```
#################################################################################
#################################################################################
### File "pal_palev_palevv";
#################################################################################
#################################################################################

## Download the package "Flexion_operations";

## The bimould pal and its even factors palev and palevv are related to their mu-dilators
## dupal, dupalev, dupalevv by the relations:

## dur(pal)=vimu(pal)(dupal)+iden(dupal):
## dur(palev)=vimu(palev)(dupalev)+iden(dupalev):
## dur(palevv)=vimu(palevv)(dupalevv)+iden(dupalevv):

## The mu-dilators themselves are given by:

lan:=proc(r):  if r=1 then 1 else relu([1,seq(Paa,k=1..r-1)])(r) fi end:
##  lan(r)(u)(v)= add( (-1)^(j-1)*(r-1)!/(j-1)!/(r-j)!*ullj,j=1..r)/u1/u2.../ullr

serkol:=nn->convert(series(-1+t/(exp(t)-1), t=0, 31),polynom):
serkolev:=nn->convert(series(-1+t/2/tanh(t/2), t=0, 31),polynom);
serkolevv:=nn->convert(series(-1+t/(exp(t/2)-exp(-t/2)), t=0, 31),polynom):

kol:=n->coeff(serkol(n+3),t,n):
kolev:=proc(n): if (-1)^n=1 then kol(n) else 0 fi end:  ## kolev:=n->coeff(serkolev(n+3),t,n):
kolevv:=n->coeff(serkolevv(n+3),t,n):

dupal:=proc(r): kol(r)*lan(r)  end:
dupalev:=proc(r): kolev(r)*lan(r)  end:
dupalevv:=proc(r): kolevv(r)*lan(r)  end:

## This leads to the extremely efficient procedure:

pal:=proc(r): if r=1 then invdur(dupal)(1) else invdur((vimu(pal)(dupal)+iden(dupal)))(r) fi end:
palev:=proc(r):   if (-1)^r= -1 then 0 elif r=2 then invdur(dupalev)(2)
                 else invdur((vimu(palev)(dupalev)+iden(dupalev)))(r) fi end:
palevv:=proc(r): if  (-1)^r= -1  then 0 elif r=2 then invdur(dupalevv)(2)
                 else invdur((vimu(palevv)(dupalevv)+iden(dupalevv)))(r) fi end:
#################################################################################
#################################################################################
```