시침, 분침, 초침의 좌표를 한 줄에 하나씩 순서대로 출력한다.

(2) 좌표는 (x, y) 형태로 출력한다.

(3) 좌푯값은 소수점 첫 번째 자리에서 반올림하여 정수로 출력한다.

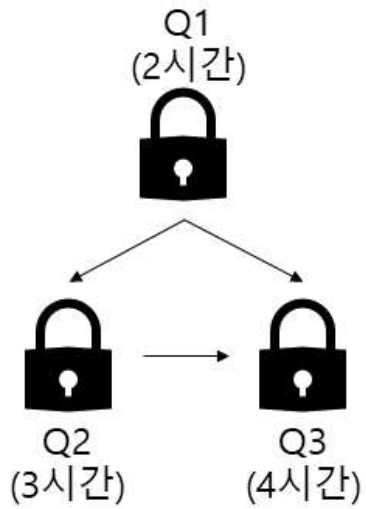
형식

입력 예시	출력 예시
09:00:30	(-120, 0) (0, 200) (0, -180)
03:26:40	(117, -27) (81, -183) (-156, -90)

문제 6 [12점]

게임은 퀘스트를 진행하는 재미가 있다. 각 퀘스트를 진행하기 위해서는 해당 퀘스트를 하기 전에 선행 퀘스트를 진행해야 하며, 각 퀘스트는 진행하는데 필요한 시간이 각각 존재한다. 이를테면 2번 퀘스트(Q2)를 하기 전에는 1번 퀘스트(Q1)가 선행 퀘스트로 존재하고 3번 퀘스트(Q3)를 하기 전에는 1번 퀘스트(Q1)와 2번 퀘스트(Q2)를 모두 수행해야 3번 퀘스트(Q3)를 진행할 수 있다고 가정하고 아래의 그림을 보자.

- 1번 퀘스트(Q1) : 2시간
- 2번 퀘스트(Q2) : 3시간
- 3번 퀘스트(Q3) : 4시간



1번 퀘스트(Q1)를 끝내려면 2시간, 2번 퀘스트(Q2)를 끝내려면 3시간, 3번 퀘스트(Q3)를 끝내려며 4시간이 걸린다고 하자. 3번 퀘스트(Q3)까지 깨기 위한 시간을 구한다고 했을 때, 3번 퀘스트(Q3)를 깨기 전 1번 퀘스트(Q1), 2번 퀘스트(Q2)를 모두 끝내야 하므로 기본적으로 5시간이 걸리며, 3번 퀘스트(Q3) 수행시간까지 합하여 9시간이라는 결과가 나온다.

게임플레이를 위한 N개의 퀘스트가 있을 때 M번 퀘스트까지 깨는 데 필요한 최소 수행시간을 구하는 프로그램을 작성하시오.(단, 퀘스트 번호 순서대로 선행퀘스트를 의미하는 것은 아니며, 선행퀘스트가 없는 퀘스트가 여러 개일 때, 동시 진행이 가능하다)

입력

(1) 첫 번째 줄에는 퀘스트의 개수 $N(1 \le N \le 200)$개와 깨고 싶은 M번 $(1 \le M \le 200)$ 퀘스트를 입력한다.
(2) 두 번째 줄부터 N개의 줄이 입력되며, 각 줄은 순차적으로 퀘스트 번호를 의미한다. 각 줄에 입력되는 값의 첫 번째는 해당 퀘스트 수행에 필요한 시간, 그 이후로는 해당 퀘스트를 수행하기 전에 수행해야 하는 선행 퀘스트의 번호가 주어진다.
(3) 각 줄에 선행 퀘스트는 -1 값이 입력될 때까지 입력받는다.
(4) 1번 퀘스트(Q1)의 선행 퀘스트는 존재하지 않는다.

출력

*M*번째 퀘스트까지 깨는 데 필요한 최소 시간을 출력한다. 퀘스트가 깨지지 않는 경우는 존재하지 않는다.

형식

입력 예시	출력 예시
3 3 2 -1 3 1 -1 4 1 2 -1	9
4 4 1 -1 2 1 -1 3 -1 4 1 2 3 -1	7
4 1 -1 2 -1 3 1 2 -1 4 3 -1	9

문제 7 [14점]

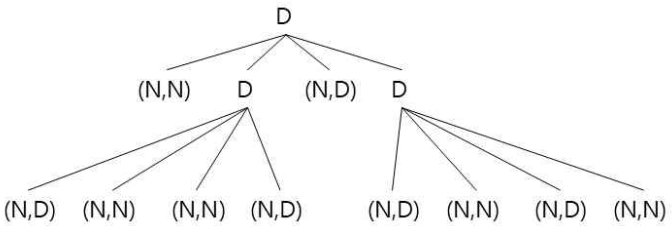
$N \times N$ 크기의 정사각형 종이가 있다. 이 종이에는 1×1 크기의 정사각형 종이가 세분되어 있다. 1×1 정사각형에 0과 1의 값을 모두 채우고 $N \times N$ 크기의 사각형을 $\frac{N}{2} \times \frac{N}{2}$ 크기로 나누었을 때 개별 사각형 안의 값이 모두 같은 값을 가지지 않는다면 해당 사각형을 위와 같이 다시 쪼개는 방식으로 진행된다. 예를 들면, <그림 1>과 같은 정사각형 모양의 종이를 모두 쪼개게 된다면 <그림 2>와 같이 쪼개지게 된다. 이 정사각형 종이를 메모리에 쉽게 저장하기 위해서 <그림 3>과 같은 모양으로 트리를 만들고 이를 CODE로 만들어 최소크기로 압축하여 저장하게 된다. <그림 3>의 트리는 가장 큰 $N \times N$ 크기의 정사각형 종이에서부터 시작하여 이 정사각형 종이가 나누어지면 'D', 나눠지지 않는다면 'N'을 배치하게 되는데 나눠지지 않을 때는 정사각형 안의 값을 보고 0이면 'N', 1이면 'D'를 함께 배치하게 된다. <그림 3>의 트리를 위의 값에서부터 순서대로 나열하게 되면 CODE를 만들 수 있으며, 우리는 이 문자열을 효율적으로 압축하여 메모리에 저장하고자 한다.

0	0	1	0
0	0	0	1
1	1	1	0
1	1	1	0

<그림1>

0	0	1	0
0	0	0	1
1	1	1	0
1	1	1	0

<그림2>



<그림 3>

기존 CODE : DNNDNDDNDNNNNNDNDNNNDNN(길이 : 23)
압축 CODE : D2NDN2DND5NDND3ND2N(길이 : 19)

CODE를 효율적으로 압축하는 방식은 더 짧은 문자열로 만들어 메모리를 적게 사용하는 데에 목표를 둔다. CODE를 압축하는 방법은 다음과 같다. 예를 들어 "DDDNNN"의 경우 문자를 한 개 단위로 잘라서 압축한다면 "3D3N"으로 압축하여 길이를 6에서 4로 줄일 수 있다. 잘린 문자와 붙어있는 동일한 문자의 개수를 이용하여 압축하는 방식을 사용한다. 한 개 단위로 자른다면 D, D, D, N, N, N이므로 3개의 D, 3개의 N이 있으므로 "3D3N"으로 압축가능하다. "DNDDND"와 같은 경우 문자를 한 개 단위로 잘라서 압축한다면 "DN2DND"로 되어 압축 전과 동일한 길이인 6으로 메모리에 저장해야 한다. 하지만 문자를 3개 단위로 잘라서 압축한다면 DND, DND 2개로 만들 수 있으므로 "2DND"와 같이 길이를 6에서 4로 만들어 효율적으로 압축하는 것이 가능하다. 단, 문자를 자르는 단위의 개수는 변화하지 않으며, CODE의 첫 부분부터 단위 개수만큼 자른다(1개 단위로 자르다가 2개 단위로 자르지 못함).

0, 1로 채워진 $N \times N$ 크기의 정사각형 종이를 입력받았을 때 위와 같은 방식으로 정사각형을 쪼개고, 쪼갠 정사각형들을 이용하여 만들어진 CODE를 출력하고, CODE를 가장 효율적으로 압축했을 때의 길이를 띄어쓰기를 이용하여 구분되게 출력하는 프로그램을 작성하시오.

입력

(1) 첫 번째 줄에는 정사각형 한 변의 크기 N을 입력받는다($N = 2^x$(x는 자연수이며, x값의 범위는 $0 < x < 6$이다).
(2) 다음 N개의 줄에는 $N \times N$ 크기의 이진 데이터가 모두 입력된다.

출력

출력은 압축되지 않은 CODE와 가장 효율적으로 압축되었을 때 코드 길이를 띄어쓰기로 구분하여 출력한다.

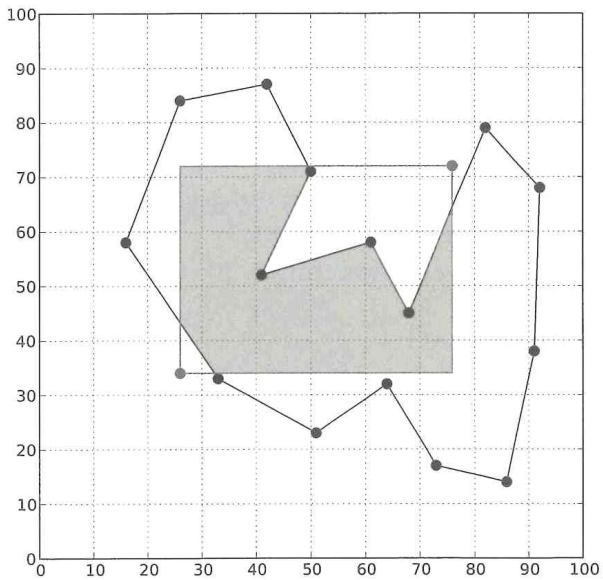
형식

입력 예시	출력 예시
2 0 1 1 0	DNNNDNDNN 8
4 0 0 1 0 0 0 0 1 1 1 1 0 1 1 1 0	DNNDNDDNDNNNNNDNDNNNDNN 19

문제 8 [14점]

영수는 금괴가 숨겨진 보물섬을 오랜 고난을 거쳐 찾아냈다. 이 보물 섬 어디엔가 금괴가 묻혀 있다고 하는데, 그 위치가 어디인지는 모른다. 섬을 다 뒤집어엎기 직전, 지도에는 여백이 부족해 적지 않았던 금괴의 위치가 적힌 쪽지를 발견했다. 이 쪽지에는 금괴가 묻혀 있는 곳이 적혀 있는데, 설명이 길고 장황한 데다 군데군데 지워져 정확히 알아보기 쉽지 않았다. 쪽지를 열심히 연구한 결과, 금괴가 묻혀 있는 곳의 범위를 어느 정도 좁힐 수 있었다.

섬의 지도가 그림과 같이 N개의 점을 갖는 다각형으로 주어질 때, 금괴가 묻혀 있을 수 있는 곳은 두 점(x_1, y_1)과 (x_2, y_2)를 서로 대칭인 꼭지점으로 갖고, 네 변이 모두 x 축 혹은 y 축에 평행한 직사각형 내부이다. 영수는 이 직사각형 내에 포함된 육지를 전부 조사하고 싶다. 영수가 조사해야 할 부분의 넓이를 계산하는 프로그램을 작성하시오.



입력

(1) 첫 줄에는 다섯 개의 정수로 직사각형의 꼭지점의 좌표 x_1, y_1, x_2, y_2 $(0 \le x_1 < x_2 \le 100, 0 \le y_1 < y_2 \le 100)$ 그리고 섬의 지도를 나타내는 다각형의 꼭지점의 수 N이 주어진다.

(2) 그 후 N줄$(3 \le N \le 100)$에 각 두 개의 정수로 각 꼭지점의 좌표 x_i, y_i $(0 \le x_i, y_i \le 100)$가 주어진다. 꼭지점은 시계 반대 방향으로 주어지며, 마지막 점은 첫 번째 점과 연결되어 있다.

(3) 주어진 섬의 면적이 0인 경우 혹은 섬의 경계선이 자기 자신과 교차하거나 겹치는 경우는 없다.

출력

조사해야 할 육지의 넓이를 출력한다(소수점 이하 6번째 자리에서 반올림하여 5번째 자리까지 출력한다).

형식

입력 예시	출력 예시
26 34 76 72 15 41 52 50 71 42 87 26 84 16 58 33 33 51 23 64 32 73 17 86 14 91 38 92 68 82 79 68 45 61 58	1343.09487