1) g est impaire, donc $g(0) = 0$ et $g\left(-\frac{T}{2}\right) = -g\left(\frac{T}{2}\right)$.
Elle est aussi T-périodique, donc $g\left(-\frac{T}{2}\right) = g\left(\frac{T}{2}\right)$, finalement : $g\left(\frac{T}{2}\right) = 0$.

2) On écrit la procédure `B` à 3 paramètres `f,T,n` qui calcule une valeur approchée du coefficient b_n de la série de Fourier de g.

```
> B:=proc(f,T,n)
> local Tf,b,omega;
> Tf:=evalf(T);
> omega:=2*evalf(Pi)/Tf;
> b:=4/Tf*int(f(t)*sin(n*omega*t),t=0..Tf/2);
> b
> end;
```

$B := \mathbf{proc}(f, T, n)$
local Tf, b, ω;
$Tf := \text{evalf}(T)\,;\ \omega := 2 * \text{evalf}(\pi)/Tf\,;\ b := 4 * \text{int}(f(t) * \sin(n * \omega * t), t = 0..1/2 * Tf)/Tf\,;\ b$
end proc

3) On écrit les deux procédures `SomPart1` et `SomPart2` qui calculent la somme partielle Sn jusqu'au rang n de la série de Fourier de g ; respectivement en utilisant et sans utiliser la fonction Maple `sum`.
La seconde contient une simple boucle à la place de `sum`.

```
> SomPart1:=proc(f,T,n)
> local Tf,Sn,omega,k;
> Tf:=evalf(T);
> omega:=2*evalf(Pi)/Tf;
> Sn:=sum(B(f,T,k)*sin(k*omega*t),k=1..n);
> Sn
> end;
```

SomPart1 $:= \mathbf{proc}(f, T, n)$
local Tf, Sn, ω, k;
$Tf := \text{evalf}(T)\,;\ \omega := 2 * \text{evalf}(\pi)/Tf\,;\ Sn := \text{sum}(\text{B}(f, T, k) * \sin(k * \omega * t), k = 1..n)\,;\ Sn$
end proc

```
> SomPart2:=proc(f,T,n)
> local Tf,Sn,omega,k;
> Tf:=evalf(T);
> omega:=2*evalf(Pi)/Tf;
> Sn:=0;
> for k from 1 to n do Sn:=Sn+B(f,T,k)*sin(k*omega*t) od;
> Sn
> end;
```

SomPart2 $:= \mathbf{proc}(f, T, n)$
local Tf, Sn, ω, k;
$Tf := \text{evalf}(T)\,;$
$\omega := 2 * \text{evalf}(\pi)/Tf\,;$
$Sn := 0\,;$
for k **to** n **do** $Sn := Sn + \text{B}(f, T, k) * \sin(k * \omega * t)$ **end do**;
Sn
end proc

4) Pour écrire la procédure `g` qui détermine la valeur de g sur $\left[-\frac{3T}{4}, \frac{3T}{4}\right]$, il faut déterminer une formule valable sur chaque intervalle.

On a déjà les valeurs en $-\frac{T}{2}$, 0, $\frac{T}{2}$.
Par ailleurs, f et T sont ici des paramètres, la variable est t.

- Pour $t \in \left[-\frac{3T}{4}, -\frac{T}{2}\right[$, $\quad g(f,T,t) = f(t+T)$ par périodicité ;
- pour $t \in \left]-\frac{T}{2}, 0\right[$, $\quad g(f,T,t) = -f(-t)$ par imparité ;
- pour $t \in \left]0, \frac{T}{2}\right[$, $\quad g(f,T,t) = f(t)$ par définition ;
- pour $t \in \left]\frac{T}{2}, \frac{3T}{4}\right[$, $\quad g(f,T,t) = -f(-t+T)$ par périodicité et imparité.

```
> g:=proc(f,T,t)
> local Tf,y;
> Tf:=evalf(T);
> if t<-Tf/2 then y:=f(t+Tf) fi;
> if t=-Tf/2 then y:=0 fi;
> if t>-Tf/2 and t<0 then y:=-f(-t) fi;
> if t=0 then y:=0 fi;
> if t>0 and t<Tf/2 then y:=f(t) fi;
> if t=Tf/2 then y:=0 fi;
> if t>Tf/2 then y:=-f(-t+Tf) fi;
> y
> end;
```

$g := \mathbf{proc}(f, T, t)$
local Tf, y;
$Tf :=$ evalf(T) ;
if $t < -1/2 * Tf$ **then** $y := f(t + Tf)$ **end if** ;
if $t = -1/2 * Tf$ **then** $y := 0$ **end if** ;
if $-1/2 * Tf < t$ **and** $t < 0$ **then** $y := -f(-t)$ **end if** ;
if $t = 0$ **then** $y := 0$ **end if** ;
if $0 < t$ **and** $t < 1/2 * Tf$ **then** $y := f(t)$ **end if** ;
if $t = 1/2 * Tf$ **then** $y := 0$ **end if** ;
if $1/2 * Tf < t$ **then** $y := -f(-t + Tf)$ **end if** ;
y
end proc

5) `Dessine` est facile à écrire. Attention à ne pas oublier l'indication donnée.

```
> Dessine:=proc(f,T,n)
> local Tf;
> Tf:=evalf(T);
> plot({'g(f,T,t)',SomPart2(f,T,n)},t=-3*Tf/4..3*Tf/4)
> end;
```

Dessine $:= \mathbf{proc}(f, T, n)$
local Tf;
$Tf :=$ evalf(T) ; plot({SomPart2(f, T, n), 'g(f, T, t)'}, $t = -3/4 * Tf..3/4 * Tf$)
end proc

6) Pour `Rang`, qui calcule le premier rang n tel que :

$$\int_0^{T/2} \Big(f(t) - Sn(t)\Big)^2 \, \mathrm{d}t \leqslant e$$

on a besoin d'une boucle avec un `while`, qui tourne tant que la condition n'est pas vérifiée, c'est à dire jusqu'à ce qu'elle soit vérifiée !

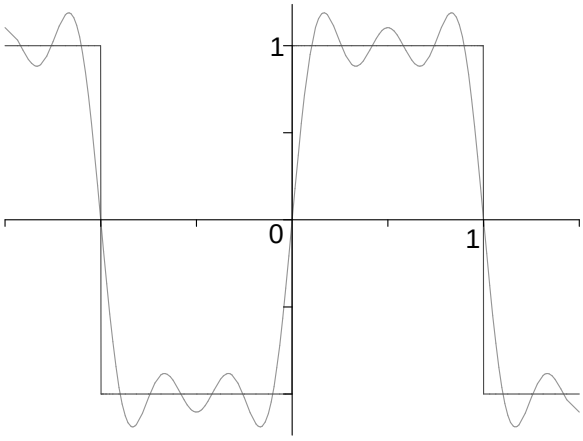
```
> Rang:=proc(f,T,e)
> local Tf,n,ef;
> Tf:=evalf(T);
> ef:=evalf(e);
> for n from 1 while int((f(t)-SomPart1(f,T,n))**2,t=0..Tf/2)>ef do od;
> n
> end;
```

$$
\begin{array}{l}
Rang := \mathbf{proc}(f, T, e) \\
\mathbf{local}\ Tf, n, ef; \\
\quad Tf := \mathrm{evalf}(T); \\
\quad ef := \mathrm{evalf}(e); \\
\quad \mathbf{for}\ n\ \mathbf{while}\ ef < \mathrm{int}((f(t) - \mathrm{SomPart1}(f, T, n))^2, t = 0..1/2 * Tf)\ \mathbf{do}\ \mathbf{end\ do}; \\
\quad n \\
\mathbf{end\ proc}
\end{array}
$$

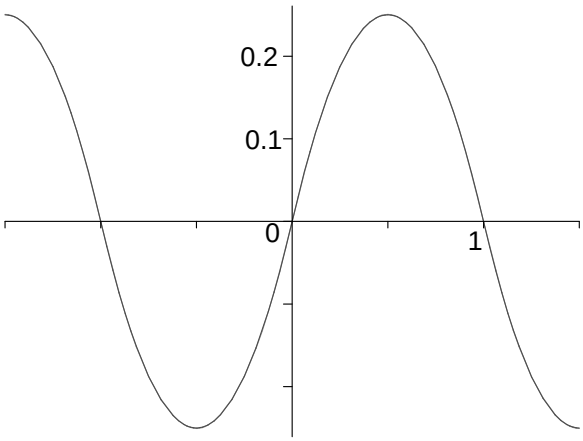
Remarquons que le corps de la boucle est vide !

7) Pour les graphes, on vous laisse voir :

```
> Dessine(t->1,2,5);
```




```
> Dessine(t->t*(1-t),2,5);
```



Pour `rang`, cela donne :

```
> Rang(t->1,2,0.05);
```

9

```
> Rang(t->t*(1-t),2,0.000001);
```

5