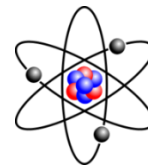




CCS Programming Contest

2022

COLONIAL CHRISTIAN SCHOOL



Super Easy Level 0 Problems

- 0.1. **Baseball:** A baseball pitcher's "Earned Run Average" is computed using the formula:

$$ERA = \frac{R \times 9}{I}$$

Where R = the number of earned runs allowed and I = the number of innings pitched. Given R and I , compute and print a pitcher's ERA . Assume a whole number of innings is pitched.

- 0.2 **Cubes:** Make a table of square roots and cube roots for all the numbers from 1 to 100. Have the table print on the printer (using LPRINT) in three columns.

Output:

Number	Root	Square Root	Cube Root
1		1	1
2		1.414214	1.259921
3		1.732051	1.4425
4		2	1.587401

- 0.3 **Loan:** The value of a loan is calculated by the following formula:

$$V = P \times (1 + I)^N$$

Input a value for P , I , and N , where P is the principle, I is the interest rate and N is the number of months of the loan.

- 0.4 **Belt:** The length, L , of a belt around two wheels can be computed by:

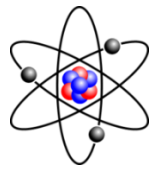
$$L = 3.26 (R_1 + R_2) + 2 \times D$$

Where R_1 and R_2 are the radii of the wheels and D is the distance between the centers of the wheels. Given R_1 , R_2 and D , find L .

Grading Key

Level	Points Received
0	1
1	25
2	100
3	225
4	400
5	625
6	1000

COLONIAL CHRISTIAN SCHOOL



0.5 **Pendulum:** The period T (in seconds) of a simple pendulum of length L (in meters) is given by the formula:

$$T = 2 \times \Pi \times \sqrt{\frac{L}{g}}$$

Find T given:

$$\Pi = 3.14$$

$$g = 9.8 \quad \leftarrow \text{(The gravity constant for planet Earth)}$$

L = The length of the pendulum (typed in.)

0.6 **Kepler:** According to Kepler's law of planetary motion, if P is the number of Earth years for one revolution of the Sun, and D is the planet's average distance (in kilometers) from the Sun, then the following formula applies:

$$\frac{P^2}{D^3} = 2.985 \times 10^{-25}$$

Given, D calculate P .

0.7 **Circular:** Given input of the radius, find and print the area and perimeter of a circle and find the surface area and volume of a sphere made from that circle. Use the following formulae:

Circle:

$$\text{Circumference} = 2 \times \Pi \times R$$

$$\text{Area} = \Pi \times R^2$$

Sphere:

$$\text{Surface} = 4 \times \Pi \times R^2$$

$$\text{Volume} = \frac{4}{3} \times \Pi \times R^3$$

Where $\Pi = 3.1416$ and R is entered from the keyboard.

0.8 **Picture:** Write a program which creates the following words on the printer followed by your own name written in the same style lettering.

```
# # #      ##### # # ##### ### #####
# # #      # # # # # # # # #
##### #   # ##### ## ### ##
# # #      # # # # # # # # #
# # #      # # # ##### # ## #####
```

0.9 **Library:** Make a program to calculate Library fines. Enter the number of days late from the keyboard. Calculate the late fine as follows: 50¢ for the first day and 25¢ for every day after that.

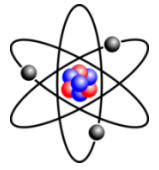
0.10 **Sum:** Find the sum of all the numbers counting up to any number specified. For example, if the user types in the number 5, the computer will add 1 + 2 + 3 + 4 + 5, and print the answer of 15 on the screen.

0.11 **Blastoff:** Make the computer count backwards from 100 to zero printing each number except the number zero. In the spot that zero would have been printed, print the word **BLASTOFF**.

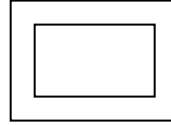
0.12 **Es:** Fill the entire screen with letter **Es**.

0.13 **Criss Cross:** Criss cross the screen with letter **Xs**. The diagonals must go completely across the screen from one corner to the other

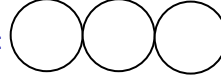
COLONIAL CHRISTIAN SCHOOL



0.14 **Boxes:** Use graphics to draw the following:



0.15 **Circles:** Use graphics to draw the following:

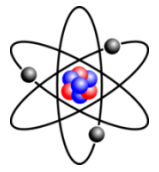


0.16 **E:** Use graphics to draw a large letter E.

0.17 **Tax:** Write a program that calculates sales tax and adds it to any entered purchase price. The sales tax is 7%.

0.18 **Button:** Make a red button that changes green when it is clicked.

0.19 **Label:** Make a program that has three labels. One label has the caption "Name"; another has the caption "Address". When ever the mouse pointer moves across the name label, your name must appear in the third label, and when ever the mouse pointer moves across the address label, your address must appear in the third label.



Easy Level 1 Problems

- 1.1 **Capitals:** Write a program to accept a sentence in upper and lower case letters. Have the computer calculate the number of capital letters and the number of times each occurs. Print the letters in alphabetical order with the number of times each occurs.

SAMPLE:

Input: I went to New York City and saw the Statue of Liberty and the Empire State Building.

Output: 9 capital letters

1 B
1 C
1 E
1 I
1 L
1 N
2 S
1 Y

- 1.2 **Sum 1/n:** The n th of a sequence is defined to be $1/n$. Find the sum of the first 100 terms, rounded to the nearest thousandth. (You may wish to test your program with the first 10 terms rounded to the nearest thousandth. That solution is 2.929.)

- 1.3 **Roots:** Write a program that accepts the input of two positive integers less than 10000. If both are perfect squares, print **BOTH** and print the square root of each. If only one is a perfect square, print **ONE** and its square root. If neither is a perfect square, print **NEITHER**.

SAMPLE:

Input: 25, 49
Output: BOTH 5, 7

Input: 9, 23
Output: ONE 3

Input: 13, 17
Output: NEITHER

- 1.4 **Standard Time:** Given time input as military time (24 hour time), convert the time to standard time (12 hour time) and label correctly as a.m. or p.m. All input times will be in the form HH:MM. Times such as 02:45 must be shown as 2:45 a.m. not 02:45 a.m.

SAMPLE:

Input: 13:45
Output: 1:45 p.m.

Input: 08:45
Output: 8:45 a.m.

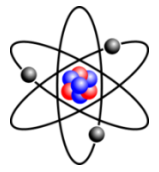
- 1.5 **Cubes:** Write a program that accepts the input of two integers X and Y such that $-1000 < X < Y < 1000$. Print the perfect cubes between X and Y , exclusive of X and Y .

SAMPLE:

Input: -30, 10
Output: -27, -8, -1, 0, 1, 8

- 1.6 **Sort Digits:** Write a program that accepts the input of a positive integer of any length less than 30 digits. Have the computer print out the digits in the number in descending order.

COLONIAL CHRISTIAN SCHOOL



SAMPLE:

Input: 239745014
Output: 975443210

- 1.7 **No Vowels:** Write a program that accepts the input of a sentence using both upper and lower case letters. Have the computer print the sentence with all vowels replaced with an "*". Vowels are defined as **a, A, e, E, i, I, o, O, u, U**.

SAMPLE:

Input: The United States of America
Output: Th* *n*t*d St*t*s *f *m*r*c*

- 1.8 **Ugly:** Write a program that asks for any number of names and then creates sentences which saying each person on the list is ugly. The final sentence should start with the words, "*And finally*" and have the word "*very*" before the word ugly.

SAMPLE:

Input: How many names->**3**

Name #1 ->Sam
Name #2 ->Phil
Name #3 ->Steve

Output: Sam is ugly.
Phil is ugly.
And finally Steve is very ugly.

- 1.9 **Backwards:** Write a program that asks for a sentence and prints it backwards word by word. Handle capitalization and the period at the end.

SAMPLE:

Input: Sam eats spam.
Output: Spam eats sam.

- 1.10 **Not /3:** Print the numbers from one to 500 skipping every number divisible by 3. Print the result so that it can be viewed on the screen all at one time.

- 1.11 **Adding Machine:** Write a program that keeps a running total of numbers typed in by the user. Have the program print and reset the total whenever the user presses the ENTER key without typing a number in.

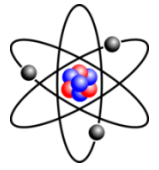
SAMPLE:

```
->400
->300
->400
-><ENTER>
1100
->29
->13          (and so on)
```

- 1.12 **States:** Write a program to list the fifty states grouped by letter. Read all fifty states from data statements and output them as shown below:

ALABAMA
ALASKA

COLONIAL CHRISTIAN SCHOOL



ARIZONA
ARKANSAS

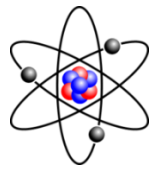
CALIFORNIA
COLORADO
CONNECTICUT

DELAWARE

FLORIDA

.
. .
.

- 1.13 **Same Dice:** Four dice are rolled. Accept the four numbers which appear on the top faces of the dice and decide whether 0, 1, 2, 3 or 4 of the other dice show the same value.
- 1.14 **Color Mix:** Write a program that raises or lowers the amount of red, green or blue light in a graphical object and displays the intensity of the three colors.
- 1.15 **Sum:** Find the sum of all numbers 1 to N, where N is any integer (no decimals).
SAMPLE: N = -3 Yields -5
- 1.16 **Star:** Write a program that makes a perfect five pointed star using graphics.



Reasonably Easy Level 2 Problems

- 2.1 **Plus Minus:** Write a program that accepts the input of an addition problem as a string of less than 30 characters, similar to $1 + 2 + 3 + 4 =$. Have the computer first perform the addition and print the results, then have the computer reverse the order of the numbers and place minus signs between each two numbers. The computer will then calculate the answer to the resulting problem and print it on the screen.

SAMPLE:

Input: $5 + 7 + 11 + 9 =$
 Output: 32 <-- (5+7+11+9=32)
 -14 <-- (9-11-7-5=-14)

- 2.2 **Word Wrap:** Write a program that accepts the input of a sentence. The sentence will have more than 10 letters and less than 125 letters. Have the computer left justify the sentence in 30 column format on the screen with appropriate word wraps. Do not split any word in the middle.

Input: This is a test of the newest, amazing word processing program written by Westminster Christian School students.

Output: This is a test of the newest,
 amazing word processing
 program written by Westminster
 Christian School students.

- 2.3 **Convert Base:** Write a program that has a menu with the following choices:

1. Convert to Binary to Decimal
2. Convert Decimal to Binary
3. Quit

The program will then accept input and perform each conversion. Return to the menu after each conversion. be sure to check the validity of the data entered.

SAMPLE:

Input: 1 (menu option)
 Enter binary number-> **10102**
 Output: Invalid entry

Input: 1 (menu option)
 Enter binary number-> **1010**
 Output: 10

- 2.4 **Roman Numeral:** Write a program that will accept the input of an Arabic number or a Roman Numeral N, such that $1 < N < 51$. Convert the number to the other system. All data will be valid and in capital letters or Arabic numbers.

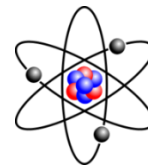
SAMPLE:

Input: **VI**
 Output: 6
 Input: **39**
 Output : XXXIX

- 2.5 **Insert:** Write a program to read the following sorted data into an array:

DATA 3,11,22,44

COLONIAL CHRISTIAN SCHOOL



2.10 **MAX Function:** Write a utility function for Q Basic, **MAX** which takes as input an array and returns the largest element in the array.

2.11 **The NO Program:** Write a program that will determine the frequency of the word "NO" in a group of strings entered that ends with the word end.

Sample:

```
THERE IS NOTHING SO STIMULATING AS NEW KNOWLEDGE.  
NO MAN'S KNOWLEDGE CAN GO BEYOND THAT WHICH IS KNOWABLE.  
THE ANSWER IS NO!  
SAY NO TO BRAIN DAMAGE.  
END
```

THE WORD NO OCCURS 3 TIMES.

2.12 **Date Conversion:** Write a program which converts the date from standard numeric form to a fully written date. For example: The input 03/17/95 would cause the computer to print MARCH 17, 1995.

2.13 **Permutations:** Write a program to print out all permutations of the digits from 1 to N , where N is inputted from the keyboard. Thus for $N=3$ the output would be:

```
123  
132  
213  
231  
312  
321
```

For $N=3$ there are $3!$ possibilities or $1 \times 2 \times 3$ possibilities. For four there would be $4!$ (that is 24) possibilities. The computer would print out the following list in what ever format the programmer desires:

```
1234  2134  3124  4123  
1243  2143  3142  4132  
1324  2314  3214  4213  
1342  2341  3241  4231  
1432  2413  3412  4312  
1423  2431  3421  4321
```

2.14 **Statistical Median:** The median in statistics is the middle number in a list of numbers arranged in numeric order. For example the list 1,3,13,14,200: the median is the number 13 because there are an equal number larger and smaller numbers in the list. If there are an even number of elements in the list, the two center numbers are averaged to find the median. One last thing, the numbers in the list are not typed in in sorted order. They come in in random order and the program must handle up to 500 numbers in a list.

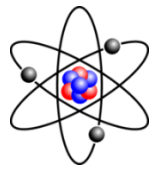
Example:


For the list 2000,37,700,7500,3,10,7503,1405; the median is 1052.5

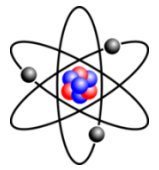
2.15 **Largest, Smallest:** Given five real numbers, print the largest and the smallest. Do not assume that the numbers are inputted in any specific order.

2.16 **Sum of Cubes:** The number 153 has an interesting property. It is equal to the sum of the cubes of its digits, that is, $153 = 1^3 + 5^3 + 3^3$. There are no two-digit or four digit whole numbers with this property and only four three-digit numbers (including 153). Find these four whole numbers.

COLONIAL CHRISTIAN SCHOOL



- 2.17 **Sum of Squares:** Find every two-digit whole number that equals the sum of the square of its digits.
- 2.18 **Sum of Fourth Powers:** Find every four-digit whole number that equals the sum of the fourth powers of its digits.
- 2.19 **Divisible by Product 3:** Find all three-digit whole numbers that are divisible by the product of their digits. There are twenty such numbers. Note: Any number with a zero digit must be excluded to avoid a zero divisor. For example, the number 315 is divisible by 15, the product of $3 \times 1 \times 5$.
- 2.20 **Divisible by Product 4:** Find the forty four digit whole numbers that are divisible by the product of their digits. (See the example in the previous problem.)
- 2.21 **Shapes:** Write a program that asks for the number of sides in a polygon and then draws a perfect polygon with that many sides.
- 2.22 **Any Dice:** There are many types of dice that have a different number of sides than the familiar six sided cubic dice we use for so many of our games in this country. Write a program that asks the user how many sides his dice have and then how many dice to roll. The program will then simulate the rolls of the dice and display the total number of times each side comes up.
- 2.23 **Drawing:** Make a program that creates the shape to the right using graphics:  Be sure to include the shading and the curved lines in the lower part of the box.
- 2.24 **Globe:** Create a program that generates a globe complete with longitude and latitude lines.
- 2.25 **Animation:** Create a car drawn in graphics that moves across the screen.



Moderate Difficulty Level 3 Problems

- 3.1 **Alphabetical:** Write a program to accept the input of a sentence in all capital letters with no punctuation. Print the longest sequence of letters in alphabetical order (disregard spaces). If there are two or more longest sequences of the same length, print all of them. The sequence ABBDE is considered to be in alphabetical order.

SAMPLE:

Input: IT IS ABOUT TIME TO DO SOME REALLY HEAVY PROGRAMMING

Output: ABOU

Input: A HOUSE IS NOT A HOME

Output: AHOU

- 3.2 **Sorted:** Write a program that accepts the input of names (first last) and city until -999 is entered as input for a name. Have the computer then print the names and corresponding cities in a list sorted alphabetically by last name. Then have them printed in alphabetic order by city, with last name as a secondary sort.

SAMPLE:

Input: Name->Bruce Raskin
City->Pembroke Pines
Name->Rita Oates
City->Coral Gables
Name->Molly Weeks
City->North Miami
Name->Bill Fleischman
City->Pembroke Pines
Name->-999

Output: Bill Fleischman Pembroke Pines
Rita Oates Coral Gables
Bruce Raskin Pembroke Pines
Molly Weeks North Miami

Rita Oates Coral Gables
Molly Weeks North Miami
Bill Fleischman Pembroke Pines
Bruce Raskin Pembroke Pines

- 3.3 **Base Conversion:** Write a program that accepts the input of a number (base 2 to 16) followed by its base, followed by the base to which the number is to be converted (base 2 to 16). Have the program make the conversion. All data will be valid.

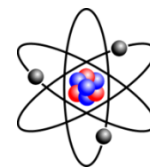
SAMPLE:

Input: 16,7,2
Output: 1101
Input: 11011,2,8
Output: 33

Input: A2,16,10
Output: 162

- 3.4 **Two Dice:** Simulate the toss of two dice, 1000 times. Show the throws of each die in columns on the screen. After all throws have been completed, show the number of times each sum (2 to 12) was obtained and the percent of each.

COLONIAL CHRISTIAN SCHOOL



SAMPLE:

2,1 3,4 6,3 4,3 1,1 4,5
 1,5 4,3 2,5 6,3 etc.

Sum	Number of times	Percent
2	26	2.6
3	55	5.5
4	82	8.2
5	109	10.9
6	140	14.0
7	167	16.7
8	138	13.8
9	111	11.1
10	84	8.4
11	58	5.8
12	30	3.0
Totals	1000	100.0%

- 3.5 **SORT Procedure:** Write a utility function for Q Basic, **SORT** which takes as input an array and sorts every element in the array.
- 3.6 **QUEUEIN and QUEUEOUT Functions:** Write two utility functions for Q Basic, **QUEUEIN** and **QUEUEOUT**, which manage a queue. A Queue is like a lunch line. Numbers or strings are entered one at a time and they exit on a first come first serve basis. **QUEUEIN** is a procedure which enters a number or string. **QUEUEOUT** is a function which returns the number or string which is currently first in line. Be sure to handle situations like **QUEUEOUT** commands used on an empty queue and queue overflow.
- 3.7 **STACKIN and STACKOUT Functions:** Write two utility functions for Q Basic, **STACKIN** and **STACKOUT**, which manage a queue. A Stack works like a stack of dishes in the cupboard. Unlike a Queue or a line, the last dish placed on the stack is the first to be used and the first dish placed in the stack will be the last to be used. **STACKIN** and **STACKOUT** work the same way that **QUEUEIN** and **QUEUEOUT** work in the previous problem. Be sure to handle situations like **STACKOUT** commands used on an empty **STACK** and **STACK** overflow.
- 3.8 **Money Changer:** Write a program to calculate the number of different ways a dollar bill can be broken into change. (eg. 1x50, 1x25, 25x1 is one way; 2x25, 2x10, 6x5 is another) Print the answer
- 3.9 **Wallis' Method:** In 1650 the English Mathematician John Wallis discovered the following formula:

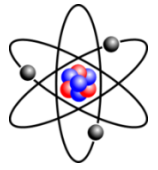
$$\frac{B}{2} = \frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \frac{8}{7} \dots$$

Using this formula, Approximate B. Continue multiplying factors until the difference between successive products is less than .00005.

- 3.10 **Leibniz's Method:** In 1674, Leibniz found the following formula:

$$\frac{B}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} \dots$$

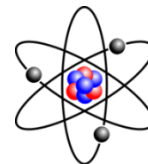
COLONIAL CHRISTIAN SCHOOL



Use this formula to approximate B. continue Adding terms until successive sums differ by less than .00005.

- 3.11 **Clock:** Simulate the movement of the second hand of a clock using graphics. (Be sure to make the hand move in increments every second.)

COLONIAL CHRISTIAN SCHOOL



Difficult Level 4 Problems

4.1 **Metric Conversion Program:** Write a program to produce English-Metric, Metric-English conversion tables. Include the following as possible table choices:

Inches		Millimeters
Feet		Meters
Yards	TO	Kilometers
Miles		
Light Years		

Teaspoons		Cubic Centimeters
Tablespoons		Milliliters
Cups	To	Liters
Pints		
Quarts		
Gallons		
Bushels		
Pecks		
Ounces		

Ounces		Grams
Pounds		Kilograms
Tons	To	Metric Tons
Imperial Tons		

4.2 **Check Writer:** Modern corporations seldom do payroll checks by hand. Computers are used to print out the checks on special paper. The checks are then separated by a buster and signed. Write a program to print checks to pay every student in the class what ever you deem their work to be worth.

- Include on each check
1. Today's date
 2. Payee's name
 3. Numeric check amount
 4. Amount written using words

Example:
 For DATA statement:
 DATA D. J. BOUWSMA,1230
 The following check would be generated:



Westminster Christian School
 6855 S.W. 152 St.
 Miami, FL 33157
 (305) 233 - 2030

JUNE 20 1999

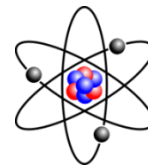
Pay to the
 Order of D. J. BOUWSMA \$ 1,230.00

ONE THOUSAND TWO HUNDRED THIRTY AND 00/100 Dollars

First Union Bank
 Of Florida
 Miami, FL 33150

FOR _____

COLONIAL CHRISTIAN SCHOOL



- 4.3 **Big Math Add:**
4.4 **Big Math Subtract:**
4.5 **Big Math Multiply:** Nobody likes Exponential Notation. (At least, nobody who really matters.) Write a program that will do math with 15, 30, 100 digit numbers - what ever the person asks for. Do addition, subtraction, multiplication and division.

Example:

```
HOW MANY DIGITS MAXIMUM? 32
```

```
ENTER YOUR FIRST NUMBER:
```

```
1>123456789012345678901234567890
```

```
ENTER YOUR SECOND NUMBER:
```

```
2>555555555555555555555555555555
```

```
ENTER OPERATION:
```

```
(+ - * /) > +
```

```
  523456789012345678901234567890
+  555555555555555555555555555555
-----
1079012344567901234456790123445
```

Output for Addition and Subtraction and Multiplication should be printed out in the form shown above. Division should show the entered numbers above and below a line indicating division. An equal sign should be written below it with the answer following.

- 4.6 **Cryptological Ciphers:** One of the earliest usages of computer equipment was code breaking. Messages sent across radio waves can be intercepted by just about anybody. For that reason, most military messages are sent in code. Write a program which can encode messages and decode messages. Our simple program will work by shifting the letters of the message over a set number of positions. Example: The input **ABC** becomes **DEF** when it is shifted over one letter. Be careful with the end of the alphabet. In our example, the letter **Z** becomes the letter **A**.

Be sure that your program does the following:

1. Encode and decode upper and lower case letters separately.
2. Encode and decode numbers and punctuation separately.
3. Allow entry of the offset.

Example:

```
ENTER OFFSET:12
```

```
ENCODE OR DECODE (E/D)?E
```

```
ENTER TEXT BELOW, PRESS RETURN TWICE WHEN DONE:
```

```
>Cats, Dogs
```

```
>
```

```
Omfe8 Pase
```

```
ENTER OFFSET:12
```

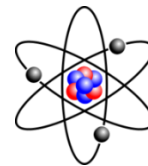
```
ENCODE OR DECODE (E/D)?D
```

```
ENTER TEXT BELOW, PRESS RETURN TWICE WHEN DONE:
```

```
>Omfe8 Pase
```

```
>
```

```
Cats, Dogs
```

Nasty Level 5 Problems

5.1 **People Ratings:** Write a program to determine who is the best human being in the class. Since beauty is in the eye of the beholder, poll everyone in the class to determine who is the best over all human.

1. Setup a rating system with at least 15 characteristics that can be rated one to ten.

Example:

Face beauty
 Hair beauty
 Sense of humor Etc.
 Intelligence
 Wit
 Social grace
 Coordination

2. Poll each student in the class regarding every other student in the class. This information will be typed into the computer.

3. Write a program which averages each person's statistics and sums them into a total 'Goodness Index'. (The Goodness Index is merely the average of all the averages.)

4. The program should then print this data in column format as shown below:

	F	H	H	I		G	C		I
	A	A	M	T	W	A	O		N
	C	I	O	E	I	C	R		E
	E	R	R	L	T	E	D	. . .	X
BEN GRICE	8.1	6.0	9.3	4.6	6.9	3.0	10.0	. . .	6.8
MR. BOUWSMA	1.5	5.2	0.3	3.5	0.5	6.0	1.1		2.6

Etc.

5.2 **The Game of Life:** Invented by mathematician John H. Conway, this game models the growth and changes in a complex collection of living organisms. The model can be interpreted as applying to a collection of microorganisms, an ecologically closed system of plants or animals, or of an urban development.

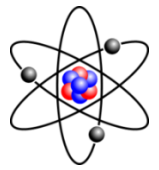
Start with a checkerboard grid on which counters are to be placed. Each location has eight neighbors. The counters are born, survive or die during a generation according to the following rules:

Survival: Counters with two or three neighboring counters survive to the next generation.

Death: Counters with four or more neighbors die from overcrowding and are removed for the next generation. Counters with zero or one neighbors die from isolation and are removed for the next generation.

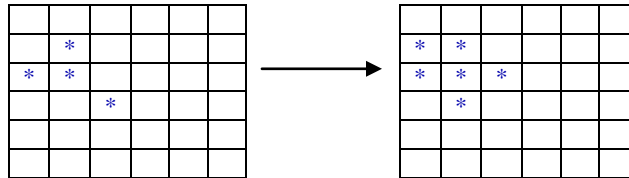
Birth: Each empty location which has exactly three counters in the eight neighboring locations is a birth location. A counter is placed in the location for the next generation.

COLONIAL CHRISTIAN SCHOOL



Example:

On a 6x6 space the pattern on the left would come to look like the pattern on the right after one generation.



□

Set up initial conditions by clearing the board and randomly placing counters in the grid. Show the grid on the screen and ask the user to press return for the next generation.

5.3 **Big Math Divide:** Nobody likes Exponential Notation. (At least, nobody who really matters.) Write a program that will do math with 15, 30, 100 digit numbers - what ever the person asks for. Do addition, subtraction, multiplication and division.

Example:

HOW MANY DIGITS MAXIMUM? 32

ENTER YOUR FIRST NUMBER:

1>123456789012345678901234567890

ENTER YOUR SECOND NUMBER:

2>555555555555555555555555555555

ENTER OPERATION:

(+ - * /) > +

```

523456789012345678901234567890
+ 555555555555555555555555555555
-----
1079012344567901234456790123445
    
```

Output for Addition and Subtraction and Multiplication should be printed out in the form shown above. Division should show the entered numbers above and below a line indicating division. An equal sign should be written below it with the answer following.

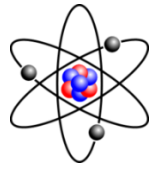
5.4 **Alarm Clock:** Write an alarm clock program which features ten different sounding alarms which are all programmable by day, date and time. Each alarm may sound several times during a day. (This program should be ideal for schools. There could be "Class Dismissal Bells" which sound a different tone from the "Tardy Bells". There could be a special "bell" sound for every period of the day. Chapel days or special schedule days would use the same sounds on an adjusted schedule which only occurs if it is the correct day of the week.)

Write the program so that it uses a text file for alarm scheduling information. Follow the following format:

```

1,MTWR,"8:00"
2,MTWR,"8:50"
1,F,"8:00"
... and so on ....
    
```

COLONIAL CHRISTIAN SCHOOL



The form for the data is:

{Alarm #},{Days to sound},{Time to sound}

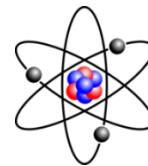
The days are abbreviated as follows:

M = Monday
T = Tuesday
W = Wednesday
R = Thursday
F = Friday
S = Saturday
X = Sunday

While the Alarm program is running, it should display the following on the screen:

1. The time of day as a clock face, the date and the day of the week.
2. A summary of the current day's schedule in a information box.
3. A logo or message identifying the program and its purpose with your name somewhere.

Alarms must be easy to hear, run for between 5 and 20 seconds and be easily distinguishable from each other.



The You Have Got To Be Kidding Level 6 Problems

6.1 **Calendar Program:** Write a program which will create a calendar for any date in history. Be sure to use the rules regarding leap years, centuries and 500 year special days. Each month should follow the standard format for full page hanging calendars. It should provide a large spacious box for appointments with the day number neatly written in the upper left hand corner, and provide a random full page high-res picture on a facing page.

Input the beginning month and year and the ending month and year in numeric form MM,YY to MM,YY.

All national holidays, like Thanksgiving, Christmas, Easter, Veteran's Day, Columbus Day, etc. as well as daylight savings information should print automatically in the correct box.

SAMPLE:

Input: ENTER BEGINNING MONTH AND BEGINNING YEAR:
MM,YYYY ->12,2022

ENTER ENDING MONTH AND ENDING YEAR:
MM,YYYY ->12,2022

Output:

A picture page appears here.

December 2022

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			1	2	3	4
5	6	7	8	9	10	11

This calendar continues this way for the rest of the page.
On the 25th, the word Christmas appears.

Prompting should be done on the screen. Printing should be done on paper.

Be sure to make the calendar as large as is possible on a standard sheet of paper. Be sure that the calendar will work for any date in history.