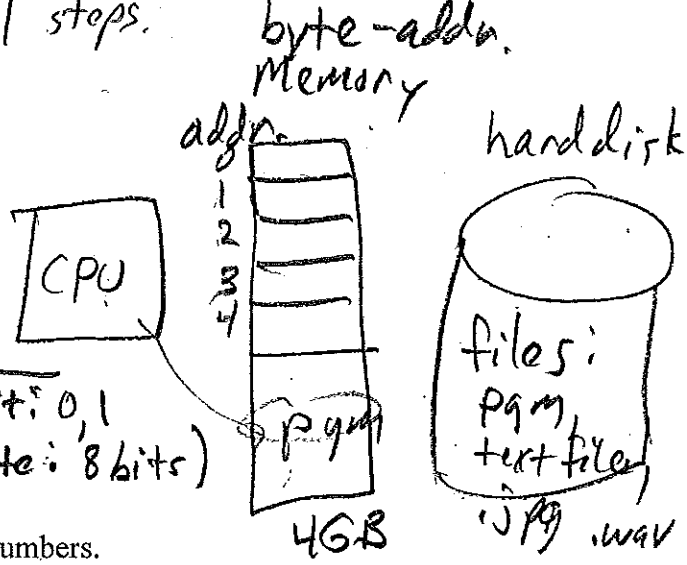
Media Computation Lecture 1 Name:_____

1. Write a detailed step-by-step process (i.e., an *algorithm*) to to exit the building if the fire alarm goes off.

- Don't Panic
- Stand up
- Push in chair
- Get in line
- Go down stairs
- exit building

high-level step - split into lower level steps.



2. What is the purpose of the following computer components?

- CPU/processor – processing "-running pgm
- main memory/RAM – running pgms + data stored
- hard-disk – files

bit: 0,1
byte: 8 bits)

3. Complete the following table about binary and hexadecimal numbers.

	Decimal (Base 10)	Binary (Base 2)	Hexadecimal (Base 16)
Number of digits:	10	2	
Digits:	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0, 1	
Counting:	0	0	
	1	01	
	2	00010	
	3	00011	
	4	00100	
	005	00101	
	006	00110	
	7	00111	
	8	01000	
	09	01001	
	10	01010	16 8 4 2 1
	11	01011	$2^4\ 2^3\ 2^2\ 2^1\ 2^0$
	12	01100	0 1 1 0 1
	13	01101	8+4+1 = 13
	14	01110	
	15	01111	
	16	10000	
	17		
	18		
	19		
	20		
	21		

10^2
1×100
7×10^1
3×10^0

4. Convert 173_{10} to a binary (base 2) value and then hexadecimal.

173
−128
45
−32
13

'A' 65_{10}

256 128 64 32 16 8 4 2 1
0 1 0 1 0 1 1 0 1$_2$

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

5. ASCII Character Representations are below. What do you notice about the following groups of characters?

0	NUL	16	DLE	32		48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	|
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

a) Upper-case letters: grouped together an increase alphabetically with increase in ASCII value

b) lower-case letters:

c) digits:

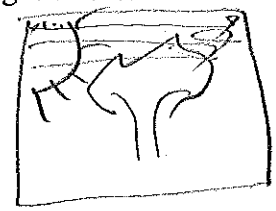
d) nonprintable characters: Mostly 0-31 ASCII values

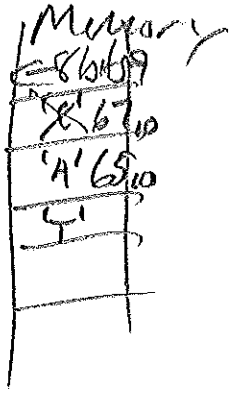
Abbreviations

NUL	Null
SOH	Start of heading
STX	Start of text
ETX	End of text
EOT	End of transmission
ENQ	Enquiry
ACK	Acknowledge
BEL	Bell (beep)
BS	Backspace
HT	Horizontal tab
LF	Line feed, new line
VT	Vertical tab
FF	Form feed, new page
CR	Carriage return
SO	Shift out
SI	Shift in
DLE	Data link escape
DC1	Device control 1
DC2	Device control 2
DC3	Device control 3
DC4	Device control 4
NAK	Negative acknowledge
SYN	Synchronous idle
ETB	End of transmission block
CAN	Cancel
EM	End of medium
SUB	Substitute
ESC	Escape
FS	File separator
GS	Group separator
RS	Record separator
US	Unit separator
DEL	Delete/Idle

6. *Digitization* - is the process of converting *analogy* media (e.g., photographs, vinyl records) by encoding many small pieces into digital (binary #'s) *data representations* and organizing these pieces into *data structure.* For example, a string of text might use ASCII for the data representation of each character with a 1-D "array" as the data structure for the string of text. For an image what might be the:

data representation (encoding of small piece) - pixel - color RGB 8bit # # # # alpha



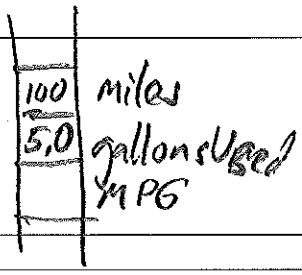


data structure (collection of pieces) - 2D array of pixels

1. In the following Python program, identify the following:

- comments: statements in the program for the human reader which are ignored by the computer.
- assignment statements: *<variableName>* = *<value to be stored>* which saves the value on the right-side of the assignment operator ("=") to memory so it can later be looked up using the variable name
- programmer-defined identifiers: symbolic names in the program that the programmer creates for variables, functions, etc. (starts with letter or underscore then followed by letters, underscores, or digits)
- reserved words: words that have special meaning in the Python programming language
- operators: operators perform operations on one or more operands. An operand is usually a piece of data, like a number or a variable's value.

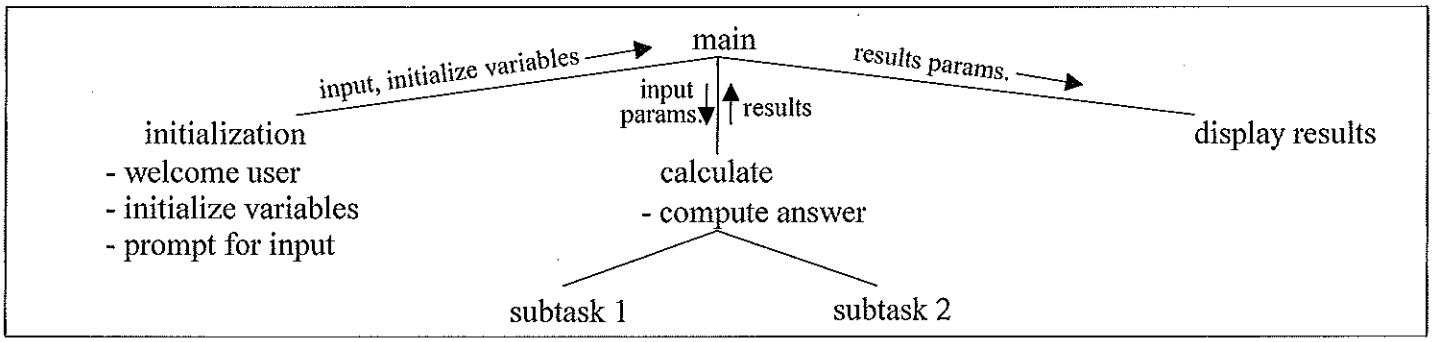
```
# Python program to calcuate miles per gallon
miles = 100
gallonsUsed = 5.0
MPG = miles / gallonsUsed
print "Your mileage was", MPG, "miles per gallon."
```




```
Your mileage was 20.0 miles per gallon.
```

2. What would make the above program better? user input

3. Most simple programs have a similar functional-decomposition with tasks: input, calculate, output




```
def main():
  """ Sequences the main tasks of the program. """
  miles, gallonsUsed = getMilesAndGallons()
  MPG = calculateMileage(miles, gallonsUsed)
  displayMileage(MPG)

def getMilesAndGallons():
  """ Gets the user's input of miles and gallons. """
  miles = requestNumber("Enter the miles traveled: ")
  gallons = requestNumber("Enter the gallons used: ")
  return miles, gallons

def displayMileage(MPG):
  """ Displays the resulting mileage to the user."""
  print "Your mileage was",MPG,"miles per gallon."

def calculateMileage(miles, gallonsUsed):    # <<<<<<<<<<<<<< COMPLETE THIS FUNCTION
  """ Calculates the miles per gallon."""



main()   # Starts the program running
```