

به نام خدا



دانشگاه صنعتی شریف  
دانشکده مهندسی کامپیوتر

درس مبانی برنامه‌سازی

پیک شادی  
(تمرین سری دوم)  
سال نو مبارک



## توضیحات :

- قبل از شروع موارد زیر را به دقت بخوانید.
- در خط اول هر یک از برنامه‌ها شماره دانشجویی خود را به فرمت زیر بنویسید:

// ID: your Student ID

مثال:

// ID: 87654321

- بعد از نوشتن همه برنامه‌ها ، همه آن‌ها را در یک فولدر با نام assignment#2\_stdID قرار دهید( که در آن stdID شماره دانشجویی شما می باشد). مثلاً:  
assignment#2\_87654321 و سپس این فولدر را فشرده کنید<sup>1</sup> . فایل فشرده شده را به آدرس [cpphomework852@gmail.com](mailto:cpphomework852@gmail.com) ارسال نمایید.

- عنوان ایمیل باید به صورت assignment#2\_stdID باشد ، که در آن stdID شماره دانشجویی شما می باشد.

- اشتباه در نام‌گذاری فایل برنامه‌ها و فایل فشرده شده و یا عنوان ایمیل فرستاده شده ، موجب عدم تصحیح تمرین شما می‌شود.

- فرمت ورودی و خروجی برنامه‌های شما باید دقیقاً به صورت گفته شده در مسئله باشد ، در غیر این صورت چون برنامه‌های شما به صورت خودکار تصحیح می شود، خروجی برنامه‌ی شما با خروجی برنامه‌ی مصحح منطبق نخواهد بود.

- به هر یک از برنامه‌های شما تعدادی تست داده خواهد شد. توصیه می شود که حتماً یک بار تست‌های نمونه را به برنامه‌های تان بدهید.

- به برنامه‌های مشابه به میزان نمره سوال ، نمره منفی تعلق خواهد گرفت .

- موفق باشید.

---

<sup>1</sup> برای این کار کافی است روی فولدر کلیک راست کنید و از منوی باز شده Send to و سپس Compressed (Zipped) Folder را انتخاب کنید. با این کار یک فایل به نام assignment#1\_stdID.zip ساخته می شود .

## مسئله اول

### نام فایل برنامه : **BigNumber.cpp**

در این مسئله شما باید دو عدد را از ورودی بخوانید و بتوانید سه عملیات زیر را روی آنها انجام دهید :

- i. دو عدد را با هم جمع کنید .
  - ii. دو عدد را در هم ضرب کنید .
  - iii. عدد اول را به توان عدد دوم برسانید .
- و نتیجه را در خروجی چاپ کنید . به همین راحتی !!!  
توجه داشته باشید که جواب ها ( و یا حتی ورودی ها ) می توانند تا 1000 رقم داشته باشند. ( یعنی خیلی بزرگ هستند.)

**راهنمایی 1:** انواع عددی استاندارد<sup>1</sup> در C حداکثر اعداد تا 60 رقم را پشتیبانی می کنند. بنابراین نمی توانید از انواع عددی C استفاده کنید و باید فکر دیگری بکنید.

**راهنمایی 2:** برای جمع کردن می توانید ورودی ها را به صورت آرایه ای از کاراکترها بگیرید و وظیفه خطیر جمع کردن را خود به عهده بگیرید !

**راهنمایی 3:** اگر جمع را به صورت تابع بنویسید ، پیاده سازی ضرب و توان به آسانی انجام پذیر است.

### قالب ورودی و خروجی ها :

ورودی را به صورت استاندارد ( از صفحه کلید ) بخوانید . در خط اول ،  $n$  تعداد پرسش ها از برنامه شما می آید . در  $n$  خط بعدی در هر خط می تواند یکی از قالب های زیر بیاید :

**ADD num1 num2**

در این حالت اعداد  $num1$  و  $num2$  را با هم جمع می کنید.

**MUL num1 num2**

در این حالت اعداد  $num1$  و  $num2$  را در هم ضرب می کنید.

**POW num1 num2**

در این حالت عدد  $num1$  را به توان عدد  $num2$  می رسانید.

در هر سه حالت ، جواب را در یک خط در خروجی استاندارد چاپ می کنید.  
**تذکر :** دقت کنید که همه حروف ADD ، MUL و POW بزرگ است .

نمونه ورودی و خروجی ها را در زیر مشاهده می کنید :

ورودی نمونه :	خروجی نمونه :
4 ADD 9999999999 2 MUL 100 123456789 POW 2 100 ADD 20 -100	10000000001 12345678900 1267650600228229401496703205376 -80

<sup>1</sup> مثل int ، longint ، int64 و ...

مسئله دوم

نام فایل برنامه : Interval.cpp

در این مسأله شما باید پس از دریافت حدود 5 بازه از ورودی، اجتماع آنها را بیابید و در خروجی چاپ کنید. توجه کنید که باید حتی‌الامکان حدود کمتری را برای بازه‌ها ذکر کنید. مثلاً مجموع دو بازه‌ی (1و5) و (3و8) را به صورت (1و8) در نظر بگیرید.

قالب ورودی و خروجی ها :

ورودی را در 5 خط و در هر خط 2 عدد بخوانید. هر خط نشانه‌ی یک بازه است. عدد اول هر خط نشانه‌ی حد سمت چپ و عدد دوم نشانه‌ی حد سمت راست بازه‌اند. خروجی می‌تواند شامل چندین بازه شود. لذا در خط اول آن تعداد بازه‌ها (n) را بنویسید و سپس در هر یک از n سطر بعد حدود یکی از بازه‌ها را بنویسید. توجه داشته باشید که باید بازه‌ها را به ترتیب صعودی بنویسید.

ورودی نمونه :	خروجی نمونه :
-1 4	3
4 9	-30 -20
-6 2	-6 9
12 21	12 21
-30 -20	

مسئله سوم

نام فایل برنامه : Check.cpp

Your task is to write a program that reads a chessboard configuration and identifies whether a king is under attack (in check). A king is in check if it is on square which can be taken by the opponent on his next move.

White pieces will be represented by uppercase letters, and black pieces by lowercase letters. The white side will always be on the bottom of the board, with the black side always on the top.

For those unfamiliar with chess, here are the movements of each piece:

**Pawn (p or P):** can only move straight ahead, one square at a time. However, it takes pieces diagonally, and that is what concerns you in this problem.

**Knight (n or N):** has an L-shaped movement shown below. It is the only piece that can jump over other pieces.

**Bishop (b or B):** can move any number of squares diagonally, either forward or backward.

**Rook (r or R):** can move any number of squares vertically or horizontally, either forward or backward.

**Queen (q or Q):** can move any number of squares in any direction (diagonally, horizontally, or vertically) either forward or backward.

**King (k or K):** can move one square at a time in any direction (diagonally, horizontally, or vertically) either forward or backward.

Movement examples are shown below, where "\*" indicates the positions where the piece can capture another piece:

Pawn	Rook	Bishop	Queen	King	Knight
.....	*.....	.....*	*...*	.....	.....
.....	*.....	*...*	*...*	.....	.....
.....	*.....	*...*	***	.....	*...*
.....	*.....	*...*	***	***	*...*
...p...	***r****	...b...	***q****	..*k...	...n...
*...*	*.....	*...*	***	***	*...*
.....	*.....	*...*	***	.....	*...*
.....	*.....	*...*	*...*	.....	.....

Remember that the knight is the only piece that can jump over other pieces. The pawn movement will depend on its side. If it is a black pawn, it can only move one square diagonally down the board. If it is a white pawn, it can only move one square diagonally up the board. The example above is a black pawn, described by a lowercase "p". We use "move" to indicate the squares where the pawn can capture another piece.

### Input

There is a board configuration in the input, consisting of eight lines of eight characters.

A "." denotes an empty square, while upper and lowercase letters represent the pieces as defined above. There will be no invalid characters and no configurations where both kings are in check.

The board will contain exactly one white king and one black king.

### Output

You must output one of the following answers:

white king is in check.

black king is in check.

no king is in check.

Sample input	Sample Output
<pre>..k.... ppp.pppp ..... .R..B.. ..... PPPPPPPP K.....</pre>	black king is in check.
<pre>rnqk.nr ppp.ppp ...p... ...p... .bPP... .....N.. PP..PPPP RNBQKB.R</pre>	white king is in check.

مسئله چهارم  
نام فایل برنامه: **Jolly.cpp**

A sequence of  $n > 0$  integers is called a *jolly jumper* if the absolute values of the differences between successive elements take on all possible values 1 through  $n-1$ . For instance,

1 4 2 3

is a jolly jumper, because the absolute differences are 3, 2, and 1, respectively. The definition implies that any sequence of a single integer is a jolly jumper. Write a program to determine whether each of a number of sequences is a jolly jumper.

**Input**

Each line of input contains an integer  $n < 3,000$  followed by  $n$  integers representing the sequence.

**Output**

For each line of input generate a line of output saying "Jolly" or "Not jolly".

Sample input	Sample Output
4 1 4 2 3 5 1 4 2 -1 6	Jolly Not jolly

مسئله پنجم  
نام فایل برنامه: **Poker.cpp**

A poker deck contains 52 cards. Each card has a suit of either clubs, diamonds, hearts, or spades (denoted C, D, H, and S in the input data). Each card also has a value of either 2 through 10, jack, queen, king, or ace (denoted 2, 3, 4, 5, 6, 7, 8, 9, T, J, Q, K, A). For scoring purposes card values are ordered as above, with 2 having the lowest and ace the highest value. The suit has no impact on value.

A poker hand consists of five cards dealt from the deck. Poker hands are ranked by the following partial order from lowest to highest.

**High Card.** Hands which do not fit any higher category are ranked by the value of their highest card. If the highest cards have the same value, the hands are ranked by the next highest, and so on.

**Pair.** Two of the five cards in the hand have the same value. Hands which both contain a pair are ranked by the value of the cards forming the pair. If these values are the same, the hands are ranked by the values of the cards not forming the pair, in decreasing order.

**Two Pairs.** The hand contains two different pairs. Hands which both contain two pairs are ranked by the value of their highest pair. Hands with the same highest pair are ranked by the value of their other pair. If these values are the same the hands are ranked by the value of the remaining card.

**Three of a Kind.** Three of the cards in the hand have the same value. Hands which both contain three of a kind are ranked by the value of the three cards.

**Straight.** Hand contains five cards with consecutive values. Hands which both contain a straight are ranked by their highest card.

**Flush.** Hand contains five cards of the same suit. Hands which are both flushes are ranked using the rules for High Card.

**Full House.** Three cards of the same value, with the remaining two cards forming a pair. Ranked by the value of the three cards.

**Four of a Kind.** Four cards with the same value. Ranked by the value of the four cards.

**Straight Flush.** Five cards of the same suit with consecutive values. Ranked by the highest card in the hand.

Your job is to compare several pairs of poker hands and to indicate which, if either, has a higher rank.

### Input

The input file contains several lines (up to 10 lines), each containing the designation of ten cards: the

first five cards are the hand for the player named "*Black*" and the next five cards are the hand for the player named "*White*".

### Output

For each line of input, print a line containing one of the following:

Black wins.

White wins.

Tie.

Sample input	Sample Output
2H 3D 5S 9C KD 2C 3H 4S 8C AH	White wins.
2H 4S 4C 2D 4H 2S 8S AS QS 3S	Black wins.
2H 3D 5S 9C KD 2C 3H 4S 8C KH	Black wins.
2H 3D 5S 9C KD 2D 3H 5C 9S KH	Tie.

- تمرین های خود را حداکثر تا ساعت 12 نیم شب 1385/1/14 ارسال نمایید. تاریخ تحویل تمرین به هیچ وجه تمدید نمی شود.