Modular Forms Project: L-Functions of Quadratic Characters at Negative Integers

Henri Cohen,
Université de Bordeaux, Institut de Mathématiques de Bordeaux,
351 Cours de la Libération, 33405 TALENCE Cedex, FRANCE

July 8, 2018

1 Answers and Programs

Exercise 1.

(1). As mentioned in the example, since $M_4(\Gamma)$ is one-dimensional, $F_{2,D}$ is proportional to E_4, the constant of proportionality being $-L(\chi_D,-1)/48$, so for $n \ge 1$ the nth Fourier coefficient of $F_{2,D}$ is equal to $-5L(\chi_D,-1)\sigma_3(n)$, giving the required equality.

(2). For any integer m we can write uniquely $m = df_1^2$ with d squarefree. If $d \equiv 1 \pmod 4$ we take $(D,f) = (d,f_1)$. If $d \equiv 2,3 \pmod 4$ then if $m \equiv 0,1 \pmod 4$ we cannot have f_1 odd since otherwise $f_1^2 \equiv 1 \pmod 4$, so f_1 is even and we take $(D,f) = (4d, f_1/2)$.

We recall the Möbius inversion formula: if f and g are two arithmetic function and $h(n) = \sum_{d\mid n} f(d)g(n/d)$, in terms of Dirichlet series this can be written with evident notation $L(h,s) = L(f,s)L(g,s)$, so $L(g,s) = L(h,s)(L(f,s))^{-1}$. In our case $f(d) = d\left(\frac{D}{d}\right)$ is *totally multiplicative*, so $L(f,s) = \prod_p 1/(1-f(p))$, hence $L(f,s)^{-1} = \prod_p(1-f(p)) = \sum_{n\ge1}\mu(n)f(n)$, so we deduce that

$$g(n) = \sum_{d\mid n}\mu(d)f(d)h(n/d)\ .$$

Applying to our specific case $f(n) = n\left(\frac{D}{n}\right)$, $g(n) = S_2(n^2D)$, and $h(n) = -5L(\chi_D,-1)\sigma_3(n)$ we obtain finally, with $m = Df^2$:

$$S_2(m) = -5L(\chi_D,-1)\sum_{d\mid f}\mu(d)d\left(\frac{D}{d}\right)\sigma_3\left(\frac{f}{d}\right)\ .$$

(3). A small computation (better done on a computer) shows for instance that $F(n)/\sigma_1(n)$ is equal to $[23/12, 29/12, 35/12, 41/12, 47/12]$ for $n = 4$, 5, 6, 7, 8, so we can reasonably conjecture that $F(n) = (n/2-1/12)\sigma_1(n)$, giving the formula

$$\sum_{d\mid n} dS_2((n/d)^2) = (5/12)\sigma_3(n) - (n/2-1/12)\sigma_1(n)\ .$$

(4). As before, we deduce that

$$S_2(f^2) = (5/12)\sum_{d\mid f}\mu(d)d\sigma_3(f/d) - \sum_{d\mid f}\mu(d)(f/2-d/12)\sigma_1(f/d)\ .$$

Both terms can in fact be simplified. We do not want to simplify the first so as to leave it in a form similar to the case of general m. For the second, we set $T(f) = \sum_{d\mid f}\mu(d)(f/2-d/12)\sigma_1(f/d)$.

$$\sum_{n\geq 1}\mu(n)(f/2-n/12)/n^s = (f/2)\zeta(s)^{-1} - (1/12)\zeta(s-1)^{-1}\ ,$$

while $\sum_{n\geq 1}\sigma_1(n)/n^s = \zeta(s)\zeta(s-1)$. It follows that $T(f) = (f/2)f - (1/12) = f^2/2 - 1/12$. Thus the general formula for $m = Df^2$ with D fundamental including $D = 1$ is

$$S_2(m) = -5L(\chi_D,-1)\sum_{d\mid f}\mu(d)d\left(\frac{D}{d}\right)\sigma_3\left(\frac{f}{d}\right) - \delta(\sqrt{m})\left(\frac{m}{2}-\frac{1}{12}\right)\ .$$

Exercise 2.

(1). (a). $k=4$: $M_8(\Gamma)$ is one-dimensional generated by $E_8 = E_4^2 = 1 + 480q + O(q^2)$. On the other hand,

$$F_{4,D}(\tau) = L(\chi_D,-3)/480 + c_{4,D}(1)q + O(q^2)\ .$$

It follows that

$$L(\chi_D,-3) = c_{4,D}(1) = \sum_{s\in\mathbb{Z}}\sigma_3\left(\frac{D-s^2}{4}\right)\ .$$

(1). (b). $k = 6$, 8, and 10: in these case $M_{2k}(\Gamma)$ is two-dimensional generated by $(F,G) = (E_6^2,\Delta)$, $(E_8^2,\Delta E_4)$, and $(E_{10}^2,\Delta E_8)$ respectively, while

$$F_{k,D}(\tau) = -(B_k/(4k))L(\chi_D,1-k) + c_{k,D}(1)q + c_{k,D}(2)q^2 + O(q^3)\ .$$

Since the three forms $(F,G,F_{k,D})$ belong to a vector space of dimension 2, the 3×3 matrix M of their first three coefficients vanish, i.e., if we set $F = 1 + aq + bq^2 + O(q^3)$ and $G = q + eq^2 + O(q^3)$ we have

$$M = \begin{pmatrix} -(B_k/(4k))L(\chi_D,1-k) & c_{k,D}(1) & c_{k,D}(2) \\ 1 & a & b \\ 0 & 1 & e \end{pmatrix}$$

and $\det(M) = 0$, so expanding along the first row gives

$$-(B_k/4k)L(\chi_D,1-k)(ae-b) - ec_{k,D}(1) + c_{k,D}(2) = 0\ ,$$

in other words

$$L(\chi_D,1-k) = -((4k/B_k)/(ae-b))(ec_{k,D}(1) - c_{k,D}(2))\ .$$

For $k = 6, 8, 10$, we have $(4k/B_k, a, b, e) = (1008, -1008, 220752, -24)$, $(-960, 960, 354240, 216)$, $(528, -528, -201168, 456)$, giving the following formulas:

$$\begin{aligned}
L(\chi_D, -5) &= -\frac{1}{195}(24c_{6,D}(1) + c_{6,D}(2)) \\
L(\chi_D, -7) &= -\frac{1}{153}(216c_{8,D}(1) - c_{8,D}(2)) \\
L(\chi_D, -9) &= \frac{1}{75}(456c_{10,D}(1) - c_{10,D}(2)) ,
\end{aligned}$$

where

$$\begin{aligned}
c_{k,D}(1) &= \sum_{s\in\mathbb{Z}} \sigma_{k-1}\left(\frac{D-s^2}{4}\right) \\
c_{k,D}(2) &= \sum_{s\in\mathbb{Z}} \sigma_{k-1}(D-s^2) + \left(\frac{D}{2}\right)2^{k-1}\sum_{s\in\mathbb{Z}} \sigma_{k-1}\left(\frac{D-s^2}{4}\right) ,
\end{aligned}$$

or, using the notation $S_k(m, N)$, and combining with the previous formulas:

$$\begin{aligned}
L(\chi_D, -1) &= -\frac{1}{5}S_2(D, 4) \\
L(\chi_D, -3) &= S_4(D, 4) \\
L(\chi_D, -5) &= -\frac{1}{195}\left(\left(24 + 2^5\left(\frac{D}{2}\right)\right)S_6(D, 4) + S_6(D, 1)\right) \\
L(\chi_D, -7) &= -\frac{1}{153}\left(\left(216 - 2^7\left(\frac{D}{2}\right)\right)S_8(D, 4) - S_8(D, 1)\right) \\
L(\chi_D, -9) &= \frac{1}{75}\left(\left(456 - 2^9\left(\frac{D}{2}\right)\right)S_{10}(D, 4) - S_{10}(D, 1)\right) .
\end{aligned}$$

(2). Since for $k \geq 4$ even E_k is an eigenfunction of the Hecke operators $T(n)$ with eigenvalues $\sigma_{k-1}(n)$, these eigenvalues satisfy the same relations as the Hecke operators themselves, and in particular

$$\sigma_{k-1}(4)\sigma_{k-1}(n) = \sum_{d|4} d^{k-1}\sigma_{k-1}(4n/d^2) ,$$

in other words

$$(4^{k-1} + 2^{k-1} + 1)\sigma_{k-1}(n) = \sigma_{k-1}(4n) + 2^{k-1}\sigma_{k-1}(n) + 4^{k-1}\sigma_{k-1}(n/4) ,$$

giving the required relation.

(3). We prove more generally that $\sigma_z(n)$ satisfies the usual Hecke relations for any complex z. We write

$$\sigma_z(m)\sigma_z(n) = \sum_{d|m,\ e|n} (de)^z = \sum_{D|mn} D^z \sum_{e|\gcd(D,n),\ (D/e)|m} 1 .$$

Now $(D/e) \mid m$ is equivalent to $D \mid em$ hence to $D/\gcd(D, m) \mid e$. Thus the number of $e \mid \gcd(D, n)$ such that $D/\gcd(D, m) \mid e$ is equal to the number of f dividing $\gcd(D, m)\gcd(D, n)/D = \gcd(D, m, n, mn/D)$. Thus,

$$\sigma_z(m)\sigma_z(n) = \sum_{D|mn} D^z \sum_{d|\gcd(D,m,n,mn/D)} 1 ,$$

so writing $D=fd$ we have $d\mid mn/(fd)$, in other words $f\mid mn/d^2$, so

$$\sigma_z(m)\sigma_z(n)=\sum_{d\mid\gcd(m,n)}d^z\sum_{f\mid mn/d^2}f^z=\sum_{d\mid\gcd(m,n)}d^z\sigma_z(mn/d^2)\;,$$

which is the desired relation.

(4). We have

$$S_k(D,1)=\sum_{s\not\equiv D\pmod 2}\sigma_{k-1}(D-s^2)+T$$

with

$$T=\sum_{s\equiv D\pmod 2}\sigma_{k-1}(4(D-s^2)/4)=(4^{k-1}+1)S_k(D,4)-4^{k-1}S_k(D,16)$$

by (2) and (3), as claimed.

Exercise 3.

(1). We first write the following small `Pari/GP` programs:

```
mysigma(n,k=1)=if(denominator(n)>1 || n<=0,0,sigma(n,k));
S(k,m,N)=sum(s=-sqrtint(m),sqrtint(m),mysigma((m-s^2)/N,k-1));
LF(k,D)=bestappr(lfun(D,1-k));
```

Of course these programs can be trivially improved, but since we are going to perform very small experiments, this is not necessary. Note that $L(\chi_D,1-k)$ always has a very small denominator, so we use the `bestappr` function to express the result as a rational number.

We can already test the example given in the exercise by writing

```
for(D=5,50,if(isfundamental(D)&&(D%3)==0,print(D,": ",LF(2,D)/S(2,D,12))))
```

which answers only -2, and similarly with `(D%3)==1` which answers only -1.

We are now going to explore systematically possible formulas for a given N and t. For this, we can proceed as follows (of course this is not the only way).

```
/* List of $p\mid N$ or $p\le t$. */
allprimes(N,t)=
{
  V=factor(N)[,1];
  forprime(p=2,t,if(N%p,V=concat(V,p)));
  return (vecsort(V));
}
/* given L a list of -1,0,1 for each p in V, test if D is OK */
isdok(V,L,D)=
{
  for(i=1,#V,if(kronecker(D,V[i])!=L[i],return(0)));
  return(1);
}
/* make a list of $24+t$ suitable D */
makedlist(V,L,t)=
```

```
{
  my(res=vector(24+t),ct=0);
  forstep(D=5,oo,[3,1],
    if(isfundamental(D) && isdok(V,L,D),ct++;res[ct]=D);
    if(ct==24+t,break())
  );
  return(res);
}
```

After executing all these programs, we have a list of $t+24$ fundamental discriminants satisfying the required Legendre symbol conditions.

We now come to the heart of the computation. To find the c_n we must simply solve the linear system $\sum_{1\le n\le t} c_n S_k(n^2D, N) = L(\chi_D, 1-k)$. To have good confidence, we will want a largely overdetermined system, and this is why we chose $t+24$ discriminants. We thus write the following:

```
/* V: primes, L: Kronecker symbols */
findformula0(k,V,L,N,t)=
{
  my(dlist,ld,M,B,res);
  dlist=makedlist(V,L,t); ld=#dlist; /* t+24 in fact. */
  M=matrix(ld,t);
  for(n=1,t,
    M[,n]=vectorv(ld,i,S(k,n^2*dlist[i],N))
  );
  B=vectorv(ld,i,LF(k,dlist[i]));
  res=matinverseimage(M,B);
  return(res);
}
```

One comment: one cannot use `matsolve` since the matrix M may have a nontrivial kernel. In that case, if the result `res` is nonempty this means that there are several solutions to our system, but the program only returns one.

The main program to exploit this is the following:

```
findformula1(k,N,t)=
{
  my(V,nb,res);
  V=allprimes(N,t); nb=#V;
  forvec(L=vector(nb,j,[-1,1]),
    res=findformula0(k,V,L,N,t);
    if(#res,print([V,L],": ",res))
  );
}
```

We now test this program for small N (and initially only $t=1$): Writing `findformula1(2,4,1)` outputs:

```
[[2]~, [-1]]: [-1/5]~
[[2]~, [0]]: [-1/5]~
[[2]~, [1]]: [-1/5]~
```

which simply means that the result is independent of $\left(\frac{D}{2}\right)$ and recovers our formula $L(\chi_D,-1)=(-1/5)S_2(D,4)$.

Writing `findformula1(2,8,1)` outputs:

```
[[2]~, [0]]: [-1]~
[[2]~, [1]]: [-1/2]~
```

This says that if $\left(\frac{D}{2}\right)=0$ we have $L(\chi_D,-1)=-S_2(D,8)$ and if $\left(\frac{D}{2}\right)=1$ we have $L(\chi_D,-1)=(-1/2)S_2(D,8)$.

If we write `findformula1(2,12,1)` the output is:

```
[[2, 3]~, [-1, 0]]: [-2]~
[[2, 3]~, [-1, 1]]: [-1]~
[[2, 3]~, [0, 0]]: [-2]~
[[2, 3]~, [0, 1]]: [-1]~
[[2, 3]~, [1, 0]]: [-2]~
[[2, 3]~, [1, 1]]: [-1]~
```

This shows that the coefficients do not depend on $\left(\frac{D}{2}\right)$, and we recover what we found above: if $\left(\frac{D}{3}\right)=0$ we have $L(\chi_D,-1)=-2S_2(D,12)$, and if $\left(\frac{D}{3}\right)=1$ we have $L(\chi_D,0)=-S_2(D,12)$.

If we write `findformula1(2,16,1)` the output is:

```
[[2]~, [1]]: [-2]~
```

So if $\left(\frac{D}{2}\right)=1$ we have $L(\chi_D,-1)=-2S_2(D,12)$.

Trying $N=20$, 24, etc... with $t=1$ does not give any formula, so we must now try $t=2$. Writing `findformula1(2,20,2)` outputs

```
[[2, 5]~, [-1, 0]]: [-4/5, -2/5]~
[[2, 5]~, [-1, 1]]: [-2/5, -1/5]~
[[2, 5]~, [0, 0]]: [-8/5, -2/5]~
[[2, 5]~, [0, 1]]: [-4/5, -1/5]~
[[2, 5]~, [1, 0]]: [-12/5, -2/5]~
[[2, 5]~, [1, 1]]: [-6/5, -1/5]~
```

It is easy to put this into a single formula: for $\left(\frac{D}{5}\right)=0$ or 1 we have

$$L(\chi_D,-1)=-(1/5)(2-\left(\tfrac{D}{5}\right))(S_2(4D,20)+2(2+\left(\tfrac{D}{2}\right))S_2(D,20))$$

and so on. We can thus accumulate a large number of (experimental) formulas. Note, however, that some of them are duplicates. For instance if you write `findformula1(2,32,2)` you will obtain

```
[[2]~, [0]]: [0, -1]~
[[2]~, [1]]: [0, -1/2]~
```

but because of the 0 coefficient of $S_2(D,32)$ and the fact that $S_2(4D,32)=S_2(D,8)$, these formulas are identical to those that we obtained for $N=8$. Thus we can add a little additional condition: if $16\mid N$ and $c_n=0$ for all n odd, the formula is equivalent to one for $N/4$, so we do not keep it. This corresponds to adding at the end of `findformula1`, just before the final `return`, the following:

```
  if (#res && N%16==0,
    forstep(n=1,t,2,if(res[n],return(res)));
    return([]~)
  );
```

(the same can be done if some $f^2 \mid N/4$ and $c_n = 0$ unless $f \mid n$).

(2). (a). Since Legendre symbols are independent, the probability is easily computed: it is the product over all primes p of the corresponding quantity. Assume first that $p > 2$. Fundamental discriminants D are distributed equally among the p^2-1 nonzero residues modulo p^2. For $p-1$ of these residues we have $\left(\frac{D}{p}\right) = 0$ (i.e., $p \mid D$), and the others are equally distributed between residues and nonresidues. Thus the respective probabilities for $\left(\frac{D}{p}\right) = (-1, 0, 1)$ are $(p/(2(p+1)), 1/(p+1), p/(2(p+1)))$.

Assume now that $p = 2$. We must now look modulo 16: fundamental discriminants are congruent to 1, 5, 8, 9, 12, and 13 modulo 16 with equal probability, so $\left(\frac{D}{2}\right) = (-1, 0, 1)$ with respective probabilities $(1/3, 1/3, 1/3)$, which happens to be exactly the same formula as for $p > 2$.

(2). (b). We must estimate the cost of computing $S_k(m, N)$ when $4 \mid N$. Since we can trivially group together s and $-s$, this is the number of $s \le m^{1/2}$ such that $s^2 \equiv m \pmod{N}$. As for probabilities, this is multiplicative, so we may assume that $N = p^a$ for some prime p. We write $m = p^b m'$ with $p \nmid m'$. If $b \ge a$ the condition is $s^2 \equiv 0 \pmod{p^a}$, so $s \equiv 0 \pmod{p^{\lceil a/2 \rceil a/2}}$, and there are $p^{\lfloor a/2 \rfloor}$ such values of s modulo N.

If $b < a$ we must have $v_p(s) = b/2$, so if b is odd the congruence is impossible. On the other hand, if b is even we write $s = p^{b/2}s'$ and the congruence is ${s'}^2 \equiv m' \pmod{p^{a-b}}$.

If $p > 2$ we must have $\left(\frac{m'}{p}\right) = 1$, and in that case there exist exactly 2 solutions modulo p, hence modulo p^{a-b} by Hensel's lemma, hence always $1 + \left(\frac{m'}{p}\right)$ values of s' modulo p^{a-b}, hence $p^b(1 + \left(\frac{m'}{p}\right))$ values of s modulo p^a.

If $p = 2$ and $a - b \ge 3$ the same argument applies: we must have $m' \equiv 1 \pmod{8}$, in which case there are 4 solutions modulo 8, hence by Hensel's lemma also modulo p^{a-b}, hence $p^{b+1}(1 + \left(\frac{m'}{2}\right))$ values of s modulo p^a.

If $p = 2$ and $a - b = 2$ we have no solutions if $m' \equiv 3 \pmod{4}$, and 2 if $m' \equiv 1 \pmod{4}$, so $p^b(1 + \left(\frac{-4}{m'}\right))$ solutions modulo p^a.

Finally, if $p = 2$ and $a - b = 1$ we have one solution, so p^b solutions modulo p^a.

This analysis leads to the following program:

```
/* Individual cost function: given m, N, approx number of s<m^(1/2)
such that s^2=m mod N, expressed as m^{1/2}/M for some rational M,
or 0 if none exist. */
cost1(m,N)=
{
  my(fa=factor(N),lipr=fa[,1],liex=fa[,2],M=1);
  for(i=1,#lipr,
    my(p=lipr[i],a=liex[i],b=valuation(m,p),mp=m\p^b);
    if(b>=a,M*=p^(ceil(a/2));next());
```

```
    if(b%2,return(0));
    if(p>2,
      if(kronecker(mp,p)==-1,return(0));
      M*=p^(a-b/2)/2; next()
    );
/* Here p=2 */
    if(a-b>=3,M*=2^(a-b/2-2);next());
    if(a-b==2,M*=2^(a-b/2-1);next());
    if(a-b==1,M*=2^(a-b/2);next())
  );
  return (M);
}
```

We will apply this program to $m = n^2D$, and since we want to express the cost as a multiple of $D^{1/2}$, we will need to divide by n the output M of this program for the result to be of the form $D^{1/2}/M$.

Now a formula for a given N and t is represented by two vectors: first the vector P of pairs $(p, \left(\frac{D}{p}\right))$ which are requested. Note that, as the examples above show, this vector can be shorter than the pair used by the program when the coefficients do not depend on some of the Legendre symbols. And second, the vector of coefficients $C = (c_1, \ldots, c_t)$. The only property that we need on these coefficients is whether they are zero or nonzero, since if they vanish we of course do not need to compute the corresponding sum.

We can thus write the following programs:

```
/* Given D, N, and a vector of coefficients C=(c_1,...,c_t),
determine the cost of the computation using the
corresponding formula, again in the form D^{1/2}/M.
Again return 0 if something goes wrong. */
cost2(D,N,C)=
{
  my(t=#C,S=0,c);
  for(n=1,t,
    if(C[n],
       c=cost1(n^2*D,N);
       if(!c,return(0),S+=n/c)
    )
  );
  return(1/S);
}

cost3(P,N,C)=
{
  my(t=#C,ll=#P,V,L,dlist,CO,M);
  V=vector(ll,i,P[i][1]); /* list of primes. */
  L=vector(ll,i,P[i][2]); /* list of required Legendre symbols. */
  dlist=makedlist(V,L,t);
  CO=vector(#dlist,i,cost2(dlist[i],N,C));
  M=vecmax(CO);
  if(vecmin(CO)==M,return(M),return(0));
```

```
}
```

This program computes the cost for the $t+24$ suitable discriminants. Three things may theoretically happen. The first is that one of the cost functions returns 0, which is an error, and shows simply that the formula is wrong. The second is that some costs are different (and nonzero). In practice this does not happen, but there is no reason why it shouldn't. If this does happen, this means that the cost depends on some extra Legendre symbols, so some additional code must be written. The third is that all costs are equal, in which case we are happy.

Finally, we must write a program which computes the probabilities. This is much simpler since it depends only on the vector P of pairs $(p,(D/p))$:

```
/* Compute the probability of P among fundamental discriminants. */
prob(P)=
{
  my(S=1,p);
  for(i=1,#P,
    p=P[i][1];
    S/=2*(p+1); if(P[i][2],S*=p,S*=2)
  );
  return(S);
}
```

We now have written the necessary programs, and to present them in a suitable manner we will simply modify the output of the basic `findformula1` program as follows:

```
findformula2(k,N,t)=
{
  my(V,nb,res);
  V=allprimes(N,t); nb=#V;
  forvec(L=vector(nb,j,[-1,1]),
    res=findformula0(k,V,L,N,t);
    if (#res,
      P=vector(nb,j,[V[j],L[j]]);
      print(P,": ",res,", ",[cost3(P,N,res),prob(P)])
    )
  );
}
```

We can now apply this program for successive values of N (and the smallest t giving formulas) and make a table of results. There is of course some hand-tayloring which must be done, first to remove useless formulas, but also to group together similar formulas as we have done above. For instance, let us write `findformula2(2,28,2)`. The output is

```
[[2, -1], [7, 0]]: [1, -1]~, [14/3, 1/24]
[[2, -1], [7, 1]]: [1/2, -1/2]~, [7/3, 7/48]
[[2, 0], [7, 0]]: [-1, -1]~, [14/3, 1/24]
[[2, 0], [7, 1]]: [-1/2, -1/2]~, [7/3, 7/48]
```

```
[[2, 1], [7, 0]]: [-3, -1]~, [14/3, 1/24]
[[2, 1], [7, 1]]: [-3/2, -1/2]~, [7/3, 7/48]
```

When $\left(\frac{D}{7}\right) = 0$, the coefficients (c_n) are clearly $(-1-2\left(\frac{D}{2}\right), -1)$ for the same cost 14/3, and a similar simplification can be done when $\left(\frac{D}{7}\right) = 1$, so grouping them together would read

```
[[7, 0]]: [-1 -2(D/2), -1]~, [14/3, 1/8]
[[7, 1]]: [-1/2 -(D/2), -1/2]~, [7/3, 7/16]
```

As a second example, consider the output of `findformula2(2,40,2)`:

```
[[2, 0], [5, 0]]: [0, -2]~, [10, 1/18]
[[2, 0], [5, 1]]: [0, -1]~, [5, 5/36]
[[2, 1], [5, 0]]: [-2, -2]~, [5, 1/18]
[[2, 1], [5, 1]]: [-1, -1]~, [5/2, 5/36]
```

Consider for instance the case $[[2,1],[5,0]]$: this has cost $D^{1/2}/5$ (and probability 1/18). However `findformula2(2,80,2)` outputs:

```
[[2, 1], [5, 0]]: [4/3, -16/3]~, [20/3, 1/18]
[[2, 1], [5, 1]]: [2/3, -8/3]~, [10/3, 5/36]
```

Here $[[2,1],[5,0]]$ has cost $D^{1/2}/(20/3)$ (and of course the same probability), which is smaller, so we can remove the formula obtained for $N = 40$. In particular, since our initial formula has cost $D^{1/2}/2$ and no Legendre symbol condition, we can remove all the formulas having larger or equal cost. For instance, in the $N = 28$ example above, $[[2,0],[7,1]]$ with $M = 7/3$ is superseded by $[[2,0]]$ for $N = 8$ with $M = 4$, and $[[2,1],[7,1]]$ with $M = 7/3$ is superseded by $[[2,1]]$ for $N = 16$ with $M = 4$, so we only keep $[[2,-1],[7,1]]$.

In view of these remarks, and using simply the formulas obtained for $N \leq 16$, we can write the following improvement to `findformula2`:

```
/* Only for k=2. */
findformula32(N,t)=
{
  my(V,nb,res);
  V=allprimes(N,t); nb=#V;
  forvec(L=vector(nb,j,[-1,1]),
    res=findformula0(2,V,L,N,t);
    if (#res,
      P=vector(nb,j,[V[j],L[j]]);
      c=cost3(P,N,res);
      if(N>16,
        if(c<=2,next());
        if(L[1]>=0 && c<=4,next()); /* case (D/2)=0 or 1. */
        if(V[2]==3 && L[2]==0 && c<=6,next()); /* case (D/3)=0. */
        if(V[2]==3 && L[2]==1 && c<=3,next()); /* case (D/3)=1. */
      );
      print(P,": ",res,", ",[c,prob(P)]),
    )
  );
}
```

Using these programs and hand simplifications, we now construct the desired table:

N	P	$L(\chi_D,-1)$	M	prob.
4	–	$(-1/5)$	2	1
8	$(2,0)$	(-1)	4	1/3
12	$(3,0)$	(-2)	6	1/4
12	$(3,1)$	(-1)	3	3/8
16	$(2,1)$	(-2)	4	1/3
20	$(2,-1),(5,0)$	$(-4/5,-2/5)$	10/3	1/18
28	$(7,0)$	$(-1-2(D/2),-1)$	14/3	1/8
28	$(2,-1),(7,1)$	$(1/2,-1/2)$	7/3	7/48
36	$(2,0),(3,1)$	$(0,-1)$	9/2	1/8
40	$(2,0),(5,0)$	$(0,-2)$	10	1/18
40	$(2,0),(5,1)$	$(0,-1)$	5	5/36
44	$(2,-1),(11,0)$	$(22-6(D/3),-8,-2)/5$	11/3	1/36
52	$(13,0)$	$(8-12(D/2)-6(D/3),-2,-2/3)$	13/3	1/14
68	$(2,-1),(17,0)$	$(13/2-3(D/3),-5/4,-1,-1/4)$	17/5	1/54
72	$(2,-1),(3,1)$	$(0,0,0,-1/3)$	9/2	1/8
76	$(2,-1),(19,0)$	$(32-18(D/3),-4,-6,-2)/5$	19/5	1/60
80	$(2,-1),(5,0)$	$(4/3,-16/3)$	20/3	1/18
84	$(3,0),(7,0)$	$(10-8(D/2),-4,-2)$	7	1/32
84	$(3,1),(7,0)$	$(2-4(D/2),-2,-2)$	14/3	3/64
88	$(2,0),(11,0)$	$(16-12(D/2),-4,-4,-2)/3$	22/5	1/36
92	$(2,-1),(23,0)$	$(5(10-6(D/3)-(D/5)),-2,-10,-4,-1)/7$	46/15	1/72
100	$(2,-1),(3,-1),(5,1)$	$(6,1/2,-1,-1/2)$	5/2	5/96
104	$(2,0),(13,0)$	$(8-6(D/3),-1,-2,-1)$	26/5	1/42
120	$(2,-1),(3,1),(5,0)$	$(0,0,0,-1/3,0,-1/3)$	30/7	1/48

(3). In view of this table, a reasonable algorithm is the following:

(1) If $D\equiv 0 \pmod 3$, $L(\chi_D,-1)=-2S_2(D,12)$, cost $D^{1/2}/6$, happens with probability 1/4.

(2) Otherwise, if $D\equiv 4 \pmod{12}$, $L(\chi_D,-1)=-S_2(D,9)$, cost $D^{1/2}/(9/2)$, happens with remaining probability 1/8.

(3) Otherwise, if $D\equiv 13 \pmod{24}$, $L(\chi_D,-1)=-S_2(2D,9)/3$, cost $D^{1/2}/(9/2)$, happens with remaining probability 1/8.

(4) Otherwise, if $D\equiv 0 \pmod 4$, $L(\chi_D,-1)=-S_2(D,8)$, cost $D^{1/2}/4$, happens with remaining probability 1/8.

(5) Otherwise, if $D\equiv 1 \pmod 8$, $L(\chi_D,-1)=-2S_2(D,16)$, cost $D^{1/2}/4$, happens with remaining probability 1/4.

(6) Otherwise $L(\chi_D,-1)=(-1/5)S_2(D,4)$, cost $D^{1/2}/2$, happens with remaining probability 1/8.

Note that after Step 5, the next best formula is $L(\chi_D,-1)=-S_2(D,12)$ when $D\equiv 1 \pmod 3$, but the reader can check that the case $D\equiv 1 \pmod 3$ is already entirely covered by the preceding ones.

The average cost of this algorithm is

$$D^{1/2}(1/24+1/36+1/36+1/32+1/16+1/16) = (73/288)D^{1/2} \approx D^{1/2}/3.945\ ,$$

so it is in average approximately twice as fast as the use of the single formula $L(\chi_D,-1) = (-1/5)S_2(D,4)$.

Exercise 4.

(1). As mentioned, the only formula that I have found is $L(\chi_D,-3) = S_4(D,4)$ with cost $D^{1/2}/2$ and probability 1.

(2). For $k=6$, we can make a table analogous (but much smaller) to that for $k=2$:

N	P	$L(\chi_D,-5)$	M	prob.
4	–	$(-24-32(D/2),-1)/195$	$2/3$	1
8	$(2,0)$	$(40,-1)/3$	$4/3$	$1/3$
12	$(2,-1),(3,0)$	$(1077,-12,-1)/26$	1	$1/12$
16	$(2,1)$	$(128,-117)/111$	$4/3$	$1/3$
20	$(2,-1),(5,0)$	$(352-5832(D/3),152,-24,-2)/75$	1	$1/18$
28	$(2,-1),(7,0)$	$(-(19294+9477(D/3)+3125(D/5)),884,-39,-12,-1)/84$	$14/15$	$1/24$
60	$(2,-1),(3,0),(5,0)$	$(-(65230+16807(D/7)),2354,447,-12,-19,-6,-1)/42$	$15/14$	$1/72$

(we have omitted a few formulas whose gain would have been negligible).

In view of this table, a reasonable algorithm is as follows:

(1) If $D \equiv 0 \pmod 4$, $L(\chi_D,-5) = (40S_6(D,8)-S_6(D,2))/3$, cost $D^{1/2}/(4/3)$, happens with probability 1/3.

(2) If $D \equiv 1 \pmod 8$, $L(\chi_D,-5) = (128S_6(D,16)-117S_6(D,4))/111$, cost $D^{1/2}/(4/3)$, happens with probability 1/3.

(3) Otherwise, if $D \equiv 21 \pmod{24}$, $L(\chi_D,-5) = (1077S_6(D,12)-12S_6(D,3)-S_6(3D,4))/26$, cost $D^{1/2}$, happens with remaining probability 1/9.

(4) Otherwise $L(\chi_D,-5) = -((24+32(D/2))S_6(D,4)+S_6(D,1))/195$, cost $D^{1/2}/(2/3)$, happens with remaining probability 2/9.

The average cost of this algorithm is

$$D^{1/2}(1/4+1/4+1/9+1/3) = (17/18)D^{1/2} \approx D^{1/2}/1.059\cdots\ ,$$

almost 60% faster than the use of the first formula alone.

(3). For $k=8$ and $k=10$ we give directly the tables:

N	P	$L(\chi_D,-7)$	M	prob.
4	–	$(-(216-128(D/2)),1)/153$	$2/3$	1
12	$(3,0)$	$(-(621+1536(D/2)),-12,1)/10$	1	$1/4$

N	P	$L(\chi_D,-9)$	M	prob.
4	–	$(456-512(D/2),-1)/75$	$2/3$	1

Exercise 5.

(1). (a). We need to rewrite from scratch all the programs that we have written for k even, but of course this will mostly be copy-paste so quite easy.

```
mysigma1(n,k=2)=
{
  if(denominator(n)>1 || n<=0,return(0));
  n>>=valuation(n,2); /* make n odd */
  sumdiv(n,d,if(d%4==1,d^k,-d^k));
}
mysigma2(n,k=2)=
{
  my(v);
  if(denominator(n)>1 || n<=0,return(0));
  v=valuation(n,2); n>>v; /* make n odd */
  sumdiv(n,d,if(d%4==1,(n/d)^k,-(n/d)^k))<<(k*v);
}
S1(k,m,N)=sum(s=-sqrtint(m),sqrtint(m),mysigma1((m-s^2)/N,k-1));
S2(k,m,N)=sum(s=-sqrtint(m),sqrtint(m),mysigma2((m-s^2)/N,k-1));
```

We first check the example given in the exercise:

```
forstep(D=-3,-50,-1,if(isfundamental(D)&&(D%8==1),\
  print1(LF(3,D)/S1(3,abs(D),2)," ")))
```

This indeed outputs only 1/7.

The main program to be modified is the program `findformula0`. Since we restrict to the simple case where $V = [2]$ and $L = [-1]$, $[0]$, or $[1]$, the program is a little simpler:

```
/* l=-1,0,or 1 */
makedlistneg(l,t)=
{
  my(res=vector(24+t),ct=0);
  forstep(D=-3,-oo,[-1,-3],
    if(isfundamental(D) && kronecker(D,2)==l,ct++;res[ct]=D);
    if(ct==24+t,break())
  );
  return(res);
}
findformulaneg0(k,l,N,t)=
{
  my(dlist,dlist4,ld,M,B,res);
  dlist=makedlistneg(l,t); ld=#dlist;
  dlist4=if(l==0,dlist/4,dlist); dlist4=apply(abs,dlist4);
  M=matrix(ld,2*t);
  for(n=1,t,
    M[,2*n-1]=vectorv(ld,i,S1(k,n^2*dlist4[i],N));
    M[,2*n]=vectorv(ld,i,S2(k,n^2*dlist4[i],N));
  );
  B=vectorv(ld,i,LF(k,dlist[i]));
  res=matinverseimage(M,B);
  return(res);
}
```

When $D\equiv 0\pmod 4$ it is more efficient to look directly for formulas involving $D/4$, although these formulas would be found anyway but later, explaining the use of the variable `dlist4`.

The cost function and driver program are immediate to write:

```
cost3neg(l,N,C)=
{
  my(P=[[2,l]],lc=#C,Ceven,Codd,Meven,Modd,S);
  Ceven=vector(lc\2,j,C[2*j]);
  Codd=vector((lc+1)\2,j,C[2*j-1]);
  Meven=if(Ceven,cost3(P,N,Ceven),0);
  Modd=if(Codd,cost3(P,N,Codd),0);
  S=0; if(Meven,S+=1/Meven); if(Modd,S+=1/Modd);
  if(l==0,S/=2); /* since (D/4)^{1/2} instead of D^{1/2}. */
  return (1/S);
}

findformulaneg2(k,N,t)=
{
  my(res);
  for(l=-1,1,
    res=findformulaneg0(k,l,N,t);
    if(#res,
       print([[2,l]],": ",res,", ",cost3neg(l,N,res))
    )
  );
}
```

(1). (b). $k=3$: writing `findformulaneg2(3,1,1)` gives three formulas with $M=1$, 2, and 1 for $(D/2)=-1$, 0, and 1 respectively, `findformulaneg2(3,2,1)` again gives three formulas all with $M=2$, and higher values of N do not give anything useful. Thus we keep the formula with $N=1$ for $(D/2)=0$, and the formulas for $N=2$ for $(D/2)=\pm 1$. This gives the following:

$$L(\chi_D,-2)=\begin{cases}\dfrac{S_3^{(1)}(|D|,2)}{7} & \text{if } D\equiv 1\pmod 8\\ -\dfrac{S_3^{(1)}(|D|,2)}{9} & \text{if } D\equiv 5\pmod 8\\ -S_3^{(1)}(|D|/4,1) & \text{if } D\equiv 0\pmod 4\ .\end{cases}$$

(1). (c). $k=5$: writing `findformulaneg2(5,1,1)` gives three formulas with $M=1/2$, 1, and 1 for $(D/2)=-1$, 0, and 1 respectively, `findformulaneg2(5,2,1)` gives one formula for $(D/2)=1$ with $M=2$, and higher values of N do not give anything useful. Thus, keeping the best formulas gives the following:

$$L(\chi_D,-4)=\begin{cases}-\dfrac{S_5^{(1)}(|D|,2)}{5} & \text{if } D\equiv 1\pmod 8\\ \dfrac{16S_5^{(2)}(|D|,1)-S_5^{(1)}(|D|,1)}{2805} & \text{if } D\equiv 5\pmod 8\\ \dfrac{16S_5^{(2)}(|D|/4,1)-S_5^{(1)}(|D|/4,1)}{5} & \text{if } D\equiv 0\pmod 4\ .\end{cases}$$

(1). (d). $k = 7$: writing `findformulaneg2(7,1,2)` gives three formulas with $M = 1/2$, $2/3$, and $1/2$ for $(D/2) = -1$, 0, and 1 respectively, `findformulaneg2(7,2,2)` gives two formulas for $(D/2) = -1$ and 1, both with $M = 2/3$, and higher values of N do not give anything useful. Thus, keeping the best formulas gives the following:

$$L(\chi_D, -6) = \begin{cases} \dfrac{S_7^{(1)}(4|D|, 2) + 64S_7^{(1)}(|D|, 2)}{183} & \text{if } D \equiv 1 \pmod{8} \\ \dfrac{S_7^{(1)}(4|D|, 2) - 64S_7^{(1)}(|D|, 2)}{183} & \text{if } D \equiv 5 \pmod{8} \\ \dfrac{S_7^{(1)}(|D|, 1) - 4096S_7^{(2)}(|D|/4, 1)}{183} & \text{if } D \equiv 0 \pmod{4}\ . \end{cases}$$

Using the relation $S_k^{(1)}(4|D|, 2) = S_k^{(1)}(|D|, 2) + S_k^{(3)}(|D|)$ for $D \equiv 1 \pmod{4}$ proved below gives the faster formulas:

$$L(\chi_D, -6) = \begin{cases} \dfrac{S_7^{(3)}(|D|) + 65S_7^{(1)}(|D|, 2)}{183} & \text{if } D \equiv 1 \pmod{8} \\ \dfrac{S_7^{(3)}(|D|) - 63S_7^{(1)}(|D|, 2)}{183} & \text{if } D \equiv 5 \pmod{8} \\ \dfrac{S_7^{(1)}(|D|, 1) - 4096S_7^{(2)}(|D|/4, 1)}{183} & \text{if } D \equiv 0 \pmod{4}\ . \end{cases}$$

(1). (e). $k = 9$: writing `findformulaneg2(9,1,2)` gives three formulas with $M = 1/4$, $1/2$, and $1/2$ for $(D/2) = -1$, 0, and 1 respectively, `findformulaneg2(9,2,2)` gives one formula for $(D/2) = 1$ with $M = 2/3$, and higher values of N do not give anything useful. Thus, keeping the best formulas and including the improvement using $S_k^{(3)}(|D|)$ and $S_k^{(4)}(|D|/4)$ gives the following:

$$L(\chi_D, -8) = \begin{cases} -\dfrac{257S_9^{(3)}(|D|) + 65809S_9^{(1)}(|D|, 2)}{29085} & \text{if } D \equiv 1 \pmod{8} \\ -\dfrac{257S_9^{(1)}(4|D|, 1) - 16912384S_9^{(2)}(|D|, 1) + 272S_9^{(1)}(|D|, 1)}{383455485} & \text{if } D \equiv 5 \pmod{8} \\ -\dfrac{257S_9^{(4)}(|D|/4) - 16781312S_9^{(2)}(|D|/4, 1) + 17S_9^{(1)}(|D|/4, 1)}{29085} & \text{if } D \equiv 0 \pmod{4}\ . \end{cases}$$

(2). Assume that m is odd. We have

$$S_k^{(1)}(4m, 2) = \sum_{s \in \mathbb{Z}} \sigma_{k-1}^{(1)}((4m - s^2)/2)\ ,$$

so s is even, and since $\sigma_{k-1}^{(1)}(m)$ only depends on the odd part of m we have

$$S_k^{(1)}(4m, 2) = \sum_{\substack{s \in \mathbb{Z} \\ s \text{ odd}}} \sigma_{k-1}^{(1)}(m - s^2) + \sum_{\substack{s \in \mathbb{Z} \\ s \text{ even}}} \sigma_{k-1}^{(1)}(m - s^2)\ .$$

The second sum is by definition equal to $S_k^{(3)}(m)$. Since m is odd, when s is odd we have $\sigma_{k-1}^{(1)}(m - s^2) = \sigma_{k-1}^{(1)}((m - s^2)/2)$, so we obtain finally

$$S_k^{(1)}(4m, 2) = S_k^{(1)}(m, 2) + S_k^{(3)}(m)\ .$$

Let us look at the costs. Using the initial formula $S_k^{(1)}(4|D|,2) = \sum_{s\in\mathbb{Z}} \sigma_{k-1}^{(1)}(|D|-s^2)$ requires $|D|^{1/2}$ terms. Using the formula that we have just proved and the fact that $S_k^{(1)}(|D|,2)$ has already been computed, the cost is that of $S_k^{(3)}(|D|) = \sum_{s\in\mathbb{Z}} \sigma_{k-1}^{(1)}(|D|-4s^2)$, which requires $|D|^{1/2}/2$ terms.

(3). We clearly have

$$S_k^{(1)}(4m,1) = S_k^{(4)}(m) + \sum_{s\in\mathbb{Z}} \sigma_{k-1}^{(1)}(m-s^2) = S_k^{(4)}(m) + S_k^{(1)}(m,1)\ .$$

Applying this to $m = |D|/4$, the initial formula requires $|D|^{1/2}$ terms, while using this formula and the fact that $S_k^{(1)}(|D|/4,1)$ has already been computed requires only $|D|^{1/2}/2$ terms.