```
> with(bailey);
```

$$[alphadown, alphafind, alphaup, betadown, betafind, betaup] \quad (1)$$

```
> alphaup();
-----------------------------------------------------------
alphaup(a,q,alpha,n)
   This proc is used to move right in  a Bailey chain.
   In the Bailey chain
   .. -> (alpha[n],beta[n]) -> (alpha'[n],beta'[n]) ->
   this proc returns alpha'[n] given alpha.
   As usual we have assumed rho1,rho2=infinity.
-----------------------------------------------------------
> with(qseries):
> beta1:=(a,q,n)->1/aqprod(q,q,n):
> for j from 0 to 3 do print(alphafind(a,q,beta1,j)); od;
```

$$1$$

$$-\frac{a\,q\,(-1+a\,q^2)}{-1+q}$$

$$\frac{(-1+a\,q^4)\,q^5\,a^2\,(-1+a\,q)}{(-1+q)\,(-1+q^2)}$$

$$-\frac{(-1+a\,q^6)\,q^{12}\,a^3\,(-1+a\,q)\,(-1+a\,q^2)}{(-1+q)\,(-1+q^2)\,(-1+q^3)} \quad (2)$$

```
> for j from 0 to 3 do print(betadown(a,q,beta1,j)); od;
```

$$1$$

$$0$$

$$0$$

$$0 \quad (3)$$

```
> alpha1:=(a,q,n)->alphafind(a,q,beta1,n);
```

$$\alpha 1 := (a, q, n) \rightarrow alphafind(a, q, \beta 1, n) \quad (4)$$

```
> for j from 0 to 3 do print(betafind(a,q,alpha1,j));od;
```

$$1$$

$$-\frac{1}{-1+q}$$

$$\frac{1}{(-1+q)\,(-1+q^2)}$$

$$-\frac{1}{(-1+q)\,(-1+q^2)\,(-1+q^3)} \quad (5)$$

```
> qfactor(betafind(a,q,alpha1,4));
```

(6)

$$\frac{1}{(1-q)\,(1-q^2)\,(1-q^3)\,(1-q^4)} \tag{6}$$