Problem Solving Session

- The remainder of today's class will comprise the ***problem solving session*** (***PSS***).
- Your instructor will divide you into ***teams of 3 or 4 students***.
- Each team will ***work together*** to solve the following problems over the course of ***20-30 minutes***.
  - You may work on paper, a white board, or digitally as determined by your instructor.
  - You will submit your solution by pushing it to GitHub before the end of class.
- Your instructor will go over the solution before the end of class.
- If there is any time remaining, you will begin work on your homework assignment.

Class participation is a significant part of your grade (20%). This includes in class activities and the problem solving session.

Your Course Assistants will grade your participation by verifying that you pushed your solutions before the end of the class period each day.

Problem Solving Team Members

Record the name of each of your problem solving team members here.

Do not forget to ***add every team member's name***! Your instructor (or course assistant) may or may not use this to determine whether or not you participated in the problem solving session.

Luke Demi
Logan Nickerson

Problem Solving 1

Ascii characters can be divided into 4 classes:

- Uppercase letters
- Lowercase letters
- Digits
- Symbols

Using the ascii table below, find the decimal value range(s) of each of the classes and record them. For the purposes of this exercise, we will only consider *visible* characters, i.e. ascii values 33 to 126. The start of the range should be *inclusive* and the end of the range should be *exclusive*.

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Uppercase Letters
65-91

Lowercase Letters
97-123

Digits
48-58

Symbols
33-48, 58-65,
91-97, 123-127

```
def create_ascii_range_string(start, stop):
     product = “”
     values = range(start, stop)
     for value in values:
          product += chr(value)
     return product
```

Problem Solving 2

Suppose we wanted to create a string containing a sequence of consecutive characters within a range of ascii values. For example, a starting ascii value of 97 (inclusive) and a stopping ascii value of 102 (exclusive), would yield the string "`abcde`".

To the left, write a function named `create_ascii_range_string` that takes two integer parameters: `start` and `stop`. The function should *create* and *return* a string consisting of consecutive characters from the `start` value (inclusive) up to the `stop` value (exclusive).

Problem Solving 3

Python includes a `random` module with functions to generate a random value between a start and stop value.

- `randint(a,b)` - returns a random value between a (inclusive) and b (inclusive)
- `randrange(a,b)` - returns a random value between a (inclusive) and b (exclusive)

Complete the `get_random_char_from_string(a_string)` function to the right to return a random character from the given string.

```
import random

def get_random_char_from_string(a_string):
     return a_string[randrange(0,
len(a_string))]
```

Problem Solving 4

Suppose we have 4 characters: "A", "a", "0", "#" and wanted to create a string with a given number of each of these characters in a random order.

For example, a randomly ordered string with 4 "A", 2 "a", 3 "0", and 1 "#" may result in a string that looks like "aA#A00aAA0".

To the right, write the code to generate and print a random string using the characters and count of each character above. You may hardcode the characters and the character counts.

```
import random
def make_random_string(): #space saving
      chr_1 = “A”  amt_1 = 4
      chr_2 = “a”  amt_2 = 3
      chr_3 = “0”  amt_3 = 2
      chr_4 = “#”  amt_4 = 1
      code = “”
      while amt_1 + amt_2 + amt_3 + amt_4 > 0:
            choice = randint(1, 4)
            if choice == 1:
                  if amt_1 != 0:
                        code += chr_1
                        amt_1 -= 1
            elif choice == 2:
                  if amt_2 != 0:
                        code += chr_2
                        amt_2 -= 1
            elif choice == 3:
                  if amt_3 != 0:
                        code += chr_3
                        amt_3 -= 1
            else:
                  if amt_4 != 0:
                        code += chr_4
                        amt_4 -= 1

      return code
```