

## FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '90

1.1 Write a program to produce the following initials for NCNB National Bank:

```

NN      N  CCCCC  NN      N   BBBB
N N     N  C      N N     N   B  B
N  N   N  C      N  N   N   BBBBB
N   N N  C      N   N N   B  B
N      NN  CCCCC  N      NN   BBBB

```

1.2 NCNB programmers in Tampa use an IBM 3090 model 400 mainframe computer that has an MVS/XA (Multiple Virtual Storage/ Extended Architecture) operating system. The machine is partitioned into two logical processors: SYSTEM 1 is used for testing and all day-time batch jobs; SYSTEM 2 is used to support on-line programs and some night-time batch jobs. Write a program to display the processor name given the system #. Examples:

```

INPUT: Enter #: 1           INPUT: Enter #: 2
OUTPUT: SYSTEM 1         OUTPUT: SYSTEM 2

```

1.3 With assets over 66 billion dollars in 1989, NCNB is the largest bank in the southeast United States. Let's assume that the Tampa Application Systems Center staffs 66 programmers. If there is a direct correlation between the number of programmers staffed in Tampa and the amount of money the bank has in assets, then display the total amount of assets NCNB will have if N more programmers are hired. Example:

```

INPUT: Enter N: 4
OUTPUT: 70 BILLION DOLLARS

```

1.4 Write a program to help the post office determine the county in which a person lives, given his/her zip code. Use the following as data:

```

HILLSBOROUGH - 33612  33613  33620  33510
PINELLAS -      33701  34685  34646
PASCO -        33525  34249  34690

```

Example:

```

INPUT: Enter zip code: 33525
OUTPUT: PASCO

```

**1.5** NCNB's dynamic chairman, Hugh McColl, resides at the national headquarters in Charlotte, North Carolina. His secretary would like to issue a statement to all the employees concerning McColl's financial goals. Write a program that accepts two numbers (MMM - amount in billions of dollars, and YYYY - year), and then produces a statement in the following form:

```
HUGH MCCOLL WOULD LIKE NCNB TO GROW
TO MMM BILLION DOLLARS IN ASSETS BY
THE YEAR YYYY
```

Example:

```
INPUT: Enter MMM: 100
Enter YYYY: 1995
```

```
OUTPUT: HUGH MCCOLL WOULD LIKE NCNB TO GROW
TO 100 BILLION DOLLARS IN ASSETS BY
THE YEAR 1995
```

**1.6** Vickie manages 7 people in the Trust Division at NCNB. Her associates pay an average of 50,000 coupons monthly. The work load is divided as evenly as possible among her associates so that no one person has more than one extra coupon to pay than another person. Write a program to calculate the MAXIMUM number of coupons that any one of N associates must pay if C coupons come in for payments. Examples:

```
INPUT: Enter N associates: 7
Enter C coupons: 49000
OUTPUT: 7000
```

```
INPUT: Enter N associates: 7
Enter C coupons: 49002
OUTPUT: 7001
```

**1.7** NCNB programmers code their programs primarily in COBOL: COmmon Business Oriented Language. Each program is required to have these four divisions (in order): IDENTIFICATION, ENVIRONMENT, DATA, and PROCEDURE. Write a program to display those divisions coming before and after a division given as input. Multiple divisions on a line are to be displayed in order and with 2 spaces in between. If none occur before/after entered division, then display NONE for that part. Examples:

```
INPUT: Enter division: DATA
OUTPUT: BEFORE = IDENTIFICATION ENVIRONMENT
AFTER = PROCEDURE
```

```
INPUT: Enter division: IDENTIFICATION
OUTPUT: BEFORE = NONE
AFTER = ENVIRONMENT DATA PROCEDURE
```

**1.8** NCNB is the largest banking company in the South and the 7th largest in the nation. NCNB has statewide banks in four states: Florida, North Carolina, South Carolina, and Texas. It also has banks in Baltimore, Atlanta, and northern Virginia. For 1990, NCNB will recognize the following number of holidays in each of the states: 10-FL, 8-NC, 7-SC, 10-TX, 11-MD, 10-GA, 10-VA. Write a program to display the states that recognize at least N holidays, where N is input as a number between 5 and 11 inclusive. States must be displayed in the order given above with one space in between each state. Example:

INPUT: Enter N: **10**  
OUTPUT: **FL TX MD GA VA**

**1.9** Anno Domini is Latin for "in the year of our Lord" and is abbreviated as A.D. In 525 A.D., Pope John I asked the monk Dionysius Exiguus to begin a Christian system of dating events, starting with the year Dionysius believed Christ was born. The years before the birth of Christ are termed B.C., while the years after the birth of Christ are termed A.D. The following is the order of years: ... 2 B.C., 1 B.C., 1 A.D., 2 A.D., ... with Christ's birth being in the year 1 A.D. Today, we know that the monk was in error. Even though we continue to use his dating system, biblical scholars currently believe that Christ was born four years earlier than what the monk thought. Write a program that corrects modern dates to account for this change. Examples:

INPUT: Enter date: **4**  
Enter A.D. or B.C.: **B.C.**  
OUTPUT: **1 A.D.**

INPUT: Enter date: **1**  
Enter A.D. or B.C.: **A.D.**  
OUTPUT: **5 A.D.**

INPUT: Enter date: **5**  
Enter A.D. or B.C.: **B.C.**  
OUTPUT: **1 B.C.**

**1.10** Write a program to display a word diamond for a 7 letter word. The following examples will illustrate the format of a word diamond. Examples:

INPUT: Enter word: **CONTEST**

OUTPUT:           **T**  
                  **NTE**  
                  **ONTES**  
                  **CONTEST**  
                  **ONTES**  
                  **NTE**  
                  **T**

INPUT: Enter word: **PROBLEM**

OUTPUT:           **B**  
                  **OBL**  
                  **ROBLE**  
                  **PROBLEM**  
                  **ROBLE**  
                  **OBL**  
                  **B**

**2.1** Write a program to encode a phrase for the Security department. Each letter in the phrase is to be replaced by the letter that precedes it in the alphabet (B becomes A, C becomes B, ... Z becomes Y), except that the letter A becomes Z. All other characters remain the same. Example:

INPUT: Enter phrase: **THIS PERSON IS A THIEF**

OUTPUT: **SGHR ODQRNM HR Z SGHDE**

**2.2** Write a program to determine if a given year is the BEGINNING/END of a DECADE/CENTURY/MILLENNIUM. Year 1 is the first year of our calendar, the year traditionally looked on as the year of Christ's birth. Year 1 began the first decade and the first century and the first millennium. The year 2000 will be the end of a decade and a century and a millennium. Input will be a year between 1 and 2000 inclusive. Output will be all valid possibilities of the form XXX OF YYY, where XXX is either BEGINNING or END, and YYY is either DECADE, CENTURY, or MILLENNIUM. For more than one line of output, the order must be DECADE, CENTURY, MILLENNIUM. Examples:

INPUT: Enter year: **1900**

OUTPUT: **END OF DECADE  
END OF CENTURY**

INPUT: Enter year: **1991**

OUTPUT: **BEGINNING OF DECADE**

**2.3** Bob, Doug, Jackie, and Jose bowl in the NCNB bowling league. Given the scores of each of their 3 games, display each person's average and handicap. If a person has an average over 200, then his handicap is 0. Otherwise, the handicap is calculated by subtracting an average from 200 and multiplying the result by 90%. Both the average and handicap must be truncated to a whole number when displayed. Example:

INPUT: Enter scores for Bob: **200, 190, 218**  
 Enter scores for Doug: **195, 207, 168**  
 Enter scores for Jackie: **109, 134, 127**  
 Enter scores for Jose: **130, 140, 144**

OUTPUT: **BOB: AVERAGE = 202 HANDICAP = 0**  
**DOUG: AVERAGE = 190 HANDICAP = 9**  
**JACKIE: AVERAGE = 123 HANDICAP = 69**  
**JOSE: AVERAGE = 138 HANDICAP = 55**

**2.4** Today we use a Gregorian Calendar, initially set up by Pope Gregory XIII in 1582. Because the previous Julian calendar was not an exact representation of a solar year, several days had accumulated since its institution. Therefore, Pope Gregory XIII decreed the elimination of 10 days from the year 1582. For many countries October 5, 1582 became October 15, 1582. However, different areas of the world did not adopt the calendar at the same time. Write a program to determine how many days to add to a Julian date to convert it to a Gregorian date. The following scale has been devised:

Add 10 days to Julian dates from 10/5/1582 to 2/28/1700  
Add 11 days to Julian dates from 3/1/1700 to 2/28/1800  
Add 12 days to Julian dates from 3/1/1800 to 2/28/1900  
Add 13 days to Julian dates from 3/1/1900 to 2/28/2100

Input will be a Julian date of the format MM/DD/YYYY and output will be of the format ADD NN DAYS. Examples:

INPUT: Enter date: **09/02/1752**      OUTPUT: **ADD 11 DAYS**

INPUT: Enter date: **03/01/1801**      OUTPUT: **ADD 12 DAYS**

**2.5** Sorting is the arranging of elements of a list into order based on the value of a particular field within the elements. There are several algorithms to accomplish the task of sorting including: bubble sort, shell sort, and quick sort. Each sorting algorithm's efficiency depends upon the number of elements to be sorted and the particular order in which they are originally given. The average number of comparisons needed to sort a list of N elements is given as:

$N * (N-1) / 2$	for BUBBLE SORT
$N * ((\text{Log base 2 of } N) \text{ squared})$	for SHELL SORT
$N * (\text{Log base 2 of } N)$	for QUICK SORT

Assuming that the efficiency depends only on the number of elements in the list, N, determine the efficiency order of the different sorting algorithms and print them from most efficient to least efficient. N will be greater than 2. Example:

INPUT: Enter N: **128**

OUTPUT: **QUICK SORT**  
**SHELL SORT**  
**BUBBLE SORT**

**2.6** During the summer and fall, NCNB has a golf league for employees. Write a program for a golfer to determine the status for each of the 9 holes, given the scores on each hole as input and using the pars listed below. The par is the score standard for each hole on a golf course. Also display the total number of strokes the golfer takes and the total par for the 9 holes. If a golfer shoots the standard score for a hole, the status is PAR. If on a hole a golfer shoots 1 score below the par, the status is BIRDIE. If the score is 2 below the par, the status is EAGLE. A score of 3 below the par is called a DOUBLE EAGLE. For scores 1 and 2 above par, the status is BOGEY and DOUBLE BOGEY respectively. Use the following pars for the holes:

```

HOLE: 1  2  3  4  5  6  7  8  9
-----
PAR:  4  3  4  5  4  3  5  4  4

```

Example:

```

INPUT: Enter score for hole 1: 3
       Enter score for hole 2: 3
       Enter score for hole 3: 5
       Enter score for hole 4: 7
       Enter score for hole 5: 6
       Enter score for hole 6: 1
       Enter score for hole 7: 2
       Enter score for hole 8: 3
       Enter score for hole 9: 4

```

```

OUTPUT: HOLE  PAR  SCORE  STATUS
-----
        1     4     3     BIRDIE
        2     3     3     PAR
        3     4     5     BOGEY
        4     5     7     DOUBLE BOGEY
        5     4     6     DOUBLE BOGEY
        6     3     1     EAGLE
        7     5     2     DOUBLE EAGLE
        8     4     3     BIRDIE
        9     4     4     PAR
        ---  -
        36   34

```

2.7 In 45 B.C. Julius Caesar instituted the use of a calendar. The Julian calendar consists of a perpetual cycle of three years of 365 days followed by one year of 366 days. However, the exact solar year consists of 365 days, 5 hours, 48 minutes, 47.8 seconds. Thus, at the end of 1 Julian year, the Julian calendar is ahead of a solar calendar that will complete its year in another 5 hours 48 minutes and 47.8 seconds. Write a program to determine how many days, hours, minutes, and seconds the Julian calendar is behind/ahead of an imaginary calendar that is based on exact solar years. Assume that both calendars start at the same time and a comparison is done at the end of N Julian years, where N is input as a natural number less than 2000. Output must have 1 space between the number and its unit; 2 spaces between the unit and the next number. Seconds are displayed using the form ##.#. Examples:

```

INPUT: Enter N: 1
OUTPUT: 0 DAYS 5 HOURS 48 MIN 47.8 SEC AHEAD

INPUT: Enter N: 2
OUTPUT: 0 DAYS 11 HOURS 37 MIN 35.6 SEC AHEAD

INPUT: Enter N: 3
OUTPUT: 0 DAYS 17 HOURS 26 MIN 23.4 SEC AHEAD

INPUT: Enter N: 4
OUTPUT: 0 DAYS 0 HOURS 44 MIN 48.8 SEC BEHIND

INPUT: Enter N: 5
OUTPUT: 0 DAYS 5 HOURS 3 MIN 59.0 SEC AHEAD

```

2.8 In June 1989, Barb and Carolyn founded a suggestion committee for the NCNB Systems and Programming Division in Tampa. In August Joe was placed on the suggestion committee. In September Carolyn resigned and Doug took her place. The committee declared that a new person from a waiting list will replace a committee member after he/she has served for six months. With the following waiting list, write a program to display the 3 committee members for every month that someone is replaced. The names on a line must follow the logical order started in the example. Input will be the last month and year that the suggestion committee meets before it disassembles. The date input will not be later than 5, 1992.

Waiting list: JACKIE, TOM, LOVETTA, GREG, TONY, AL, KAREN, JAN,  
NORM, TRUDY, THERESA, ALICE, DAVE, JIM, STEVE

Example:

```

INPUT: Enter month, year: 6, 1990

OUTPUT: 9/1989 - BARB JOE DOUG
        12/1989 - JACKIE JOE DOUG
        2/1990 - JACKIE TOM DOUG
        3/1990 - JACKIE TOM LOVETTA
        6/1990 - GREG TOM LOVETTA

```





**2.10** A new employee of the NCNB computer programming department desires training. Write a program to produce the following NCNB menu of currently available in-house computer courses. Estimate the TOTAL number of hours he/she will spend on training given the courses that he/she has selected. Upon completion of course selection, 000-000 is entered. The screen is then cleared, and the course names chosen are displayed again in the order that they were selected. Estimated hours for each course selected must appear to the right of the course name: The heading "EST. HOURS" starts 26 columns after the first column of the "COURSE NAME," as shown in the menu selection. Also, the TOTAL estimated hours must show underneath all the individual course hours in the form: TOTAL = ##.# - ## HOURS. Example:

OUTPUT:

**NCNB IN-HOUSE TRAINING LIST**

<b>COURSE #</b>	<b>COURSE NAME</b>	<b>EST. HOURS</b>
-----	-----	-----
187-11X	ISPF/PDS FUNDAMENTALS	6.5 - 8
187-15X	ISPF/PDS FOR PROGRAMMERS	4.5 - 6
220-AXX	JCL FUNDAMENTALS	15 - 20
200-AXX	VSAM CONCEPTS	4 - 7
123-2XX	MVS/SP/XA VSAM	7 - 11
130-11X	CICS/VS SKILLS I	6 - 8
130-15X	CICS/VS SKILLS II	4 - 6

INPUT: Enter course # (or 000-000 to end): **220-AXX**  
 Enter course # (or 000-000 to end): **130-11X**  
 Enter course # (or 000-000 to end): **187-11X**  
 Enter course # (or 000-000 to end): **000-000**

OUTPUT: (Screen is cleared)

<b>COURSE NAME</b>	<b>EST. HOURS</b>
-----	-----
JCL FUNDAMENTALS	15 - 20
CICS/VS SKILLS I	6 - 8
ISPF/PDS FUNDAMENTALS	6.5 - 8
	-----
TOTAL = 27.5 - 36 HOURS	



**3.3** Three local cross country teams compete in a double dual race. Each team consists of seven runners, but only the first five finishers of a team contribute to that team's score. As the runners cross the finish line, gasping for breath, the judge writes the INITIAL of the runner's team name and the NUMBER indicating the runner's finishing position, e.g. 1 for 1st, 2 for 2nd and so on. To find the score for teams A and B and to decide which of the two wins, the scorekeeper temporarily eliminates all of C's positions, and then repositions the runners from A and B into places 1 through 14. The team's score consists of the sum of the places of their first five runners. The lower team score wins. If there is a tie then the team whose sixth runner crossed the finish line first is the winner.

Write a program that computes the score for each pair of three teams and determines the winner of each pair (pairs may be displayed in any order). The program must allow the user to assign all 21 runners' team INITIAL to finishing places. Team initials can be any letter in the alphabet. Example:

INPUT: Place 1: <b>A</b>	Example of
Place 2: <b>B</b>	repositioning
Place 3: <b>A</b>	teams A and B:
Place 4: <b>B</b>	
Place 5: <b>A</b>	1: A
Place 6: <b>B</b>	2: B
Place 7: <b>C</b>	3: A
Place 8: <b>C</b>	4: B
Place 9: <b>C</b>	5: A
Place 10: <b>C</b>	6: B
Place 11: <b>B</b>	7: B
Place 12: <b>A</b>	8: A
Place 13: <b>C</b>	9: B
Place 14: <b>B</b>	10: B
Place 15: <b>C</b>	11: A
Place 16: <b>B</b>	12: A
Place 17: <b>A</b>	13: B
Place 18: <b>A</b>	14: A
Place 19: <b>C</b>	
Place 20: <b>B</b>	
Place 21: <b>A</b>	

OUTPUT: **TEAM A: 28 POINTS**  
**TEAM B: 28 POINTS**  
**TEAM B WINS!**

**TEAM A: 25 POINTS**  
**TEAM C: 31 POINTS**  
**TEAM A WINS!**

**TEAM B: 24 POINTS**  
**TEAM C: 31 POINTS**  
**TEAM B WINS!**

**3.4** Within NCNB's Application Systems Division, there are many programming teams consisting of 2 to 8 programmers and 1 team leader. Suppose that NORM is a team leader in charge of 3 programmers: AL, DOUG, and JAN. Norm has been given 30 programs (numbered 1 through 30) to distribute amongst his employees. Al is given all programs divisible by X, and Doug is given all programs divisible by Y. Jan is given all programs divisible by Z, and Norm takes the rest of the programs that were not assigned to anyone and will work on some himself or he will redistribute some later. Given X, Y, Z as different input numbers between 2 and 10 inclusive, write a program to display those program numbers shared by all 3 of the programmers, those shared between only 2, and those assigned solely to 1 person. If no programs are assigned exclusively to one particular group, then display NONE instead of program numbers. The output must be in the same format as shown: eight lines, with numbers in increasing order. Examples:

INPUT: Enter X, Y, Z: 2, 3, 5

OUTPUT: AL, DOUG, AND JAN = 30  
AL AND DOUG = 6 12 18 24  
AL AND JAN = 10 20  
DOUG AND JAN = 15  
AL = 2 4 8 14 16 22 26 28  
DOUG = 3 9 21 27  
JAN = 5 25  
NORM = 1 7 11 13 17 19 23 29

INPUT: Enter X, Y, Z: 2, 6, 9

OUTPUT: AL, DOUG, AND JAN = 18  
AL AND DOUG = 6 12 24 30  
AL AND JAN = NONE  
DOUG AND JAN = NONE  
AL = 2 4 8 10 14 16 20 22 26 28  
DOUG = NONE  
JAN = 9 27  
NORM = 1 3 5 7 11 13 15 17 19 21 23 25 29

3.5 The numbers 1 through 8 and a blank are randomly placed in a 3 X 3 array on the screen. For example:

```
5 8 3
  4 1
  6 2 7
```

Write a program so that when a key from 1 - 8 is pressed, the corresponding number on the screen slides horizontally or vertically to the adjacent blank location. If a key is pressed that is not horizontally or vertically adjacent to the blank, the computer does nothing. (For example: in the above diagram, only the 3, 1, and 7 can be moved.) The user can repeat this process of sliding numbers until the digit 9 is pressed, which causes the program to terminate. Example:

```
OUTPUT:  4  3  1
         2  8  5
         7  6
```

INPUT: (Press 2)

```
OUTPUT:  4  3  1
         8  5
         2  7  6
```

INPUT: (Press 8)

```
OUTPUT:  4  3  1
         8  5
         2  7  6
```

INPUT: (Press 6)

OUTPUT: (No change)

INPUT: (Press 9)

OUTPUT: (Program terminates)

**3.6** Write a program to simulate a chess game between two players. Display a chess board with the white chess pieces at the bottom of the board (designated by a leading W), and the black pieces at the top of the board (designated by a leading B). The board is set up according to the international chess notation: a letter of the alphabet designates vertical columns, and a number designates horizontal rows. The board must be displayed as below. The computer allows WHITE to move first, and then allows BLACK to move. The computer continues to alternate moves until a king is captured. The White king is designated as WK and the black king is designated as BK. The user enters a valid move in the format L#-L# where L is one of the letters (A, B, C, D, E, F, G, H) and # is one of the digits (1, 2, 3, 4, 5, 6, 7, 8). The piece occupying the coordinate of the first L# is moved to the coordinate of the second L#. If a piece is moved onto the opponent's square, the opponent's piece is removed. If a piece takes over an opponent's king, then print CHECK MATE, CCCCC WON (where CCCCC is either WHITE or BLACK). Only valid moves will be entered, and neither castling nor the "en passant" move will be done. No "check" warning is given in this game. Example:

```

OUTPUT: BR1 BK1 BB1 BQ  BK  BB2 BK2 BR2  !  8
        BP1 BP2 BP3 BP4 BP5 BP6 BP7 BP8  !  7
                                           !  6
                                           !  5
                                           !  4
                                           !  3
        WP1 WP2 WP3 WP4 WP5 WP6 WP7 WP8  !  2
        WR1 WK1 WB1 WQ  WK  WB2 WK2 WR2  !  1
-----
        A  B  C  D  E  F  G  H
    
```

INPUT: Enter white move: E2-E4

```

OUTPUT: BR1 BK1 BB1 BQ  BK  BB2 BK2 BR2  !  8
        BP1 BP2 BP3 BP4 BP5 BP6 BP7 BP8  !  7
                                           !  6
                                           !  5
                        WP5                !  4
                                           !  3
        WP1 WP2 WP3 WP4      WP6 WP7 WP8  !  2
        WR1 WK1 WB1 WQ  WK  WB2 WK2 WR2  !  1
-----
        A  B  C  D  E  F  G  H
    
```

INPUT: Enter black move: F7-F6

OUTPUT: (continued on next page)

(output continued)

```

OUTPUT: BR1 BK1 BB1 BQ BK BB2 BK2 BR2 ! 8
        BP1 BP2 BP3 BP4 BP5 BP7 BP8 ! 7
                BP6 ! 6
                ! 5
                WP5 ! 4
                ! 3
        WP1 WP2 WP3 WP4 WP6 WP7 WP8 ! 2
        WR1 WK1 WB1 WQ WK WB2 WK2 WR2 ! 1
-----
        A B C D E F G H

```

INPUT: Enter white move: D1-H5

```

OUTPUT: BR1 BK1 BB1 BQ BK BB2 BK2 BR2 ! 8
        BP1 BP2 BP3 BP4 BP5 BP7 BP8 ! 7
                BP6 ! 6
                        WQ ! 5
                WP5 ! 4
                ! 3
        WP1 WP2 WP3 WP4 WP6 WP7 WP8 ! 2
        WR1 WK1 WB1 WK WB2 WK2 WR2 ! 1
-----
        A B C D E F G H

```

INPUT: Enter black move: B8-C6

```

OUTPUT: BR1 BB1 BQ BK BB2 BK2 BR2 ! 8
        BP1 BP2 BP3 BP4 BP5 BP7 BP8 ! 7
                BK1 BP6 ! 6
                        WQ ! 5
                WP5 ! 4
                ! 3
        WP1 WP2 WP3 WP4 WP6 WP7 WP8 ! 2
        WR1 WK1 WB1 WK WB2 WK2 WR2 ! 1
-----
        A B C D E F G H

```

INPUT: Enter white move: H5-E8

```

OUTPUT: BR1 BB1 BQ WQ BB2 BK2 BR2 ! 8
        BP1 BP2 BP3 BP4 BP5 BP7 BP8 ! 7
                BK1 BP6 ! 6
                ! 5
                WP5 ! 4
                ! 3
        WP1 WP2 WP3 WP4 WP6 WP7 WP8 ! 2
        WR1 WK1 WB1 WK WB2 WK2 WR2 ! 1
-----
        A B C D E F G H

```

CHECK MATE, WHITE WON

**3.7** Write a program to determine the date of Easter and the date of Lent for a given year between 1970 and 2009 inclusive. Easter falls on the first Sunday following the arbitrary Paschal Full Moon, which does not necessarily coincide with a real or astronomical full moon. The Paschal Full Moon occurs on one of the following 19 dates corresponding to a key. The key is the remainder obtained by dividing the year by 19.

Key	Date	Key	Date
---	----	---	----
0	April 14	10	March 25
1	April 3	11	April 13
2	March 23	12	April 2
3	April 11	13	March 22
4	March 31	14	April 10
5	April 18	15	March 30
6	April 8	16	April 17
7	March 28	17	April 7
8	April 16	18	March 27
9	April 5		

Therefore, the key for the year 1990 is 14 (because  $1990 / 19 = 104$  remainder 14), and the Paschal Full Moon occurs on April 10. Since April 10, 1990 is a Tuesday, Easter Sunday is April 15. Note, if the Paschal Full Moon falls on a Sunday, Easter is the following Sunday. The earliest Easter can fall is March 23, and the latest is April 25. Lent begins on Ash Wednesday which comes 40 days before Easter, excluding Sundays (46 days including Sundays). Note, January 1, 1970 is a Thursday, and every year between 1970 and 2009 that is divisible by 4 is a leap year. Examples:

INPUT: Enter year: 1970  
 OUTPUT: **EASTER IS ON MARCH 29**  
**LENT IS ON FEBRUARY 11**

INPUT: Enter year: 1996  
 OUTPUT: **EASTER IS ON APRIL 7**  
**LENT IS ON FEBRUARY 21**

INPUT: Enter year: 2000  
 OUTPUT: **EASTER IS ON APRIL 23**  
**LENT IS ON MARCH 8**



3.8 Write a program to keep score for a bowler. Input will be 10 frames of numbers. Output will be the scoring of each frame, as shown below in the example. The standard method of scoring will be used in this program.

In each frame the bowler has at most two chances to roll down all 10 pins. If the bowler does not knock down all the pins in that frame, then his score is added to the number of pins that he previously knocked down. (Note: the bowler's score starts out as 0).

For the first 9 frames, if the bowler knocks all 10 pins down on the 2nd roll (indicated by a /) then his score for that frame is his previous score + 10 + the number of pins that he knocks down on his next roll in the next frame. If he rolls all the pins down on his 1st ball of a frame (indicated by an X) then he gets 10 + the total number of pins that he knocks down on the next 2 rolls. A new frame is started after 2 balls are rolled or all ten pins are knocked down on the 1st ball.

In the 10th frame, the bowler is allowed at most three rolls, depending upon his first and second rolls. If he gets all the pins down on the 1st roll then his score will be his previous score plus 10 plus the number of pins he knocks down on his next 2 rolls. If he knocks all 10 pins down after 2 rolls, his score will be his previous score + 10 + the number of pins he knocks down on his next roll. Frame inputs are right justified and the score is left justified in each frame. Example:

```
INPUT: Enter frame 1: 7/
      Enter frame 2: X
      Enter frame 3: 62
      Enter frame 4: X
      Enter frame 5: X
      Enter frame 6: 8/
      Enter frame 7: 63
      Enter frame 8: X
      Enter frame 9: X
      Enter frame 10: X9-
```

(Note: 62 indicates that 6 pins were knocked down on 1st roll and 2 pins were knocked down on 2nd roll;  
 / indicates all remaining pins knocked down on 2nd roll;  
 X indicates 10 pins knocked down on the 1st roll;  
 - indicates 0 pins knocked down.)

```
OUTPUT: -1- -2- -3- -4- -5- -6- -7- -8- -9- -10-
        ---!---!---!---!---!---!---!---!---!---!---!
        7/! X! 62! X! X! 8/! 63! X! X!X9-!
        20 !38 !46 !74 !94 !110!119!149!178!197!
        -----
```

**3.9** Write a program to solve a system of  $N$  equations with  $N$  unknowns. The program will first accept a value for  $N$  between 2 and 4 inclusive. For each equation the program should accept the values for the  $N$  coefficients followed by the constant. Each given system of equations will have one unique solution whose values will be integers. A possible algorithm to be used to solve an  $N \times N$  system of equations is shown to the right of the example input/output. Examples:

INPUT: Enter N: <b>3</b>	$2x + 4y + 6z = 4$ (Divide by 2)
Enter coefficients for row1	$2x + y + z = 3$
Col: <b>2</b>	$-x + 2y + z = 2$
Co2: <b>4</b>	
Co3: <b>6</b>	1 2 3 2
Enter constant: <b>4</b>	2 1 1 3 (Subtract 2*Row1)
Enter coefficients for row2	-1 2 1 2
Col: <b>2</b>	
Co2: <b>1</b>	1 2 3 2
Co3: <b>1</b>	0 -3 -5 -1
Enter constant: <b>3</b>	-1 2 1 2 (Subtract -1*Row1)
Enter coefficients for row3	
Col: <b>-1</b>	1 2 3 2
Co2: <b>2</b>	0 -3 -5 -1 (Divide by -3)
Co3: <b>1</b>	0 4 4 4
Enter constant: <b>2</b>	
OUTPUT: <b>(1, 2, -1)</b>	1 2 3 2
	0 1 5/3 1/3
	0 4 4 4 (Subtract 4*Row2)
	1 2 3 2
	0 1 5/3 1/3
	0 0 -8/3 8/3 (Divide by -8/3)
	1 2 3 2
	0 1 5/3 1/3 (Subtract 5/3*Row3)
	0 0 1 -1
	1 2 3 2 (Subtract 3*Row3)
	0 1 0 2
	0 0 1 -1
	1 2 0 5 (Subtract 2*Row2)
	0 1 0 2
	0 0 1 -1
	1 0 0 1
	0 1 0 2
	0 0 1 -1

