ISS Numerical exercise #1
Reference solution

Signals – 1, 2, 3

n	-2	-1	0	1	2	3	4	5	6
$x_1[n]$	0	0	1	1	1	1	0	0	0
$x_2[n]$	-1	0	1	2	3	4	5	6	7
$x[n]$	-1	0	2	3	4	5	5	6	7

Time shifts – 4, 5, 6, 7, 8

n	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
$x[n]$						3	2	1				
$y_1[n]=x[n-2]$								3	2	1		
$y_2[n]=x[n+2]$				3	2	1						
$y_3[n]=x[-n]$				1	2	3						
$y_4[n]=x[1-n]$					1	2	3					
$y_5[n]=x[-2-n]$		1	2	3								

checks: – choose an important event in the shifted signal
– evaluate the time modification for its position 'n'
– look in the original signal, if the same event is there.

Example: $y_5[n]=x[-2-n]$, looking for sample 3 placed at $n=-2$. $-2-(-2)=0$, look in the original at $n=0$, yes, it's there!

Filters

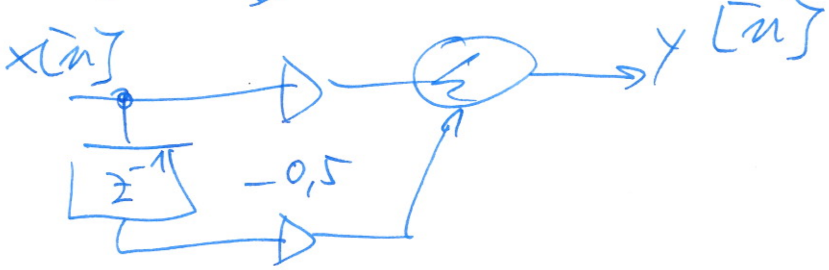
9. It performs <u>averaging</u> of two neighboring samples. Averaging ≈ smoothing. Smooth signal means low frequencies ⇒ low pass.

10. Performs <u>difference</u> of neighboring samples. Kills constants (D.C. values), amplifies fast differences. Fast signal ≈ high frequencies ⇒ high pass.

11. Produces just a constant. Not really a filter...

12.

13.

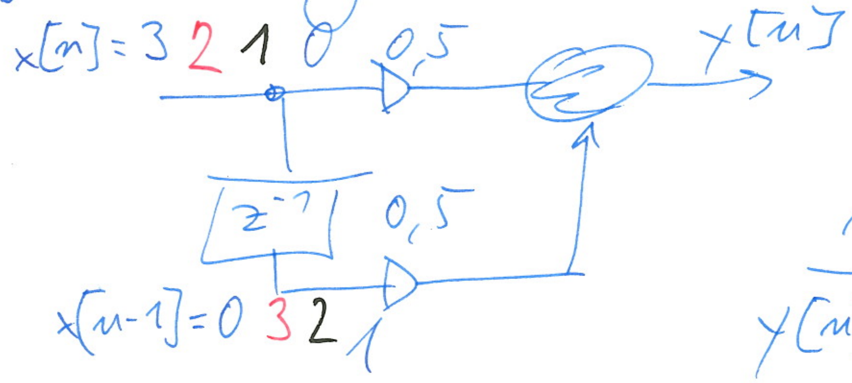


14. and 15

n	-2	-1	0	1	2	3
$h_{12}[n]$			0,5	0,5		
$h_{13}[n]$			0,5	-0,5		

Filtering and convolution

16.



n	0	1	2	3	4 …
y[n]	1,5	2,5	1,5	0,5	0 …

17. Similarly …

n	0	1	2	3	4 …
y[n]	1,5	−0,5	−0,5	−0,5	0 …

18. Convolution

k	−3	−2	−1	0	1	2	3	4	5
x[k]				3	2	1			
h[0−k]			0,5	0,5					
h[1−k]				0,5	0,5				
h[2−k]					0,5	0,5			
h[3−k]						0,5	0,5		
y[n]				1,5	2,5	1,5	0,5		

always multiply and add

19. Similarly …

k	−3	−2	−1	0	1	2	3	4	5
x[k]				3	2	1			
h[n−k]			−0,5	0,5					
				−0,5	0,5				
					−0,5	0,5			
						−0,5	0,5		
y[n]				1,5	−0,5	−0,5	−0,5	0	0 …

20. The other way round ...

k	-3	-2	-1	0	1	2	3	4	5
h[k]				0,5	-0,5				
		1	2	3					
x[n-k]			1	2	3				
				1	2	3			
					1	2	3		
y[n]				1,5	-0,5	-0,5	-0,5	0	0 ...

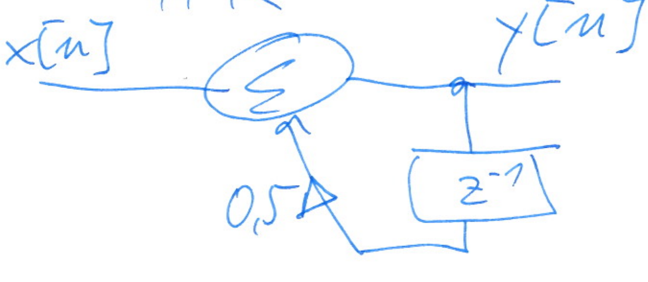
Yes, the result is the same, good :-)

21. The 1st filter made the signal smoother
=> probably low pass.
The 2nd one generated more changes
=> probably high-pass.

22.

```
float filter (float xn) {
    static float xnminus1 = 0.0;
    float yn;
    yn = 0.5 * xn + 0.5 * xnminus1;
    xnminus1 = xn;
    return yn;
}
```

IIR

23. 

24.

n	-2	-1	0	1	2	3	4	5	6		
u[n]	0	0	1	0,5	0,25	0,125	. .	-	-	-	.

25.

n	-2	-1	0	1	2	3	4	5	6
x[n]			1	1					
y[n]	0	0	1	1,5	0,75	0,375	.	. .	- -

26. $-a_1$ must be in interval $<-1, +1>$, otherwise infinities!

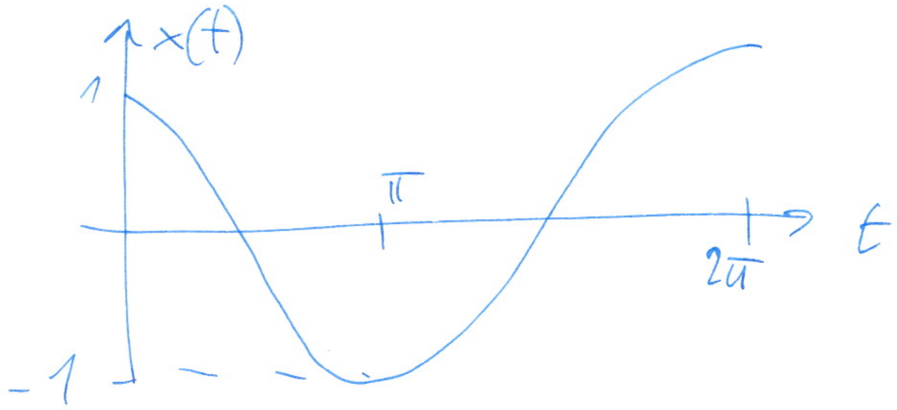
$$|a_1| < 1$$

27.

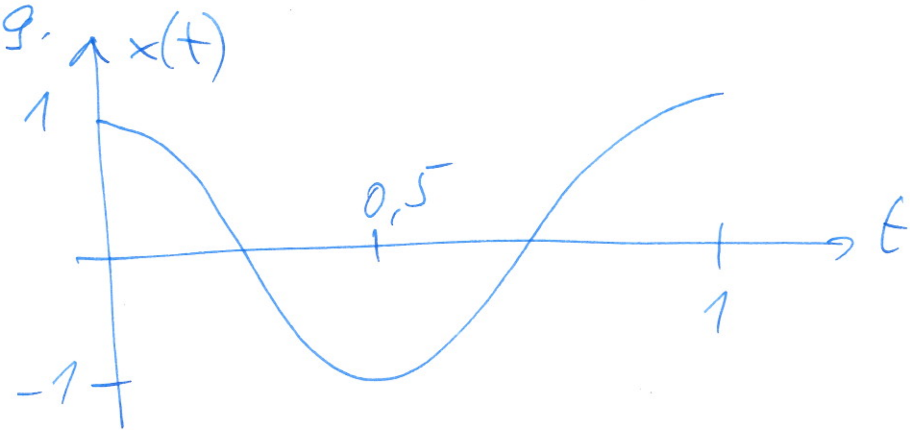
```
float iir (float xn) {
    static float ynm1 = 0.0;
    float yn;
    yn = xn + 0.5 * ynm1;
    ynm1 = yn;
    return yn;
}
```

cosines

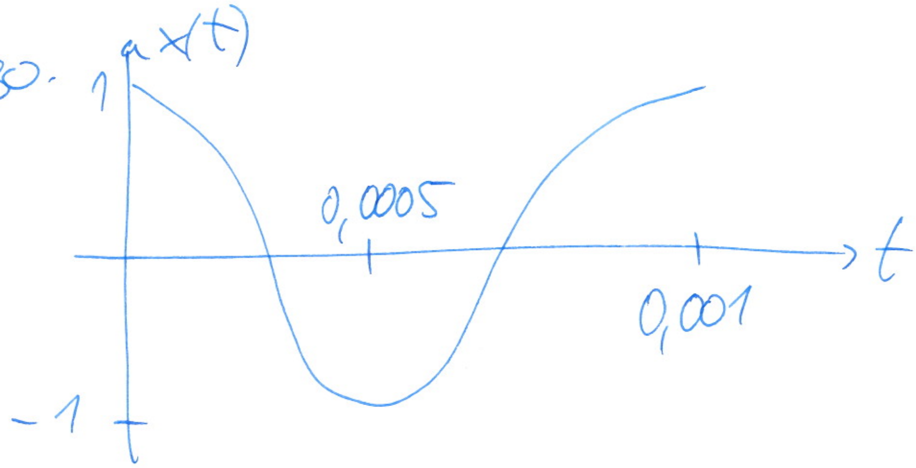
28.



29.

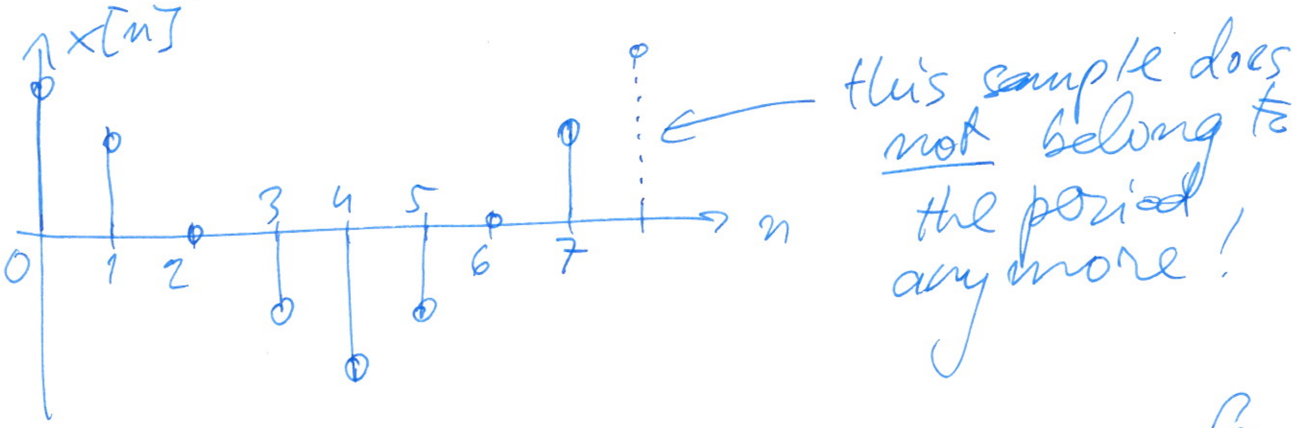


30.



31. period $T = 0{,}001\,s$. Frequency $f = \frac{1}{T} = 1000\,Hz$.

32.

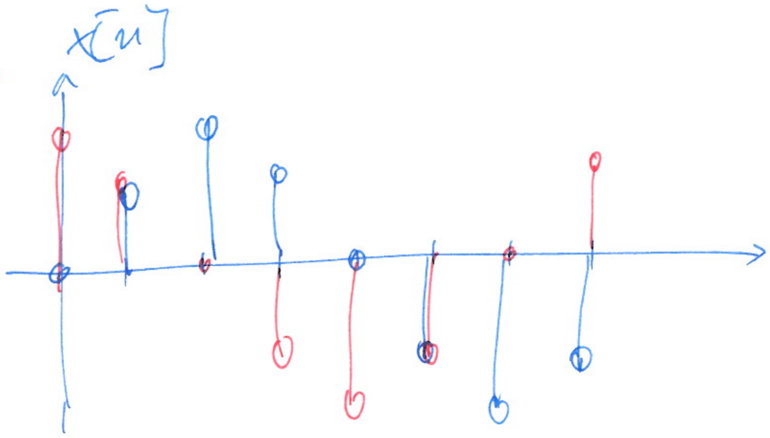


33. $N = 8$ $f' = \frac{1}{N} = \frac{1}{8}$ ← norm. freq. has no unit.

34. $f = f' \cdot F_s = \frac{1}{8} \cdot 8000 = 1000\,Hz$

35. $f = f' \cdot F_s = \frac{1}{8} \cdot 48k = 6\,kHz$

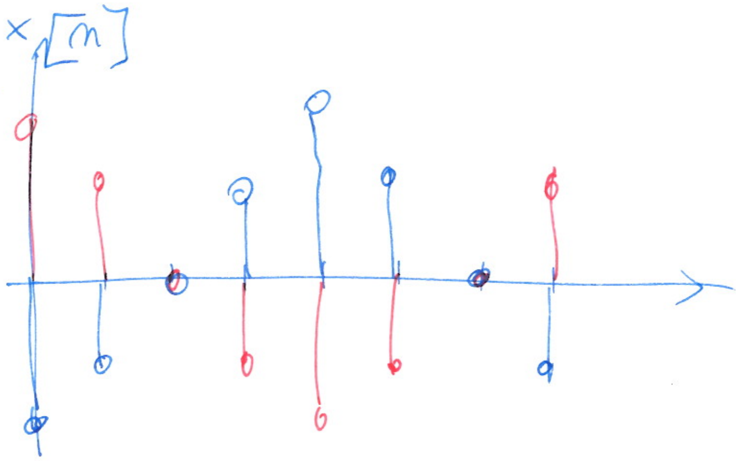
36



red is the original one

$-\frac{\pi}{2} \approx$ delay of quarter a period

37.

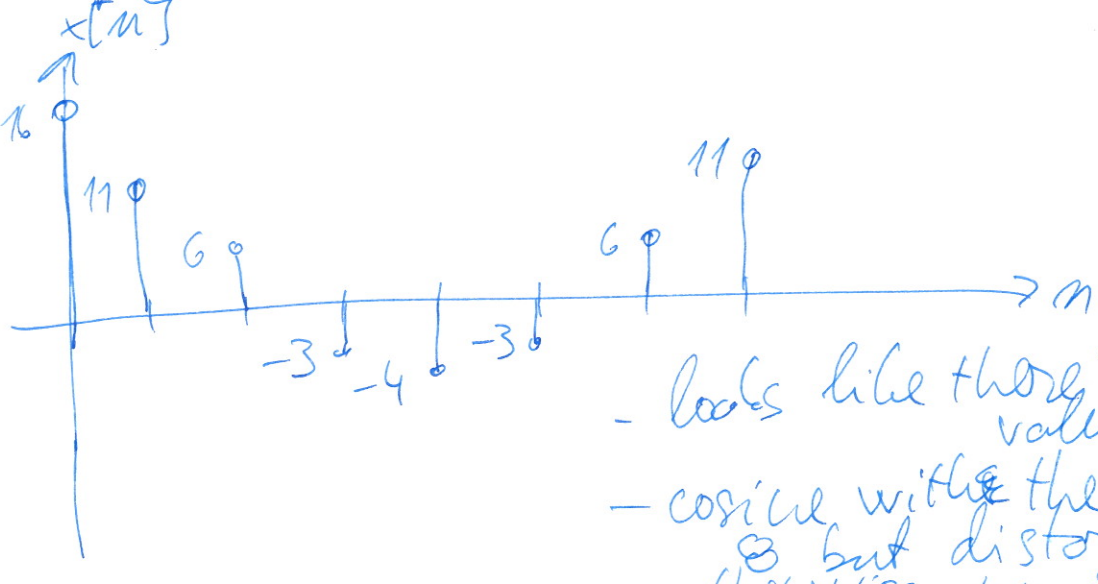


$+\pi \approx$ advance half a period

this is actually a minus cosine...

Analysis – advice the students to use calculators or cell phones to perform the summation...
For exercises 39 – 42 also ask them to divide the results by the number of samples = 8.

38.



- looks like there's a D.C. value.
- cosine with the period of 8 but distorted.
- otherwise, no idea...

39. A simple sum of x[n] values ...

$\sum x[n]a[n] = 40$ $\frac{40}{8} = 5$ wow, this looks like the D.C. value.

40.

16	11	6	-3	-4	-3	6	11
1	0,7	0	-0,7	-1	-0,7	0	0,7
16	7,7	0	2,1	4	2,1	0	7,7

$\sum x[n]a[n] = 39,6$ $\frac{39,6}{8} \doteq 5$

strong presence of cosine with the period of 8!

41.

16	11	6	-3	-4	-3	6	11
1	0	-1	0	1	0	-1	0
16	0	-6	0	-4	0	-6	0

$\sum x[n]a[n] = 0$ No contribution of cosine with period of 4.

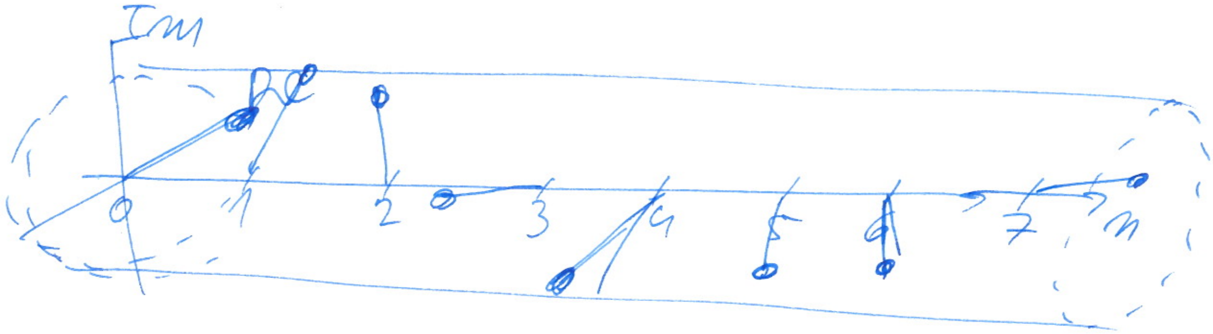
42.

16	11	6	-3	-4	-3	6	11
1	-1	1	-1	1	-1	1	-1
16	-11	6	3	-4	3	6	-11

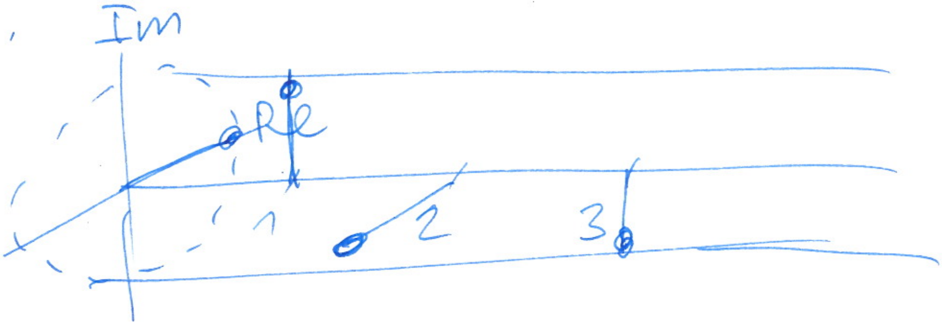
$\sum x[n]a[n] = 8$ $\frac{8}{8} = 1$

presence of cosine with period $N=2$, ~~[illegible]~~ 5 times weaker than the main one, so it was not visible, but we found it!!!

43.



44.



---

45 and 46 DFT

+ some more values if time allows ...

k	0	1	2	3	4	5	...	128	...	255	~~256~~
f'	0	1/256	2/256	3/256	4/256	5/256		1/2		255/256	~~1~~
f [Hz]	0	250	500	750	1000	1250	..	32k		63,75k	~~64k~~

this is not allowed, DFT returns only 0..255 !

46:
```
k = 0:(N-1);
fnorm = k/N;
f     = fnorm * Fs;
```