

Assignment 1

Basic Java Syntax

COMP1004/1406 Introduction to Computer Science II
Summer 2022

Due date: July 13, submit your work on Brightspace.

1. Moving Average

In statistics the moving average is a calculation to analyze data points by creating a series of averages of different subsets of the full data set. The moving average for any given data point (using width =7) is the average of that data point along with the 6 previous data points. This is known as a one-sided moving average. The formula for a moving average (MA) of X at time t with a width 7 is as follow:

$$MA_7 = \frac{X_{t-6} + X_{t-5} + X_{t-4} + X_{t-3} + X_{t-2} + X_{t-1} + X_t}{7} \quad (1)$$

In A1Q1.java complete the `movingAverage()` method. This method will compute the moving average of some data using a specified width (number of data points used for the average). This method takes an array of floating point numbers (called data) and an integer called (width) as input. The method returns a new array that is the same size as the input array and will contain the moving average of the input array data using width data points for each value.

```
public static double[] movingAverage(double[] data, int width)
```

For each position where there is not enough data points (width) to compute the average, you will use the NaN (Not-A-Number) value. In particular, the first width-1 elements of the output array should have this value. Here are some example:

```
A1Q1.movingAverage(new double[]{1,2,4}, 1) -> [1.0, 2.0, 4.0]
A1Q1.movingAverage(new double[]{1,2}, 2) -> [NaN, 1.5]
A1Q1.movingAverage(new double[]{1,2,4,9}, 3) -> [NaN, NaN, 2.3333333333333335, 5.0]
A1Q1.movingAverage(new double[]{1,2,4}, 4) -> [NaN, NaN, NaN]
```

Read on Java's Double class to learn more about the value NaN.

Restrictions:

```
data: 0 <= data.length <= Integer.MAX_VALUE/100
size: 1 <= width <= Integer.MAX_VALUE
```

2. Formatting

Write a function called `MultiplicationTable()` in `A1Q2.java` that prompts the user for a positive integer; the program should then prints the multiplication table as shown:

Enter the size: 3

```
* | 1 2 3
```

```
1 | 1 2 3
```

```
2 | 2 4 6
```

```
3 | 3 6 9
```

Enter the size: 10

```
* | 1 2 3 4 5 6 7 8 9 10
```

```
1 | 1 2 3 4 5 6 7 8 9 10
```

```
2 | 2 4 6 8 10 12 14 16 18 20
```

```
3 | 3 6 9 12 15 18 21 24 27 30
```

```
4 | 4 8 12 16 20 24 28 32 36 40
```

```
5 | 5 10 15 20 25 30 35 40 45 50
```

```
6 | 6 12 18 24 30 36 42 48 54 60
```

```
7 | 7 14 21 28 35 42 49 56 63 70
```

```
8 | 8 16 24 32 40 48 56 64 72 80
```

```
9 | 9 18 27 36 45 54 63 72 81 90
```

```
10 | 10 20 30 40 50 60 70 80 90 100
```

*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75	78	81	84	87	90	93	96	99	102	105
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	100	104	108	112	116	120	124	128	132	136	140
5	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	125	130	135	140	145	150	155	160	165	170	175
6	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120	126	132	138	144	150	156	162	168	174	180	186	192	198	204	210
7	7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119	126	133	140	147	154	161	168	175	182	189	196	203	210	217	224	231	238	245
8	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160	168	176	184	192	200	208	216	224	232	240	248	256	264	272	280
9	9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180	189	198	207	216	225	234	243	252	261	270	279	288	297	306	315
10	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300	310	320	330	340	350
11	11	22	33	44	55	66	77	88	99	110	121	132	143	154	165	176	187	198	209	220	231	242	253	264	275	286	297	308	319	330	341	352	363	374	385
12	12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192	204	216	228	240	252	264	276	288	300	312	324	336	348	360	372	384	396	408	420
13	13	26	39	52	65	78	91	104	117	130	143	156	169	182	195	208	221	234	247	260	273	286	299	312	325	338	351	364	377	390	403	416	429	442	455
14	14	28	42	56	70	84	98	112	126	140	154	168	182	196	210	224	238	252	266	280	294	308	322	336	350	364	378	392	406	420	434	448	462	476	490
15	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255	270	285	300	315	330	345	360	375	390	405	420	435	450	465	480	495	510	525
16	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288	304	320	336	352	368	384	400	416	432	448	464	480	496	512	528	544	560
17	17	34	51	68	85	102	119	136	153	170	187	204	221	238	255	272	289	306	323	340	357	374	391	408	425	442	459	476	493	510	527	544	561	578	595
18	18	36	54	72	90	108	126	144	162	180	198	216	234	252	270	288	306	324	342	360	378	396	414	432	450	468	486	504	522	540	558	576	594	612	630
19	19	38	57	76	95	114	133	152	171	190	209	228	247	266	285	304	323	342	361	380	399	418	437	456	475	494	513	532	551	570	589	608	627	646	665
20	20	40	60	80	100	120	140	160	180	200	220	240	260	280	300	320	340	360	380	400	420	440	460	480	500	520	540	560	580	600	620	640	660	680	700
21	21	42	63	84	105	126	147	168	189	210	231	252	273	294	315	336	357	378	399	420	441	462	483	504	525	546	567	588	609	630	651	672	693	714	735
22	22	44	66	88	110	132	154	176	198	220	242	264	286	308	330	352	374	396	418	440	462	484	506	528	550	572	594	616	638	660	682	704	726	748	770
23	23	46	69	92	115	138	161	184	207	230	253	276	299	322	345	368	391	414	437	460	483	506	529	552	575	598	621	644	667	690	713	736	759	782	805
24	24	48	72	96	120	144	168	192	216	240	264	288	312	336	360	384	408	432	456	480	504	528	552	576	600	624	648	672	696	720	744	768	792	816	840
25	25	50	75	100	125	150	175	200	225	250	275	300	325	350	375	400	425	450	475	500	525	550	575	600	625	650	675	700	725	750	775	800	825	850	875
26	26	52	78	104	130	156	182	208	234	260	286	312	338	364	390	416	442	468	494	520	546	572	598	624	650	676	702	728	754	780	806	832	858	884	910
27	27	54	81	108	135	162	189	216	243	270	297	324	351	378	405	432	459	486	513	540	567	594	621	648	675	702	729	756	783	810	837	864	891	918	945
28	28	56	84	112	140	168	196	224	252	280	308	336	364	392	420	448	476	504	532	560	588	616	644	672	700	728	756	784	812	840	868	896	924	952	980
29	29	58	87	116	145	174	203	232	261	290	319	348	377	406	435	464	493	522	551	580	609	638	667	696	725	754	783	812	841	870	899	928	957	986	1015
30	30	60	90	120	150	180	210	240	270	300	330	360	390	420	450	480	510	540	570	600	630	660	690	720	750	780	810	840	870	900	930	960	990	1020	1050
31	31	62	93	124	155	186	217	248	279	310	341	372	403	434	465	496	527	558	589	620	651	682	713	744	775	806	837	868	899	930	961	992	1023	1054	1085
32	32	64	96	128	160	192	224	256	288	320	352	384	416	448	480	512	544	576	608	640	672	704	736	768	800	832	864	896	928	960	992	1024	1056	1088	1120
33	33	66	99	132	165	198	231	264	297	330	363	396	429	462	495	528	561	594	627	660	693	726	759	792	825	858	891	924	957	990	1023	1056	1089	1122	1155
34	34	68	102	136	170	204	238	272	306	340	374	408	442	476	510	544	578	612	646	680	714	748	782	816	850	884	918	952	986	1020	1054	1088	1122	1156	1190
35	35	70	105	140	175	210	245	280	315	350	385	420	455	490	525	560	595	630	665	700	735	770	805	840	875	910	945	980	1015	1050	1085	1120	1155	1190	1225