Aufgabe 1:

```
erzeugeZufahlsZahlen(N=10000)=
{   local(a,p,l);
    l=vector(N,i, []);

    for(i=1,N,
        a=100000+random(900000);
        b=100000+random(900000);
        if(b%2==0, b++);
        l[i]=[a,b]
    );

    write("Blatt7-Aufgabe1-Liste.gp", l);
}

\\erzeugeZufahlsZahlen();


myJacobi(a,b)=
{   local(vorzeichen=1, potenz);
    \\if(b%2==0, error("Fehler: b ist gerade!"));

    if(gcd(a,b)>1, return(0));

    until( (a==2) || (a== -1),
        a=a%b;

        if(a==1, break);

        \\ solange durch zwei teilen (falls mgl.),
        \\ bis a==2 ist:
        n=0;
        while((a%2==0) && (a >2),
            a=a \ 2;
            n++;
        );
        \\ dann hat man das Jacobi symbol um (2/b)^n geändert
        \\ und (2/b) = (-1)^((b^2-1)/8)
        potenz = (b^2-1)/8;
        vorzeichen=vorzeichen*(if((n*potenz)%2==1, -1, 1));
        if(a==2, break);


        \\ wenn a und b ungerade: QR anwenden:
        \\ b ist ungerade und a ebenfalls, wenn wir bis hierhin gekommen sind
        potenz= (a-1)*(b-1) /4;
        vorzeichen=vorzeichen*(if(potenz%2==1, -1, 1));
        \\ Quadratische Reziprozität anwenden:
        return(vorzeichen*myJacobi(b,a));
    );


    if(a== 1, return(1));

    if(a== 2,
        potenz = (b^2-1)/8;
        vorzeichen=vorzeichen*(if(potenz%2==1, -1, 1));
        return(vorzeichen);
    );

    if(a==-1,
        potenz = (b-1)/2;
```

```
        vorzeichen=vorzeichen*(if(potenz%2==1, -1, 1));
        return(vorzeichen);
    );
}

#
{
    Liste=read("Blatt7-Aufgabe1-Liste.gp");
    for(i=1,#Liste,
        L=Liste[i];
        ls=myJacobi(L[1],L[2]);
        \\rs=kronecker(L[1],L[2]);
        \\if(ls!=rs, error("Fehler bei ",L, " ",ls, ":",rs));
    );
}
\\ time = 1,264 ms.
```

Aufgabe 2: Es sei (x_0, y_0) eine Lösung der diophantischen Gleichung $ax + by = 1$, welche z.B. mit dem Satz von Bezout ermittelt werden kann.

Die Bijektion ist dann gegeben, durch die Funktion:

$$\begin{aligned} f_{a,b} : \mathbb{Z} &\longrightarrow \left\{(x, y) \in \mathbb{Z}^2 : ax + b = 1\right\} \\ t &\mapsto (x_0 - bt, y_0 + at) \end{aligned}$$

Zu zeigen: die Abbildung ist injektiv und surjektiv, also bijektiv.

injektiv: wir betrachten das Bild von t_1 und t_2 in der ersten Komponenten:

$$x_0 - bt_1 = x_0 - bt_2 \Rightarrow t_1 = t_2$$

Die Bilder sind nur für das gleiche Argument gleich, also $f_{a,b}$ ist injektiv.

surjektiv: Also zu jedem Paar (x, y) welche die diophantische Gleichung erfüllt, gibt es ein eindeutiges $t \in \mathbb{Z}$.

Sei (x, y) ein Lösung, dann ist $t = \frac{x_0 - x}{b}$. Dieses t eingesetzt in $x' = x_0 - bt, y' = y_0 + at$ ergibt $ax' + by' = 1$.

Aufgabe 3:

```
/* gibt die unimodulare Matrix zurück, die das
   m-fache der Spalte "von" zur Spalte "nach" addiert */
MatrixC1(N, m, von, nach)=
{   local(Rij=matrix(N,N,i,j,0));

    Rij[von,nach]=m;
    C=matdiagonal(vector(N,i,1))+Rij;

    return(C);
}

/* gibt die unimodulare Matrix zurück, die die Spalte
   "SpalteEins" mit der Spalte "SpalteZwei" vertauscht */
MatrixC2(N,SpalteEins, SpalteZwei)=
{   local( C=matdiagonal(vector(N,i,1)) );

    C[SpalteEins, SpalteEins]= 0;
    C[SpalteZwei, SpalteZwei]= 0;
    C[SpalteEins, SpalteZwei]= 1;
    C[SpalteZwei, SpalteEins] =1;
```

```
    return(C);
}

/* gibt die unimodulare Matrix zurück, die die Spalte
   "Spalte" mit -1 Multipliziert */
MatrixC3(N, Spalte)=
{   local(C= matdiagonal(vector(N,i,1)) );

    C[Spalte,Spalte]=-1;

    return(C);
}


/* gibt den Index i zurück mit
   0 < |v[i]| <= |v[j]| für alle j <> i: */
findeMinimumBetrag(v)=
{
    local( BetragVektor = vector(#v, i , abs(v[i])),
           merkIndex=0
    );

    \\ solange merkIndex erhöhen, bis ungleich Null:
    until(BetragVektor[merkIndex++], );

    for(i=1,#BetragVektor,
        if( (BetragVektor[i] > 0) &&
            (BetragVektor[i] < BetragVektor[merkIndex]) ,
            merkIndex = i;
        );
    );

    return(merkIndex);
}

map(a)=
{   local(T, i,   result,
          dim=#a, tempa=a );

    T= matdiagonal(vector(dim, i,1)); \\ Einheitsmatrix
    while( sum(k=1,dim,abs(tempa[k])) > 1,
        i= findeMinimumBetrag(tempa);
        val= tempa[i];

        for(j=1,dim,
            if(j==i, next);

            m= tempa[j] \ val;
            tempa[j]= tempa[j] - tempa[i]*m;
            \\ Über diese Berechnung in T Buch führen:
            T= T*MatrixC1(dim, -m, i,j);
        );
    ); \\ von while

    \\ ist noch eine Permutation nötig, um die Eins
    \\ an die erste Stelle zu bringen?
    if(tempa[1]==0,
        j=1;
        \\ erhöhe j bis tempa[j] ungleich 0:
        until(tempa[j++], );
```

```
        T=T*MatrixC2(dim, 1, j);
        tempa[1] = tempa[j];
        tempa[j] = 0;
    );

    \\ ist es noch nötig mit -1 zu multiplizieren?
    if(tempa[1] == -1,
        T=T*MatrixC3(dim, 1);
    );

    /* aus irgendeinem Grund muss ich die Transponierte
       Matrix zurückgeben:
        ich soll U*a lösen, aber die MatrixC[1-3] Routinen sind
        eigentlich für a~*U ausgelegt (Zeile und dann
        RechtsMultiplikation mit den Spaltentransformationen)
    */
    T=T~;
    if(T*a != vector(#a,i,if(i==1,1,0))~,
        error("Fehler: U erfüllt die Bedingung nicht!");
    );

    if(abs(matdet(T)) != 1,
        error("Fehler: U nicht unimodular!");
    );

    return(T);
}
/*
a=[11, 7, 19]~;
U=map(a);

printp(U*a);
printp(U);
*/

\\ Aufgabe4:
N=5;
a=(vector(N,i,i))~;
T=map(a);

printp(T*a);  \\ für T gilt:  T*a = [1,0,0,0,0]~
printp(T^-1); \\ <- hat a als erste Spalte!, U=T~ ist das
              \\    U aus dem Lemma von Seite 28
              \\ irgendwo im Skript/Aufgabenzettel sind
              \\ sind Zeilen und Spalten vertauscht
printp((T^-1)~);    \\ <- das ist U=(T~)^-1 aus dem Satz von Seite 28
U=(T^-1)~;
printp((U^-1)*[1,r_1,r_2,r_3,r_4]~);
/*
  [(-2 r_1 + (-3 r_2 + (-4 r_3 + (-5 r_4 + 1))))]
  [(r_1)]
  [((r_2))]
  [((r_3))]
  [((r_4))]
*/
printp(a~*((U^-1)*[1,r_1,r_2,r_3,r_4]~)); \\ == 1 \ Hurra.
/*
  U=(T^(-1))~ ist das U aus dem Satz. Eine Lsg
  der dioph. Gl. wird durch U^(-1)*[b/g, r_1,...,r_4]~
  beschrieben. (hier ist b/g == 1)
  r_1,...,r_4 sind unabhängige Parameter in Z.
*/
```

Aufgabe 4: (Siehe auch das Beispiel am Ende des vorigen Pari/GP-Skriptes.)

Bestimmen Sie eine Lösung von $\sum_{j=1}^{n} jx_j = 1$. Der Koeffizientenvektor hat ggt 1. Nach einem Satz der Vorlesung gibt es eine Lösung.

Die Lösung wird beschrieben durch:

$$\left\{ \vec{x}\ in\mathbb{Z}^n \middle| x_1 = 1 - \sum_{j=2}^{n} jr_j, \quad x_j = r_j \quad \text{für } j \geq 2, \quad r_j \in \mathbb{Z} \right\}$$

Aufgabe 5: Man zeige, dass die Gruppe $\mathrm{GL}(2,\mathbb{Z})$ erzeugt wird von den Matrizen

$$A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Alle drei Matrizen haben die Determinante ± 1, sind also in $\mathrm{GL}(2,\mathbb{Z})$. Wir müssen noch zeigen, dass sich jede Matrix $M \in \mathrm{GL}(2,\mathbb{Z})$ als Produkt mit diesen Matrizen schreiben lässt.

Mit Pari/GP können wir uns davon überzeugen, dass

- $A \cdot A = B \cdot B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ und $A \cdot C \cdot A = C^{-1}$
- AM die erste Zeile von M mit -1 multipliziert
- BM die Zeilen von M vertauscht
- $C^n M$ das $n-$fache der zweiten Zeile von M zur ersten Zeile von M addiert
- $BC^n BM$ das $n-$fache der ersten Zeile von M zur zweiten Zeile von M addiert

(Wenn M nicht von rechts sondern von links heranmultipliziert wird, dann gelten die entsprechenden Aussagen für Spalten statt für Zeilen.)

Wir nutzen den Hinweis ($\exists U \in \mathrm{GL}(2\mathbb{Z}) : UM = T$, mit $T \in \mathrm{GL}(2,\mathbb{Z})$ und die ersten Spalte von T hat ist $[1,0]^{\tilde{}}$) aus und zeigen, dass sich T als solches Produkt von A, B und C schreiben lässt. In einem zweiten Schritt zeigen wir, dass auch für U eine solche Produktdarstellung existiert.

1. Es sei $T = \begin{bmatrix} 1 & \alpha \\ 0 & \beta \end{bmatrix}$ und $T \in GL(2,\mathbb{Z})$, d.h. insbesondere gilt: $\det T = \pm 1 \Rightarrow \beta = \pm 1$.

   Im Fall $\beta = -1$ ist $T = BABC^{\alpha}$, im Fall $\beta = 1$ ist $T = C^{\alpha}$.

2. Es ist noch zu zeigen, dass es ein $U \in \mathrm{GL}(2,\mathbb{Z})$ gibt, welches M nach T überführt:

   Sei nun $M = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \in \mathrm{GL}(2,\mathbb{Z})$. O.B.d.A sei $0 \leq |a| < |b|$ (ansonsten setze M := B*M). Wegen $ad - bc = \pm 1$ ist $b = qa + r$, dabei ist $q = \lfloor d/c \rfloor$ und $r = \det M$.

   Wir addieren nun das $-q$-fache der ersten Zeile zur zweiten Zeile (Reduktion mod a), das ergibt $r = \det M = \pm 1$ in der zweiten Zeile, dann ziehen wir das ra-fache der zweiten Zeile von der ersten Zeile ab, das ergibt eine 0 in der ersten Zeile. Es folgen Zeilenvertauschung und, falls $\det M = -1$ ist, noch Multiplikation der ersten Zeile mit (-1):

   $$\underbrace{B \cdot C^{ra} \cdot BC^{-q}B}_{=:U} \cdot M$$

   Bzw. falls $\det M = -1$: $U := A * U$.

Wir haben gezeigt: U ist ein Produkt der drei gegebenen Matrizen und auch $T = UM$, also auch $M = U^{-1}T$. (U^{-1} ist das Produkt der Inversen in umgekehrter Reihenfolge, die Inversen sind ebenfalls als Produkte mit den drei gegebenen Matrizen darstellbar.)