

Laboratory work: 4

Sorting

1. Read Chapter 11 (Lists)
2. Watch a quick introduction to the “bubble sort” method - <https://www.youtube.com/watch?v=Vca808JTbI8>
3. Create a blank 40.py file. Fill 40.py with the code provided in the attachment.

Elementary statistical characteristics

1. Create new file **41.py** by copying the contents of **40.py**. The goal of the program **41.py** is to determine the following statistics characteristics of the array (list):
 - (a) The number of elements in set
 - (b) The minimum numeric value of a set element
 - (c) The maximum numeric value of a set element
 - (d) The arithmetic average of the set
 - (e) The median value of the set
Median is the value of the average element of sorted set. If a set contains an odd number of elements, then the median is value of the average element in that set. If a set contains even number of elements, then median is the arithmetic average of the two middle elements
 - (f*) The mode value of the set
The mode of a set is the most commonly occurring value. If several different numbers are more common than others and are in the same number in each group, then this set is multimodal. Then, in addition to the values of modes, display the number of modes
 - (g*) The number of modes
2. Create script **42.py** that arranges your name or surname
 - (a) The name or surname must be written without accents or dashes. Use only English letters
 - (b) Size of the list is selected by the author of the program
 - (c) The corresponding character-numeric values can be found in the ASCII table (annex). Use ord() function to get the corresponding ASCII value
 - (d) Display numerical values of the ordered list and their corresponding letters
 - (e) Calculate statistical characteristics - see. point 1a - 1g
3. Create **43.py** that sorts any word you enter with up to 20 characters
 - (a) The text is entered from the keyboard
 - (b) The corresponding character-numeric values can be found in the ASCII table
 - (c) Display numerical values of the ordered list and their corresponding letters

Annex

```
# L40.py

def sort(nums):

    for i in range(len(nums)-1,0,-1):
        for j in range(i):
            if nums[j]>nums[j+1]:
                temp = nums[j]
                nums[j] = nums[j+1]
                nums[j+1] = temp

nums = [5, 3, 8, 6, 7, 2]
print("Given: ", nums)

sort(nums)

print("Sorted: ",nums)
```

```
# L42.py

def sort(nums):

    for i in range(len(nums)-1,0,-1):
        for j in range(i):
            if nums[j]>nums[j+1]:
                temp = nums[j]
                nums[j] = nums[j+1]
                nums[j+1] = temp

nums = ["J", "A", "N", "I","S"];

print("Given: ", nums)
sort(nums)
print("Sorted: ",nums)

print("-----")

for k in range(len(nums)):
    print(nums[k], " ", ord(nums[k]))
```

ASCII table

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	TAB	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