DKC^3 2018 – Computing Problems

1. Universal Product Code (UPC) (75 points)

The UPC is a barcode symbology that is widely used in the US and other for tracking trade items in stores.

UPC consists of numeric digits that are uniquely assigned to each trade item and is mainly used for the scanning of trade items at the point of sale (register/checkout). The UPC data structures are a component of GTINs (Global Trade Item Number) which follow specifications based on international standards.

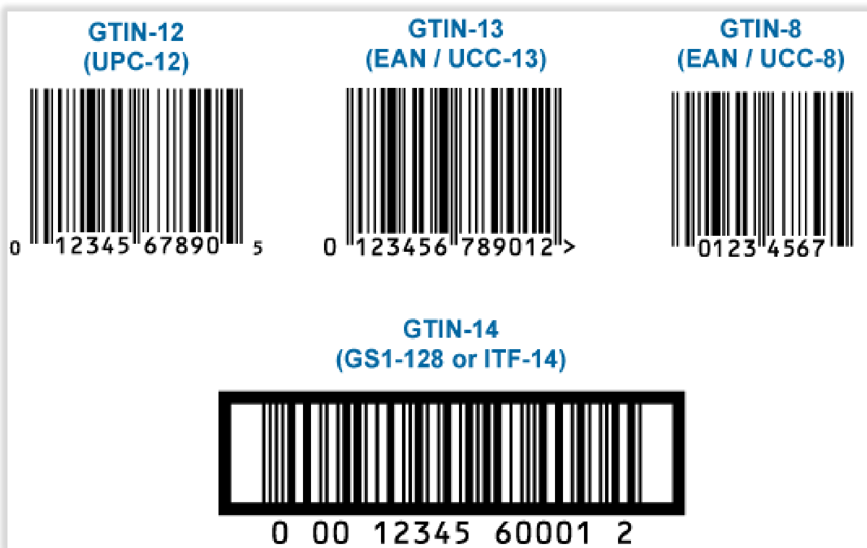


Figure A. GTIN Examples – Our focus is on GTIN-12 (UPC-A

There are multiple variations of a UPC, but we will concentrate on the common UPC-12 which consists of 12 digits and can be seen in the upper left corner of **Figure A** above. In most cases, the number stored in the UPC is displayed as part of the code which helps to identify the item if the barcode itself if obscured, but is not actually part of the UPC which is read by a scanning device.

Aside from that, the primary focus of a UPC is on the **BARS** (black) and **SPACES** (white). The bars/spaces range in thickness from 1 to 4 **modules**, and a **module** can be thought of in terms of a pixel or character, the bars/spaces range from 1 pixel wide up to 4 pixels wide and alternate bar-space-bar-space and space-bar-space-bar (more on that later)...

The next thing you will notice with the UPC-12 and similar formats is that some of the bars/spaces are longer than the rest. These are important as they help to determine the **S** (start), **M** (middle), and **E** (end) of the barcode. While they are different lengths than the rest of the barcode, they are also standardized in that the **S** (start) and **E** (end) are a group of 3 (BAR-SPACE-BAR) that are 3 modules thick, and the **M** (middle) which is a group of 5 (SPACE-BAR-SPACE-BAR-SPACE and is 5 modules thick.

DKC^3 2018 – Computing Problems

The first and last numerical coded group tend to be the same length as S/E/M, but they are indeed part of the numerical code which are made up the shorter bars. For the most part, the shorter bars/spaces make up the actual numeric code we are used to seeing. These bars/spaces are split in half around the **M** (middle), and in its simplest form the first half (L) denotes the Company who owns the specific product, and the second half (R) is the unique identifier for the product itself. A breakdown of the individual components can be found below in **Figure B**, followed by a more detailed description of each piece.

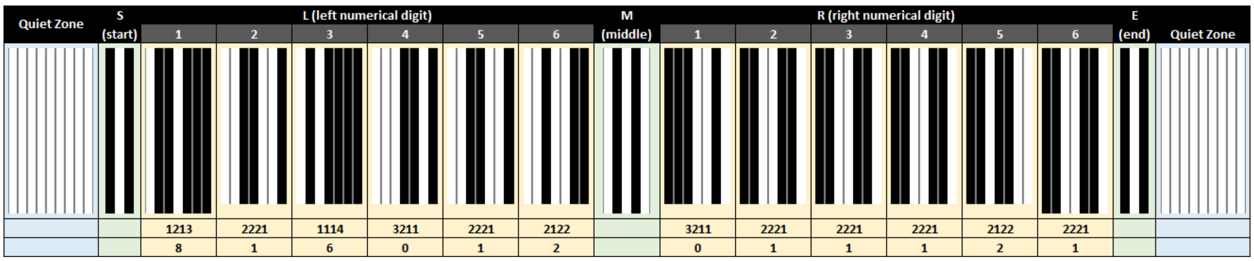


Figure B. UPC-12 Breakdown

- **Quiet Zone:** A buffer on either side of the UPC which consists of 9 modules of spaces each. **Not used in this problem.**
- **S (start):** Starting location of the barcode, broken into a group of 3 (BAR-SPACE-BAR) and consisting of 3 modules total.
- **L1 -> L6:** First 6 digits of the code, broken into groups of 4 (SPACE-BAR-SPACE-BAR), each group consisting of 7 modules for a total of 42 modules. L1 begins with the first SPACE after S. L6 ends with a BAR before M.
- **M (middle):** Middle of the barcode, broken into a group of 5 (SPACE-BAR-SPACE-BAR-SPACE) and consisting of 5 modules total.
- **R1 -> R6:** Last 6 digits of the code, broken into groups of 4 (BAR-SPACE-BAR-SPACE), each group consisting of 7 modules for a total of 42 modules. R1 begins with the first BAR after M. R6 ends with a SPACE before E.
- **E (end):** Ending location of the barcode, broken into a group of 3 (BAR-SPACE-BAR) and consisting of 3 modules total.

Overall Lengths: Minus the Quiet Zones, the barcode is 95 modules in length…

- S = 3
- L1 -> L6 = 7 each (7 * 6 = 42 total for L)
- M = 5
- R1 -> R6 = 7 each (7 * 6 = 42 total for R)
- E = 3

In order to decode a UPC, you must move from left to right and follow the breakdown in **Figure B** to break the UPC into usable segments and find the coded groups (L/R) to help create the individual numeric codes (L1 -> L6 / R1 -> R6) which are then mapped to digits.

The codes are created by counting the number of modules in the alternating bar/spaces of each grouping. For instance, **L1** from **Figure B** can be read as 1-SPACE, 2-BAR, 1-SPACE, 3-BAR. That gives us the code 1213 which maps to the number 8 from the chart in **Figure C** below. Following the same logic for each group will reveal the full list of numbers that make up the 12 digit UPC.

L1: 1-SPACE, 2-BAR, 1-SPACE, 3-BAR (1213 = 8)
L2: 2-SPACE, 2-BAR, 2-SPACE, 1-BAR (2221 = 1)
L3: 1-SPACE, 1-BAR, 1-SPACE, 4-BAR (1114 = 6)
L4: 3-SPACE, 2-BAR, 1-SPACE, 1-BAR (3211 = 0)
L5: 2-SPACE, 2-BAR, 2-SPACE, 1-BAR (2221 = 1)
L6: 2-SPACE, 1-BAR, 2-SPACE, 2-BAR (2122 = 2)
R1: 3-BAR, 2-SPACE, 1-BAR, 1-SPACE (3211 = 0)
R2: 2-BAR, 2-SPACE, 2-BAR, 1-SPACE (2221 = 1)
R3: 2-BAR, 2-SPACE, 2-BAR, 1-SPACE (2221 = 1)
R4: 2-BAR, 2-SPACE, 2-BAR, 1-SPACE (2221 = 1)
R5: 2-BAR, 1-SPACE, 2-BAR, 2-SPACE (2122 = 2)
R6: 2-BAR, 2-SPACE, 2-BAR, 1-SPACE (2221 = 1)

CODE	DIGIT
3211	0
2221	1
2122	2
1411	3
1132	4
1231	5
1114	6
1312	7
1213	8
3112	9

Figure C. Code to Digit Chart

Input
There will be 10 test cases for this problem with each test case consisting of 5 lines, each containing characters to make up a UPC. These characters will be a SPACE (“ “, char 0x00FF) and HASH (#, char 0x0023), each of these characters will represent a module and alternate in groups to make up the full UPC. Each test case will be separated by a line with a single ASTERISK (*, char 0x002A).

Output
The output will consist of a single line per test case that represents the 12 digit UPC code.

816012011121

DKC^3 2018 – Computing Problems

Input File: C:\DKC3\UPCIn.txt
Output File: C:\DKC3\UPCOut.txt

Examples:

Input:

```
# # ## ###  ##  # # ####   ## #  ##  #  #  ## # # ###  # ##  ## ##  ## ##  ## ## ##  ##  ## # #
# # ## ###  ##  # # ####   ## #  ##  #  #  ## # # ###  # ##  ## ##  ## ##  ## ## ##  ##  ## # #
# # ## ###  ##  # # ####   ## #  ##  #  #  ## # # ###  # ##  ## ##  ## ##  ## ## ##  ##  ## # #
# # ## ###                                    # #                                    ##  ## # #
# # ## ###                                    # #                                    ##  ## # #
*
# #   ## # #   ##   # ##   ## #   ## #   ## # # # ###  # #  ### # ###  #    # # #    ##  ## # #
# #   ## # #   ##   # ##   ## #   ## #   ## # # # ###  # #  ### # ###  #    # # #    ##  ## # #
# #   ## # #   ##   # ##   ## #   ## #   ## # # # ###  # #  ### # ###  #    # # #    ##  ## # #
# #   ## #                                  # # #                                    ##  ## # #
# #   ## #                                  # # #                                    ##  ## # #
```

Output:

816012011121
049000054361

DKC^3 2018 – Computing Problems

2. Cribbage Scoring (75 points)

Cribbage is a game played with a standard deck of 52 cards. Each player's hand consists of 4 cards and an additional face up card. There are several phases or rounds to playing cribbage: deal, discard, play and show. This challenge is specific to the **show** phase of gameplay only. During the show round, each player scores points based on the content in their hand (plus the face up card). Points are awarded for the following:

- Any number of cards that add up to 15 (regardless of suit) – 2 points
    - Face cards are valued as ten for this purpose.
- Runs of three, four, or five cards – 3, 4 and 5 points respectively
    - Runs are considered by card rank: Ace (1) through 10; then Jack, Queen, King
    - Runs cannot "wrap around" – Q-K-A is not a run.
    - Multiple runs are possible: a hand containing 2-2-3-4 is two runs of three cards.
    - You must take the maximum possible run. A hand containing 2-3-4-5-6 is a **run of five** and cannot be downgraded to multiple runs of four and three.
- Two, three, or four of a kind – 2, 6 and 12 points respectively
- Flushes: 4 or 5 cards of the same suit – 4 and 5 points respectively
    - The additional face up card is not counted for a 4-card flush
- Nobs: A Jack of the same suit as the additional face up card – 1 point

Note: cards can be used more than once for each combo. A five, a queen, a jack, and a king, all of spades, count as three fifteens (5 and Q, 5 and J, 5 and K) and a four card flush.

Input
Your program should take an array of 5 cards, each separated by a comma. Each card will be designated by a rank: Ace through 10 or Jack (11) – Queen (12) – King (13) as well as a suit: Hearts, Clubs, Spades and Diamonds. The first 4 cards are the cards in your hand and the final card is the additional face up card.

Output
Your program should output the total score of each hand.

Input File: C:\DKC3\CribbageIn.txt
Output File: C:\DKC3\CribbageOut.txt

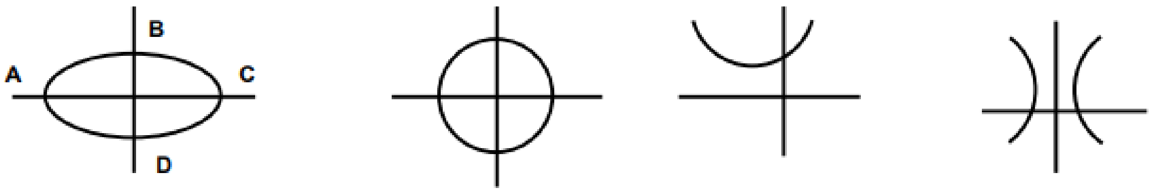
Examples:

Input:
5D,QS,JC,KH,AC
8C,AD,10C,6H,7S

Output:
10
7

DKC^3 2018 – Computing Problems

3. Hairy Conic Junior (75 points)



ELLIPSE CIRCLE PARABOLA HYPERBOLA

$$\mathbf{Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0}$$

Circle	A = C and B = 0
Ellipse	A ≠ C but have the same sign. B = 0
Parabola	A or C = 0. B=0
Hyperbola	A and C have different signs. B=0

All of the above figures are related mathematically because the equations used to describe each figure are similar. The *general* form of the equation for each is $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$. They differ because of the restrictions placed on the coefficients in the table above. For the circle A and C must be equal and B must equal 0. For the ellipse, A and C cannot be equal but must have the same sign. B must equal 0. An example of an equation of a circle is: $x^2 + y^2 + 4x - 6y - 3 = 0$. An example of an equation of an ellipse is: $x^2 + 4y^2 - 6x - 16y - 11 = 0$. In this program you will be given the general form of the equation and must determine if it is a circle, an ellipse, a hyperbola or a parabola. Further, you will have to give certain other information about the figure.

The general equation above is not very useful for determining information about the position of the figure on the coordinate axis system. The equation must be modified. The method commonly used is called "completing the square". Using the equation of the circle above, the result is as follows:

$$x^2 + 4x + 4 + y^2 - 6y + 9 = 3 + 4 + 9$$

$$x^2 + 4x + 4 + y^2 - 6y + 9 = 16$$

$$(x + 2)^2 + (y - 3)^2 = 16$$

In this form, it is easy to find the center of the circle and its radius. The center is at (-2, 3) and the radius is 4 (the principal square root of 16).

Finding information about the ellipse is done in a similar manner. However, since the "completing the square" method only works when the coefficient of the squared term is a positive one, the second trinomial must be modified.

$$x^2 + 4y^2 - 6x - 16y - 11 = 0$$
$$x^2 - 6x + 9 + 4(y^2 - 4y + 4) = 11 + 9 + 16$$
$$(x-3)^2 + 4(y-2)^2 = 36$$

In this form, the center of the ellipse can be found to be at (3, 2).

If both sides of the equation are divided by the value on the right side, the equation is now said to be in *standard* form:

$$\frac{(x-3)^2}{36} + \frac{(y-2)^2}{9} = 1$$

In the figure of the ellipse, above, line segments AC and BD are axes. Since segment AC is longer it is called the major axis and since BD is shorter it is called the minor axis. The bigger denominator above determines the major axis. Its length is found by taking two times the principal square root of the denominator. Here the major axis will have a length of 12 and in a similar manner, the minor axis will have a length of 6.

Using the procedures above, examples of equations of a hyperbola in standard form are as follows:

$$\frac{(x-3)^2}{1} - \frac{(y-2)^2}{3} = 1 \qquad\qquad \frac{(y-2)^2}{1} - \frac{(x-3)^2}{3} = 1$$

The center of both hyperbolas is (3, 2). Since the general equation that produced the standard form on the left had a positive A and a negative C, the principal axis, the line that connects the foci and passes through the center, is the line y = 2. The standard form on the right had a negative A and a positive C and has a principal axis of x = 3.

Again, using the procedures above, examples of equations of a parabola in standard form are as follows:

$$(x-3)^2 = 12(y-1) \qquad\qquad (y-1)^2 = 12(x-3)$$

The vertex of both parabolas is (3, 1). Since the general form of the equation for the standard form on the left had C= 0, the axis of symmetry, is x = 3. The general form of the equation for the standard form on the right had A = 0 and its axis of symmetry is y = 1.

Input
There will be 10 test cases, each on a new line. Each test case will consist of a string representing an equation in general form. The ^ will be used to denote exponents.

Output
For each test case, print the type of figure the equation produces. If it is a circle, print the location of its center in ordered pair format (x,y) and its radius. If it is an ellipse, print the location of its center and the

length of its major axis. If it is a hyperbola, print its center and the equation of the principal axis. If it is a parabola, print its vertex and the equation of the axis of symmetry.

Input File: C:\DKC3\HairyConicIn.txt
Output File: C:\DKC3\HairyConicOut.txt

Examples:

Input:
x^2+y^2+4x-6y-3=0
x^2+4y^2-6x-16y-11=0
4x^2-y=0
9x^2-4y^2+18x+24y-63=0

Output:
Circle, (-2, 3), 4
Ellipse, (3,2), 12
Parabola, (0,0), x=0
Hyperbola, (-1, 3), y=3

4. Prize Attendance (75 points)

A particular high school started a "Prize Attendance" program offering cash rewards to students with good attendance and punctuality during the week of final exams (which lasts Monday through Thursday) to cut down on students trying to take their exams early and have an extra week off. During this week, if they are absent for two consecutive days or late on more than one occasion then they forfeit their prize.

During an N-day period a trinary string is formed for each child consisting of L's (late), O's (on time), and A's (absent).

For example, there are 81 trinary strings for a 4-day period that can be formed, exactly 30 of those strings would lead to a prize:

ALAO, ALOA, ALOO, AOAL, AOAO,
AOLA, AOLO, AOOA, AOOL, AOOO,
LAOA, LAOO, LOAO, LOOA, LOOO,
OALA, OALO, OAOA, OAOL, OAOO,
OLAO, OLOA, OLOO, OOAL, OOAO,
OOLA, OOLO, OOOA, OOOL, OOOO

The "Prize Attendance" program has been running in the high school for about 4 years and doing really well so the school superintendent wants to look into offering the cash reward more often throughout the school year and for potentially longer periods of time than 4 days. However, they want to have significant research done before implementation so as not to take away from other programs or go bankrupt altogether while trying to persuade the students to stay in school.

Input
First, they need to know how many possible matches there are for different periods of days. You will be supplied with a single number representing the period of days the school wants to potentially offer the cash prize for.

Output
The school needs to know the full number of string possibilities and the combinations of the strings that would lead to a prize. The superintendent asks that you have the strings in alphabetical order, with a comma and space between each string, and a max of five strings per line. Leave a line with an asterisk between test case answers for easy readability. If, for some reason, there are no combinations that would lead to a prize, leave a blank line to make it obvious to the superintendent that this period of days would be a waste of time.

Input File: C:\DKC3\PrizeIn.txt
Output File: C:\DKC3\PrizeOut.txt

Examples:

Input:
4
3

Output:
81
ALAO, ALOA, ALOO, AOAL, AOAO,
AOLA, AOLO, AOOA, AOOL, AOOO,
LAOA, LAOO, LOAO, LOOA, LOOO,
OALA, OALO, OAOA, OAOL, OAOO,
OLAO, OLOA, OLOO, OOAL, OOAO,
OOLA, OOLO, OOOA, OOOL, OOOO
*
27
ALA, ALO, AOA, AOL, AOO,
LAO, LOA, LOO, OAL, OAO,
OLA, OLO, OOA, OOL, OOO
*