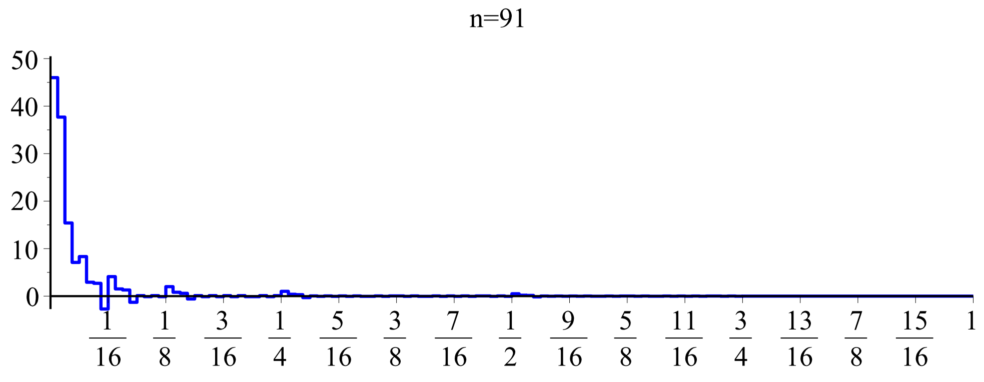
```
> restart:
  libname:="C:\\Maple\\lib",libname:
  with(WalshPaley):
```

The package contains some commands to handle the Fejér means and kernels with respect to the Walsh-Paley system.
These commands work similarly to the ones for handling the Fourier series and Dirichlet kernels.
The Fejér kernels can be obtained by the comands `KVal(n)` and `KFunc(n,x)`.
The values are calculated recursively, hence we do not deal with setting the dimension.
Create an animation with the first 100 Fejér kernels.

```
> plots[display](seq(lineplot(KVal(n),tozero,title=cat("n=",n),
  color="blue"), n=1..100), insequence=true,tickmarks=[[seq(i/16=
  i/16,i=1..16)],default]);
```



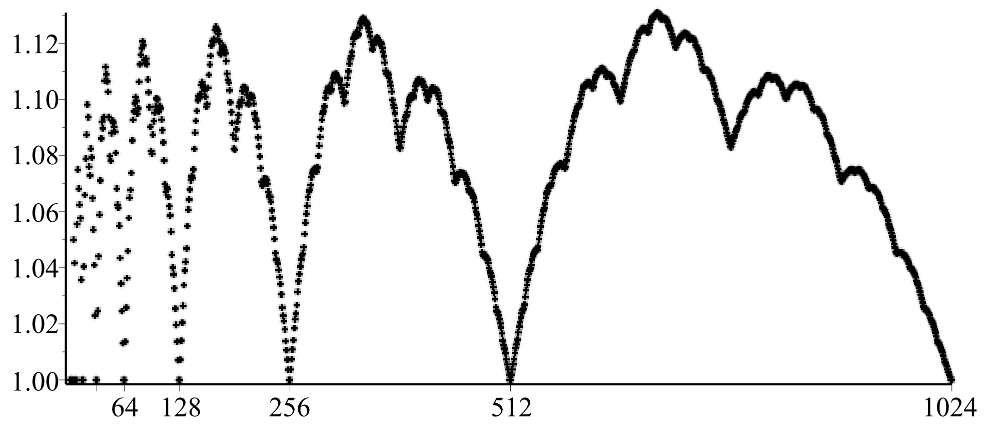
Calculate the Fejér kernels with index 5 at the point 0.8.

```
> KFunc(5,0.8);
```

$$\frac{1}{5} \tag{1}$$

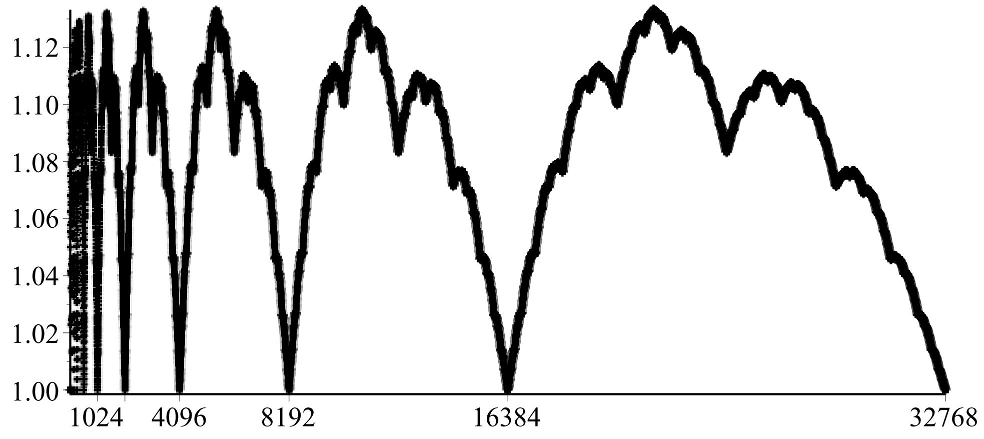
Plot the L1-norm of Fejér kernels from 0 up to 1024.

```
> m:=2^10:
  plots[pointplot]({seq([n, NormValues[1](KVal(n))], n = 1 .. m)},
  symbol=diamond, symbolsize=4,tickmarks=[[seq(2^i,i=5..10)],
  default]);
```



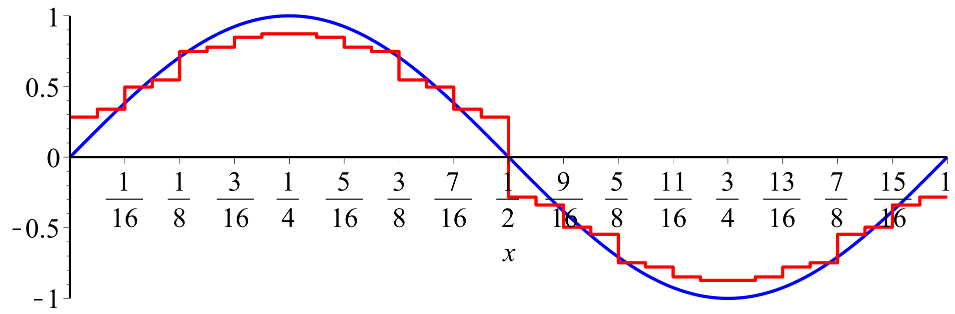
A better alternative to calculate the L1-norm of Fejér kernels is the use of KNorm procedure which is based in a new faster formula.
With the use of this formula we can calculate the L1-norm of Fejér kernel for very large values of n in a relatively short period of time.

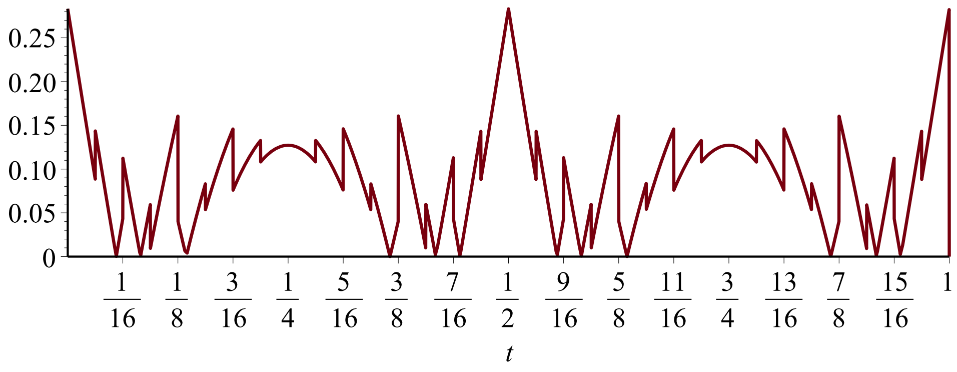
```
> m:=2^15:
  plots[pointplot]({seq([n, KNorm(n)], n = 1 .. m)}, symbol=
  diamond, symbolsize=4,tickmarks=[[seq(2^i,i=10..15)],default]);
```



The commands **FejerMVal(n,f)** and **FejerMFunc(n,f,x)** gives us the **n**-th Fejér mean of the function **f**. This commands compute the partial sums from the values of Fejér kernels using convolution.
Like Fourier series it is not necessary to set dimensions.
We can use the option floating to calculate the values numerically.
Plot the 28-th Fejér mean for f(x)=sin(2*PI*x) and the absolute difference between the function and the partial sum..

```
> f:=x->sin(2*Pi*x):n:=28:
  fPlot:=plot(f(x),x=0..1,color=blue):
  FejerPlot:=lineplot(FejerMVal(n,f(x),floating),color=red):
  plots[display]([fPlot, FejerPlot],tickmarks=[[seq(i/16=i/16,i=1.
  .16)],default]);
  plot(abs(sin(2*Pi*t)-FejerMFunc(28,f(x),t)),t=0..1,tickmarks=[
  [seq(i/16=i/16,i=1..16)],default]);
```



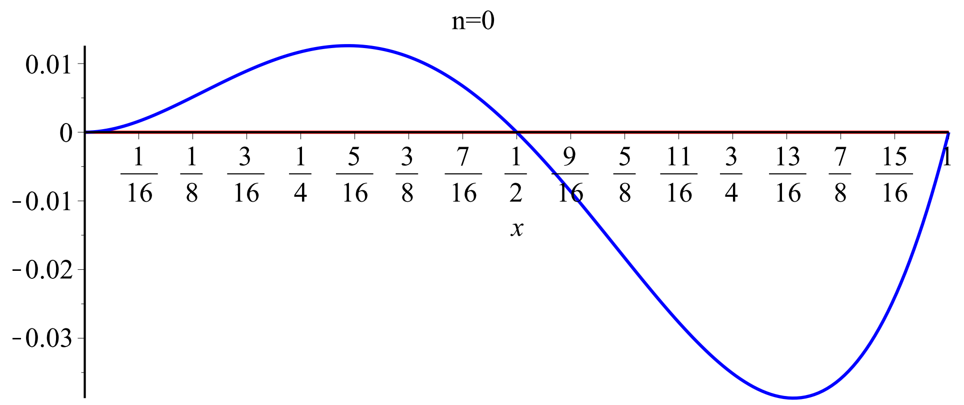


The command **FejerMsVal(k,f)** returns a matrix with the values of Fejér means from 1 up to 2^k. We suggest to use this matrix to create animations.

```
> SetDim(9):
  f:=x-> x^2*(x-1)*(x-1/2):
  n:=500:
```

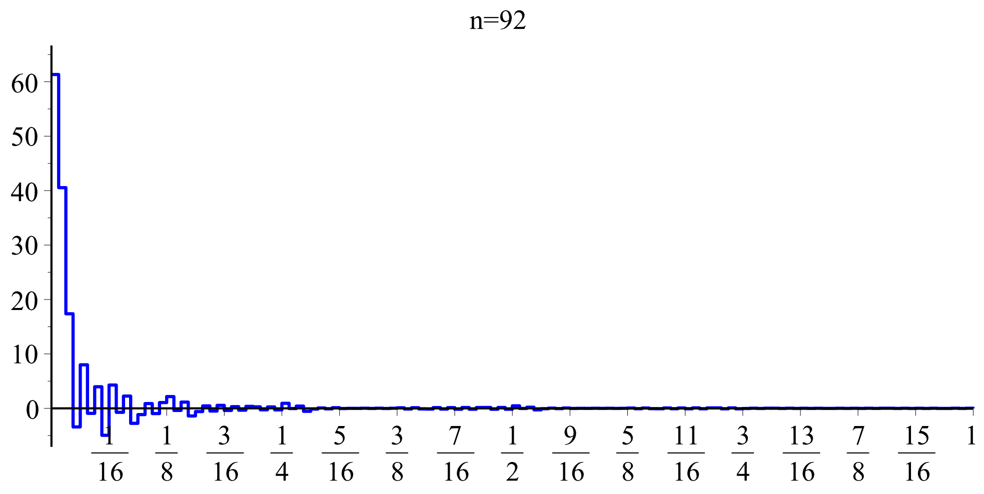
```
> fPlot:=plot(f(x),x=0..1,color=blue):
  FMV:=FejerMsVal(9,f(x),floating):
  for i from 1 to n do
      FejerPlots[i]:=lineplot(FMV[i], title=cat("n=",i), color=red)

  end do:
  FejerPlots[0]:=lineplot([0], title=cat("n=",0), color=red):
  FejerSeq:=plots[display](seq(FejerPlots[i], i=0..n), insequence=
  true):
  plots[display](FejerSeq,fPlot,tickmarks=[[seq(i/16=i/16,i=1..16)
  ],default]);
```



The module also contain procedures for calculating Cesaro kernels and means of order alpha when 0 < alpha < 1.
The use of these procedures is similar to the ones used for Fejér kernes and means.
However, in case of Cesaro kernels and means of order alpha, we calculate the procedures by the appropriate linear combination of Walsh functions. It means that the values of Walsh functions are required, so we need to set the dimension to work with a large value of the indeces.
Create an animation with the first 100 Cesaro kernels of order alpha=1/2.

```
> SetDim(7);
  plots[display](seq(lineplot(KCVal[1/2](n),tozero,title=cat("n=",
  n),color="blue"), n=1..100), insequence=true,tickmarks=[[seq
  (i/16=i/16,i=1..16)],default]);
```



Note that procedure **KCVal[alpha](n)** did not work as fast as procedure **KVal(n)** since the latter is implemented by iteration.

Calculate the Cesaro kernels of order alpha=1/2 with index 5 at the point 0.8.

```
> KCFunc[1/2](5,0.8);
```

$$\frac{230}{693} \qquad \textbf{(2)}$$

It is known that the Cesaro kernels of order alpha=0 are the Dirichlet kernels and the Cesaro kernels of order alpha=1 are the product of the Fejér kernels by n/(n+1).
Indeed, we can ensure the above as follows.

```
> interface(rtablesize=32):
  n:=29:KCVal[0](n);DiriVal(n);
[29, 3, 3, -3, 3, -3, -3, 3, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1,
    -1, 1, 1, -1]
[29, 3, 3, -3, 3, -3, -3, 3, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1,     (3)
    -1, 1, 1, -1]
```

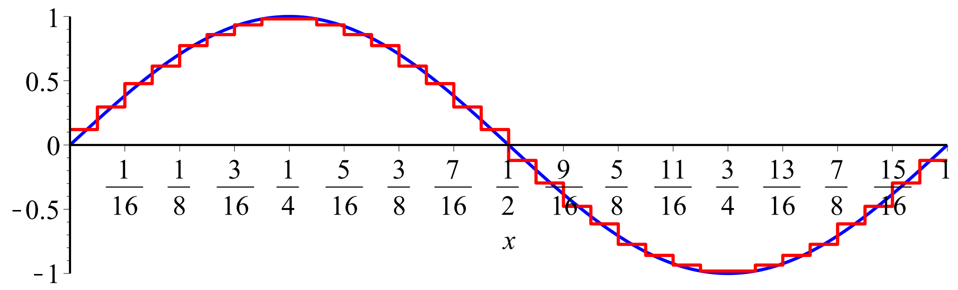
```
> interface(rtablesize=32):
  n:=29:KCVal[1](n);(n/(n+1))*KVal(n);
```

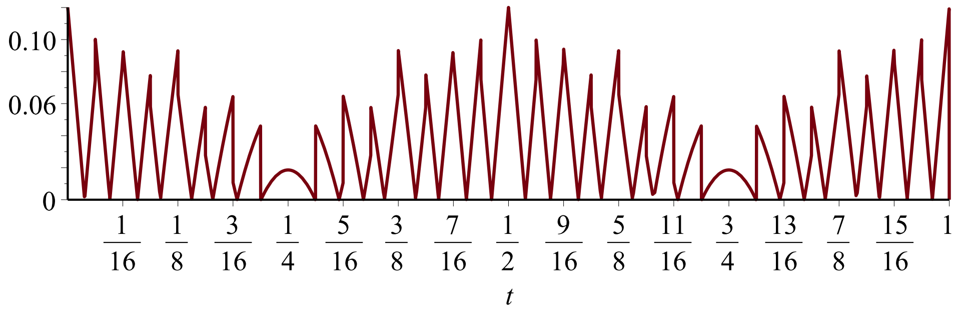
$$\left[\frac{29}{2}, \frac{253}{30}, \frac{25}{6}, \frac{1}{10}, \frac{61}{30}, \frac{1}{10}, \frac{1}{10}, -\frac{1}{10}, \frac{29}{30}, \frac{1}{10}, \frac{1}{10}, -\frac{1}{10}, \frac{1}{10}, -\frac{1}{10}, -\frac{1}{10}, \frac{1}{10}, \frac{1}{2}, \frac{1}{30}, \frac{1}{30}, -\frac{1}{30}, \frac{1}{30}, -\frac{1}{30}, -\frac{1}{30}, \frac{1}{30}, \frac{1}{30}, -\frac{1}{30}, -\frac{1}{30}, \frac{1}{30}, -\frac{1}{30}, \frac{1}{30}, \frac{1}{30}, -\frac{1}{30}\right]$$

$$\left[\frac{29}{2}, \frac{253}{30}, \frac{25}{6}, \frac{1}{10}, \frac{61}{30}, \frac{1}{10}, \frac{1}{10}, -\frac{1}{10}, \frac{29}{30}, \frac{1}{10}, \frac{1}{10}, -\frac{1}{10}, \frac{1}{10}, -\frac{1}{10}, -\frac{1}{10}, \frac{1}{10}, \frac{1}{2}, \frac{1}{30}, \frac{1}{30}, -\frac{1}{30}, \frac{1}{30}, -\frac{1}{30}, -\frac{1}{30}, \frac{1}{30}, \frac{1}{30}, -\frac{1}{30}, -\frac{1}{30}, \frac{1}{30}, -\frac{1}{30}, \frac{1}{30}, \frac{1}{30}, -\frac{1}{30}\right] \quad \textbf{(4)}$$

Plot the 28-th Cesaro mean of order alpha=1/10 for f(x)=sin(2*PI*x) and the absolute difference between the function and the partial sum.

```
> f:=x->sin(2*Pi*x):
  n:=28:
  alpha:=1/10:
  fPlot:=plot(f(x),x=0..1,color=blue):
  CesaroPlot:=lineplot(CesaroMVal[alpha](n,f(x),floating),color=
  red):
  plots[display]([fPlot, CesaroPlot],tickmarks=[[seq(i/16=i/16,i=1.
  .16)],default]);
  plot(abs(sin(2*Pi*t)-CesaroMFunc[alpha](28,f(x),t)),t=0..1,
  tickmarks=[[seq(i/16=i/16,i=1..16)],default]);
```


1
0.5
0
-0.5
-1
1/16 1/8 3/16 1/4 5/16 3/8 7/16 1/2 9/16 5/8 11/16 3/4 13/16 7/8 15/16 1
x


0.10
0.06
0
1/16 1/8 3/16 1/4 5/16 3/8 7/16 1/2 9/16 5/8 11/16 3/4 13/16 7/8 15/16 1
t