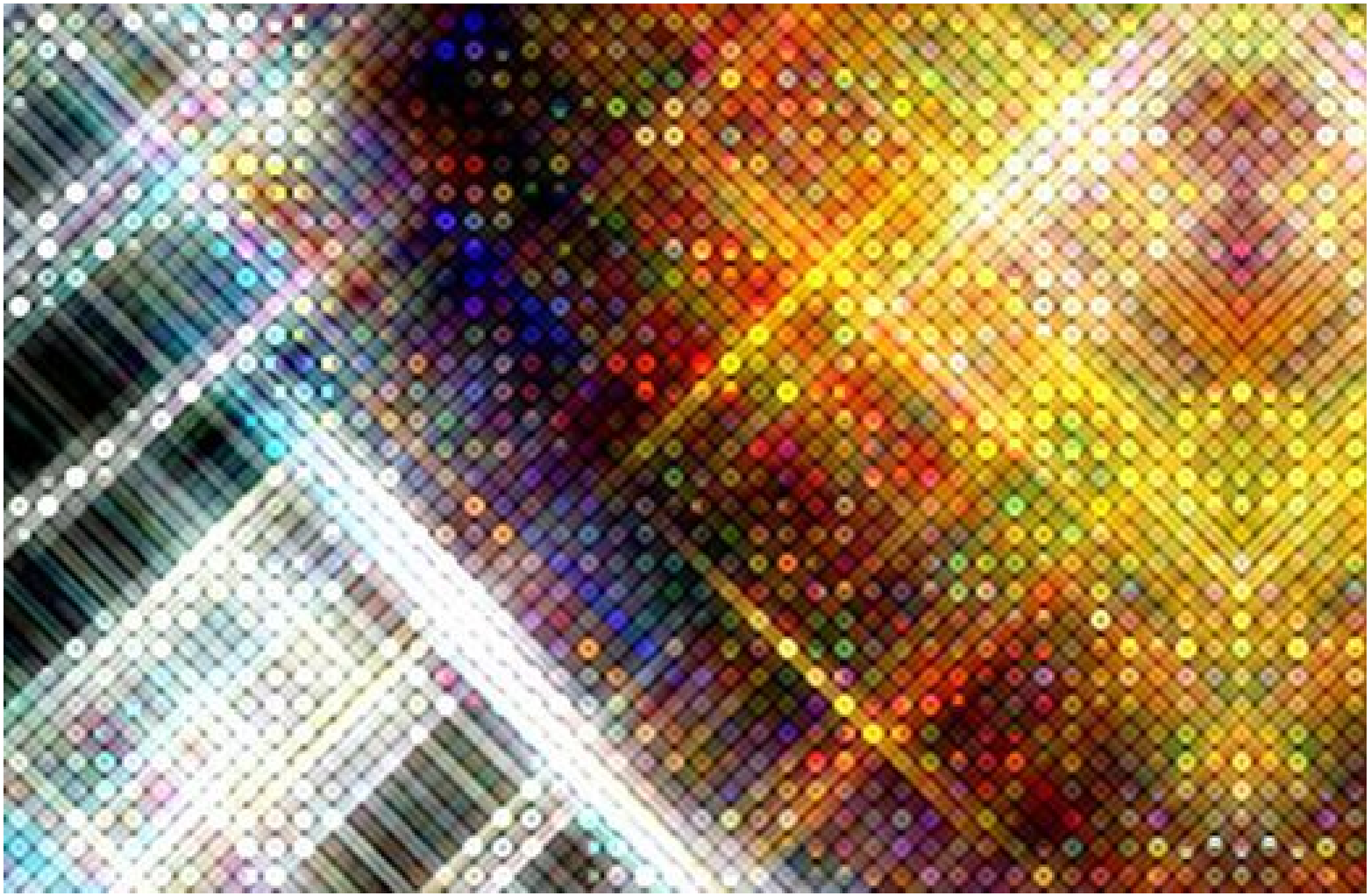
ИвГУ, ф-т МиКН, курс 2

"КОМПЬЮТЕРНАЯ АЛГЕБРА"

Тема 13.

Квадратичные вычеты и невычеты

Лектор: Н. И. Яцкин, 2014



ИЗВЛЕЧЕНИЕ КВАДРАТНОГО КОРНЯ В КОЛЬЦАХ ВЫЧЕТОВ

Как и в предыдущей **Теме 12**, мы будем работать в кольцах и полях классов вычетов $\mathbb{Z}_m$. Однако, вместо ***линейных*** уравнений вида $a \cdot x = b$, исследование которых сводится к исследованию ***обратимости*** вычетов, мы будем заниматься ***квадратичными*** уравнениями вида $x^2 = a$.

Главным окажется вопрос о существовании ***квадратного корня*** из вычета.

Карикатура на **Адриена Мари Лежандра** (1820 г.) — единственный известный портрет учёного.

По ошибке долгие годы в качестве портрета ***Адриена Мари Лежандра*** *фигурировал гораздо более "благородный" лик французского политика* ***Луи Лежандра*** *(разобрались только в 2005 году).*

Адриен Мари Лежа́ндр
(*Adrien-Marie Legendre*, **1752** — **1833**) — французский математик, преподаватель Военной школы в Париже, член Парижской Академии наук. В годы Французской революции Лежандр, вместе с **Лагранжем** и **Лапласом**, активно участвовал в Комиссии по введению метрической системы, профессор Нормальной и Политехнической школ. Его имя внесено в список величайших учёных Франции, помещённый на первом этаже Эйфелевой башни.

Далее:

ОПРЕДЕЛЕНИЕ СИМВОЛА ЛЕЖАНДРА

Предположим, что модуль $m = p$ является ***простым нечетным*** числом, т. е. рассмотрим случай поля вычетов $\mathbb{Z}_p$.(где $p \neq 2$.). ***Лежандр*** ввел ***символ-индикатор*** $\left(\frac{a}{p}\right)$ для целого числа a. ***По определению:***

$\left(\frac{a}{p}\right) = 0 \Leftrightarrow p|a$, т. е. если число a определяет ***нулевой*** вычет;

$\left(\frac{a}{p}\right) = 1 \Leftrightarrow a$ является ***квадратичным вычетом*** $\mathbf{mod}\ p$,

т. е. если $p \nmid a$ и существует целое число x такое, что $x^2 = a \bmod p$;

$\left(\frac{a}{p}\right) = -1 \Leftrightarrow a$ является ***квадратичным невычетом*** $\mathbf{mod}\ p$,

т. е. если $p \nmid a$ и ***не*** существует такого целого числа x, что $x^2 = a \bmod p$.

То же самое – как ***функция*** на поле $\mathbb{Z}_p$ со значениями в множестве $\{\mathbf{0}, \mathbf{1}, \mathbf{-1}\}$:

$$\mathbb{Z}_p \ni a \longmapsto \left(\frac{a}{p}\right) = \begin{cases} 0, \text{ if } a = 0; \\ 1, \text{ if } (a \neq 0) \wedge \left(\exists\, x \in \mathbb{Z}_p\right)(x^2 = a); \\ -1, \text{ if } (a \neq 0) \wedge \left(\nexists\, x \in \mathbb{Z}_p\right)(x^2 = a). \end{cases}$$

Задание 1. Представить процедуру-функцию `LEGENDREsymb(a,p)`, вычисляющую символ **Лежандра**.

Замечание. *В пакете* `numtheory` *предусмотрена "фирменная" процедура* `legendre`.

Код к заданию 1 и примеры применения.

```
> LEGENDREsymb:=proc(a::integer,p::prime)
  local a1,i;
if p=2 then
  ERROR(`The 2-nd argument, p, must be an odd prime.`);
end if;
a1:=a mod p;
if a1=0 then
  RETURN(0);
else
  for i from 1 to p-1 do
    if i^2 mod p=a1 then
      RETURN(1);
    end if;
  end do;
  RETURN(-1);
end if;
end proc;
```

```
> LEGENDREsymb(2562357894096485,1000003);
```

-1

```
> LEGENDREsymb(80,2);
```

```
Error, (in LEGENDREsymb) The 2-nd argument, p, must be
an odd prime.
```

Задание 2. Модифицировать процедуру `LEGENDREsymb(a,p)` так, чтобы новая процедура `sqrtmod(a,m)` работала в произвольном ***кольце*** $\mathbb{Z}_m$ и (в любом случае) возвращала ***множество*** решений уравнения $x^2 = a$ в этом кольце.

Код к заданию 2 и примеры применения.

```
> sqrtmod:=proc(a::integer,m::posint)
  local a1,i,ans;
a1:=a mod m;
ans:={};
for i from 0 to m-1 do
  if i^2 mod m=a1 then
    ans:=ans union {i};
  end if;
end do;
RETURN(ans);
end proc;
```

Ниже для $m = 7, 9, 10$ заполняются
таблицы квадратных корней.
Квадратичные вычеты и невычеты наблюдаются в этих таблицах "невооруженным глазом".

```
> tab:=Matrix(2,m):
for i from 0 to m-1 do
  tab[1,i+1]:=i;
  tab[2,i+1]:=sqrtmod(i,m);
od:
tab;
```

$$m := 7$$

0	1	2	3	4	5	6
{0}	{1, 6}	{3, 4}	{ }	{2, 5}	{ }	{ }

$$m := 9$$

0	1	2	3	4	5	6	7	8
{0, 3, 6}	{1, 8}	{ }	{ }	{2, 7}	{ }	{ }	{4, 5}	{ }

$$m := 10$$

0	1	2	3	4	5	6	7	8	9
{0}	{1, 9}	{ }	{ }	{2, 8}	{5}	{4, 6}	{ }	{ }	{3, 7}

СВОЙСТВА СИМВОЛОВ ЛЕЖАНДРА

(Доказательства см. в учебниках по теории чисел.)

(1) $[a' \equiv a \bmod p] \Rightarrow [\left(\frac{a'}{p}\right) = \left(\frac{a}{p}\right)]$;

(2) $\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \bmod p$ ***(критерий Эйлера);***

(3) $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$ ***(мультипликативность);***

(4) $[(a,p) = 1] \Rightarrow [\left(\frac{a^2 b}{p}\right) = \left(\frac{b}{p}\right)]$;

(5) $\left(\frac{1}{p}\right) = 1;\ \left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}};\ \left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}$.

ЗАКОН КВАДРАТИЧНОЙ ВЗАИМНОСТИ

(Сформулирован Эйлером в 1783 г., доказан Гауссом – в 1796 г.)

Теорема 1. ***Для любых простых нечетных p и q справедливо:***

$$\left(\frac{p}{q}\right) = (-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}} \left(\frac{q}{p}\right). \qquad (*)$$

Замечание. *Доказательство можно найти в учебниках по теории чисел. Свойства* **(1) – (5)** *и соотношение* **(*)** *используются при построении эффективных алгоритмов вычисления символов Лежандра (для больших* p*).*

РЕКУРСИВНЫЙ АЛГОРИТМ ВЫЧИСЛЕНИЯ СИМВОЛОВ ЛЕЖАНДРА

[1.] Приводим a по модулю p (см. свойство (1)).

[2.] Разлагаем результат на простые множители $a = p_1^{k_1} \dots p_s^{k_s}$ и применяем мультипликативность (3):

$$\left(\frac{a}{p}\right) = \left(\frac{p_1}{p}\right)^{k_1} \dots \left(\frac{p_s}{p}\right)^{k_s}. \qquad (**)$$

[3.] Множители в (**) с четными степенями можно удалить, т. к. они равны 1 (см. свойствово (4)).

[4.] Если $p_j = 2$ и k_j нечетно, то используем третью из формул (5) для вычисления $\left(\frac{2}{p}\right)$.

[5.] Если $p_j \neq 2$ и k_j нечетно, то используем закон взаимности (*) для "переворачивания" символов Лежандра:

$$\left(\frac{p_j}{p}\right) = (-1)^{\frac{p-1}{2}\cdot\frac{p_j-1}{2}} \left(\frac{p}{p_j}\right).$$

[6.] Рекурсивно применяем процедуру к каждому из символов $\left(\frac{p}{p_j}\right)$.

ПРОЦЕДУРА, РЕАЛИЗУЮЩАЯ РЕКУРСИВНЫЙ АЛГОРИТМ

Используемые стандартные процедуры: `ifactors(a)` – факторы (множители) целого числа a; возвращает **структурированный список** следующего вида:

$$\mathbf{fct} = \left[u, \left[[p_1, k_1], \ldots, [p_s, k_s]\right]\right],$$

где u – обратимый элемент (1 или -1); p_i – простые множители; k_i – кратности $(i = 1, \ldots, s)$.

Вложенный список, следующий за обратимым элементом, может быть извлечен указателем $\mathbf{fct}\,[2]$; $\mathbf{fct}\,[2][\mathbf{i}][1]$ указывает на i-й простой множитель, $\mathbf{fct}\,[2][\mathbf{i}][2]$ – на его кратность.

```
> LEGrec:=proc(a::integer,p::prime)
  local a1,i,L,fct,nf,q,k,v;
a1:=a mod p;
if a1=0 or a1=1 then
  RETURN(a1);
else
  L:=1;
  fct:=ifactors(a1);
  nf:=nops(fct[2]);
  for i from 1 to nf do
    q:=fct[2][i][1];
    k:=fct[2][i][2];
    if q=2 and type(k,odd) then
      L:=L*((-1)^((p^2-1)/8));
    elif q<>2 and type(k,odd) then
```

Вот она – рекурсия!

```
      L:=L*((-1)^((p-1)*(q-1)/4))*LEGrec(p,q);
    end if;
  end do;
  RETURN(L);
end if;
end proc;
```

Пример применения.

```
> LEGrec(2562357894096485,1000003);
                                -1
```

Готовимся к знакомству еще с одним выдающимся математиком.

Что это за формула?

Чьё имя она носит?

$$[[\boldsymbol{a},\boldsymbol{b}],\boldsymbol{c}]+[[\boldsymbol{b},\boldsymbol{c}],\boldsymbol{a}]+[[\boldsymbol{c},\boldsymbol{a}],\boldsymbol{b}]=\boldsymbol{0}$$

Ответ: **тождество Якоби.**

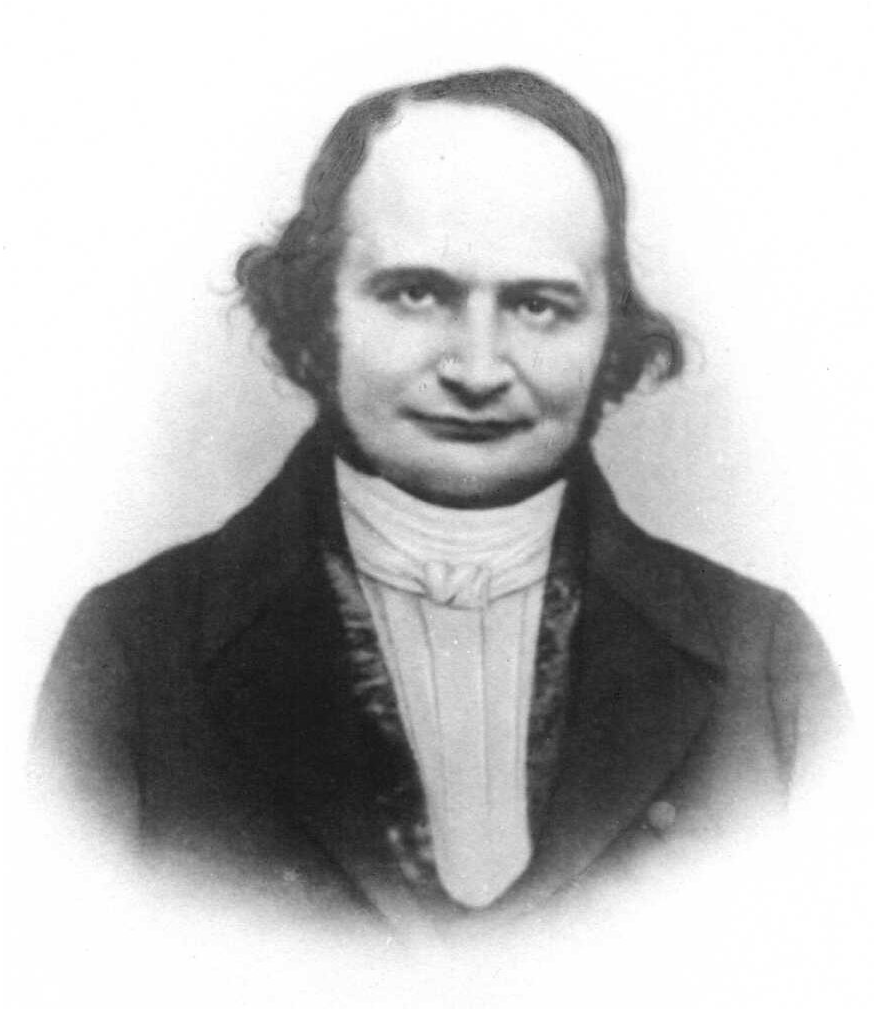
Оно справедливо, например, для **коммутаторов**

$$[A, B] = A \cdot B - B \cdot A$$

в **алгебре квадратных матриц**,

а также - для **векторного произведения** в трехмерном пространстве (здесь: $[\bar{a}, \bar{b}] = \bar{a} \times \bar{b}$).

А кто такой **Якоби?**



Карл Гу́став Я́коб Яко́би (*Carl Gustav Jacob Jacobi*; **1804** —**1851**) — немецкий математик и механик. Внёс огромный вклад в комплексный анализ, линейную алгебру, динамику и другие разделы математики и механики. Младший брат российского академика, физика **Бориса Семёновича Якоби**. Член Берлинской академии наук, Лондонского королевского общества, член-корреспондент Парижской академии наук, иностранный член-корреспондент **Петербургской Академии наук** и т. д.

Родился в семье банкира **Симона Якоби**, в которой были ещё двое сыновей и дочь. Старший брат, **Мориц** (**Борис Семенович**), стал российским академиком, младший продолжил отцовское дело.

Во время революции 1848 года Якоби имел неосторожность поддержать либералов в парламенте; после подавления революции возмущённый король отменил пенсию Якоби.

См.: матрица **Якоби,** тождество **Якоби,** уравнения **Гамильтона – Якоби, якобиан** отображения, **символ Якоби**.

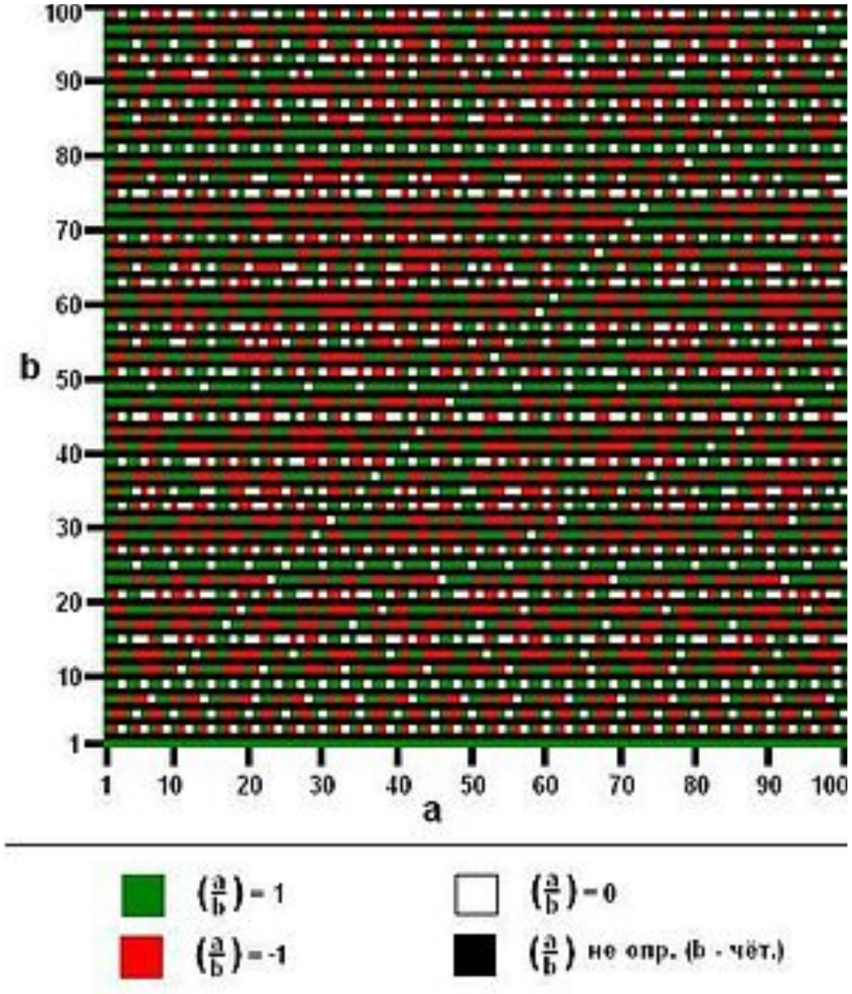
Именно последняя тема будет нас интересовать в данном курсе.

СИМВОЛ ЯКОБИ

обозначается так же, как символ **Лежандра**: $\left(\frac{a}{m}\right)$, но теперь внизу – не обязательно простое (но, по-прежнему обязательно, - нечетное) натуральное число m. При $m = 1$ полагается $\left(\frac{a}{1}\right) = 1$ для любого целого a. При нечетном $m > 1$ рассматривается разложение на простые множители $m = q_1^{l_1} \dots q_t^{l_t}$ и дается ***определение***:

$$\left(\frac{a}{m}\right) = \left(\frac{a}{q_1}\right)^{l_1} \dots \left(\frac{a}{q_t}\right)^{l_t}, \qquad (***)$$

где в правой части фигурируют символы Лежандра.



Символ Якоби уже не связан напрямую с извлечением квадратных корней и разрешимостью квадратичных сравнений.

Он используется в вероятностных тестах простоты и криптографии.

Задание 3. Представить процедуру-функцию

`JACOBIsymb(a,m)`, вычисляющую символ **Якоби**.

(Вам понадобится какая-либо версия процедуры вычисления символа **Лежандра**. Можно задействовать более простую, без рекурсии, версию **`LEGENDREsymb(a,p)`**.)

*Затем вам будет предложено снова представить себя компьютером и вычислить (подсматривая в коды) **(а)** символ **Лежандра** $\left(\frac{-6}{7}\right)$; **(б)** символ **Якоби** $\left(\frac{-6}{63}\right)$.*

Код к заданию 2 и примеры применения.

```
> JACOBIsymb:=proc(a::integer,m::posint)
  local a1,i,symb,fct,q,k;
if m=2 then
  ERROR(`The 2-nd argument, m, must be odd.`);
end if;
a1:=a mod m;
symb:=1;
fct:=ifactors(m)[2];
for i from 1 to nops(fct) do
  q:=fct[i][1];
  k:=fct[i][2];
  symb:=symb*(LEGENDREsymb(a1,q))^k;
end do;
RETURN(symb);
end proc;
```

```
> LEGENDREsymb(-6,7);JACOBIsymb(-6,7);JACOBIsymb(-6,63);
```

1

1

0

Замечание. *В пакете* **numtheory** *предусмотрена "фирменная" процедура* **jacobi**.

```
> evalb(JACOBIsymb(-6,63)=jacobi(-6,63));
```

true

А есть еще:

СИМВОЛ КРОНЕКЕРА.

СИМВОЛ КРОНЕКЕРА

обозначается так же, как символ Лежандра и символ Якоби: $\left(\frac{a}{m}\right)$, но теперь не только вверху, но и внизу может располагаться произвольное целое число.

Определение в ***особых*** случаях:

$$\left(\frac{a}{0}\right) = \begin{cases} 1, \text{if } |a| = 1; \\ 0, \text{if } |a| \neq 1; \end{cases}$$

$$\left(\frac{a}{1}\right) = 1;$$

$$\left(\frac{a}{-1}\right) = \begin{cases} 1, \text{if } a \geq 0; \\ -1, \text{if } a < 0; \end{cases}$$

$$\left(\frac{a}{2}\right) = \begin{cases} 0, \text{if } 2 \mid a = 0; \\ 1, \text{if } a \bmod 8 = 1 \text{ or } a \bmod 8 = 7; \\ -1, \text{if } a \bmod 8 = 3 \text{ or } a \bmod 8 = 5. \end{cases}$$

Замечание. *Именно этот пункт определения является ключевым. Он учитывает особые свойства вычетов* **mod 2** *по сравнению с вычетами по нечетному простому модулю.*

Рассмотрение случаев продолжается:

Если m - ***положительное нечетное*** число, то ***индекс Кронекера*** $\left(\frac{a}{m}\right)$ полагается равным ***индексу Якоби***.

Если m - ***положительное четное*** число, то представляем его в виде $m = 2^k l$, где $k \geq 1$, а l - ***нечетное*** число, и полагаем:

$$\left(\frac{a}{m}\right) = \left(\frac{a}{2}\right)^k \left(\frac{a}{l}\right).$$

Если же m - ***отрицательное*** $(m \leq -2)$ число, то полагаем:

$$\left(\frac{a}{m}\right) = \left(\frac{a}{-1}\right)\left(\frac{a}{-m}\right).$$

Мотивировка данных определений: *обеспечивается выполнение ряда свойств, которые, в свою очередь, позволяют применять* ***символы Кронекера*** *в прикладной компьютерной алгебре.*

(1) $\left(\frac{a}{m}\right) = \pm 1$ тогда и только тогда, когда a и m ***взаимно просты***;

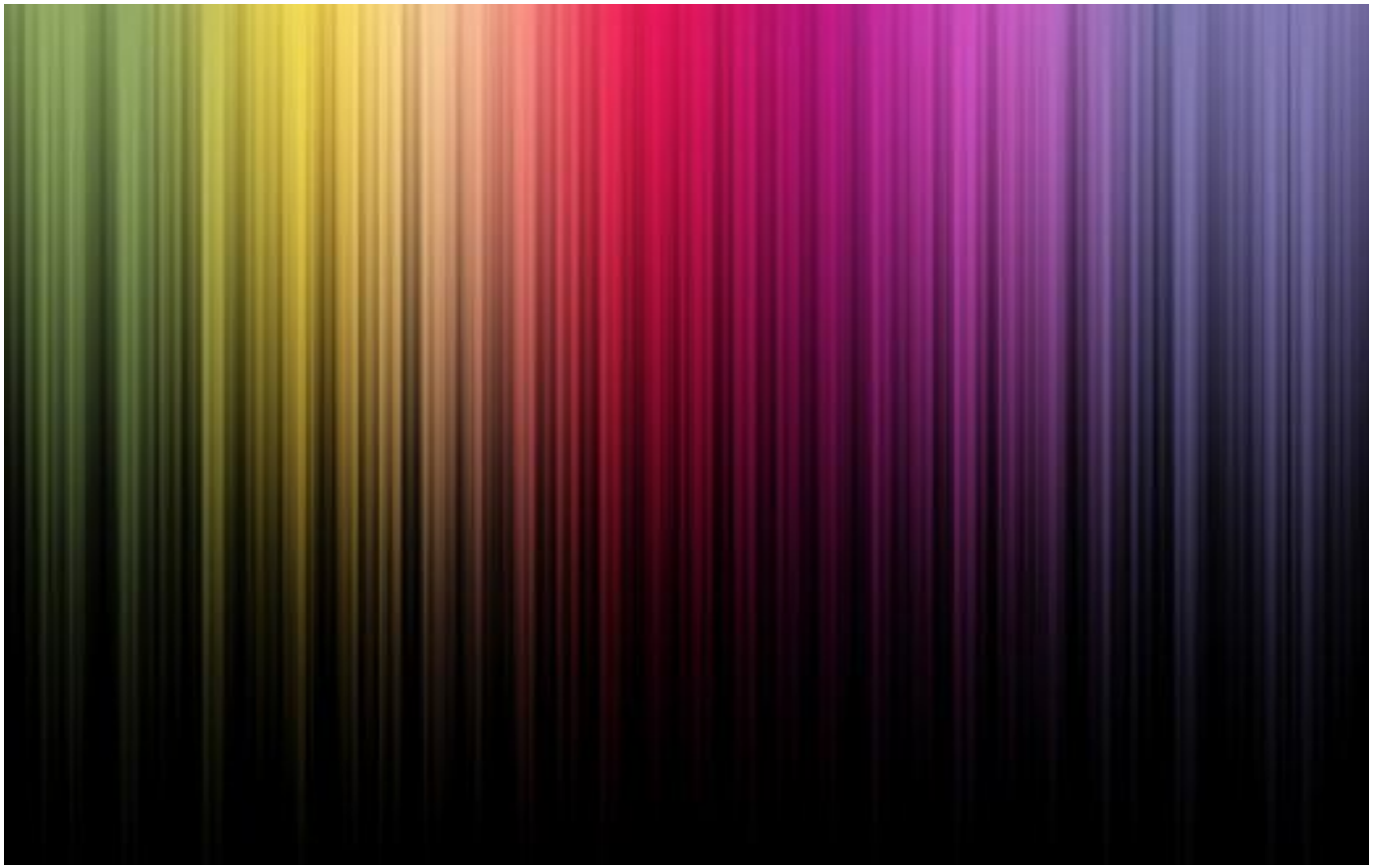
в противном случае: $\left(\frac{a}{m}\right) = 0$.

(2) $\left(\frac{ab}{m}\right) = \left(\frac{a}{m}\right)\left(\frac{b}{m}\right)$, за исключением случая, когда $m = -1$ и одно из чисел, a или b, обращается в 0.

(3) $\left(\frac{a}{kl}\right) = \left(\frac{a}{k}\right)\left(\frac{a}{l}\right)$, за исключением случая, когда $a = -1$ и одно из чисел, k или l, обращается в 0.

И т. д. *Существует, в частности, версия* ***закона квадратичной взаимности*** *для индексов Кронекера.* ***Подробнее*** *см. в* Wikipedia: http://en.wikipedia.org/wiki/Kronecker_symbol

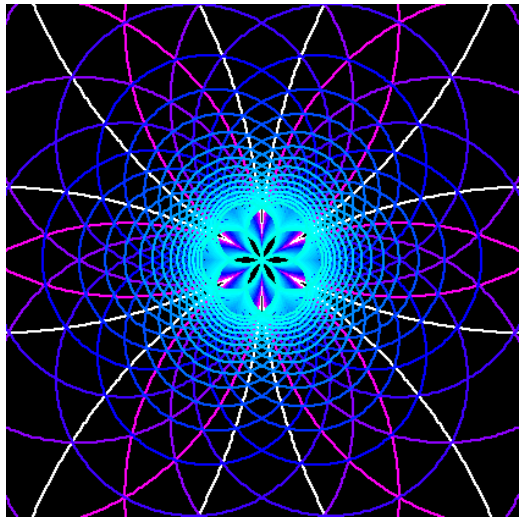
Контрольный вопрос. *Можно ли "сокращать" символ Лежандра (или Якоби) "как дробь"?*



Контрольный вопрос. *Можно ли "сокращать" символ* ***Лежандра*** *(или* ***Якоби****) "как дробь"?*

Контрольный ответ. Н-и-и-з-з-з-я-я!

$$\left(\frac{2}{3}\right) = -1;\ \left(\frac{6}{9}\right) = \left(\frac{6}{3}\right)^2 = 0.$$



Задание 4. Представить процедуру-функцию **KRONECKERsymb(a,m)**, вычисляющую символ Кронекера. (Понадобится процедура вычисления символа Якоби **JACOBIsymb(a,m)**.)

Идея: *вычисление символа* ***Кронекера*** *сводится к вычислению символов* ***Якоби*** *во всех случаях, кроме нескольких особых:*

$m = 0; m = 1; m = -1; m = 2; m = -2;$

m – *четное положительное число;*

m – *четное отрицательное число.*

Процедура будет рекурсивной.

```
> KRONECKERsymb:=proc(a::integer,b::integer)
  local k;
if b=0 and abs(a)=1 then
  RETURN(1);
elif b=0 and abs(a)<>1 then
  RETURN(0);
elif b=1 then
  RETURN(1);
elif b=-1 and a<0 then
  RETURN(-1);
elif b=-1 and a>=0 then
  RETURN(1);
elif b=2 and a mod 2=0 then
  RETURN(0);
elif b=2 and ((a mod 8=1) or (a mod 8=7)) then
  RETURN(1);
elif b=2 and ((a mod 8=3) or (a mod 8=5)) then
```

```
  RETURN(-1);
elif b=-2 then
  RETURN(KRONECKERsymb(a,-1)*KRONECKERsymb(a,2));
elif b>2 and b mod 2<>0 then
  RETURN(JACOBIsymb(a,b));
elif b>2 and b mod 2=0 then
  k:=1;
  while b mod (2^k)=0 do
    k:=k+1;
  end do;
  RETURN((KRONECKERsymb(a,2))^(k-1)*
                          JACOBIsymb(a,b/(2^(k-1))));
elif b<-2 then
  RETURN(KRONECKERsymb(a,-1)*KRONECKERsymb(a,-b));
end if;
end proc;
```

Пример применения. Заполним таблицу значений $\left(\frac{a}{m}\right)$ для $a = -4..4$ (откладывается по **горизонтали**) $m = -4..4$ (откладывается по **вертикали**).

```
> M:=Matrix(10):M[1,1]:=``:
for j from 2 to 10 do
  M[1,j]:=-6+j:M[j,1]:=-6+j:
end do:
for i from 2 to 10 do
  for j from 2 to 10 do
    M[i,j]:=KRONECKERsymb(-6+j,-6+i):
  end do;
end do;
M;
```

	-4	-3	-2	-1	0	1	2	3	4
-4	0	-1	0	-1	0	1	0	1	0
-3	1	0	-1	1	0	1	-1	0	1
-2	0	1	0	-1	0	1	0	-1	0
-1	-1	-1	-1	-1	1	1	1	1	1
0	0	0	0	1	0	1	0	0	0
1	1	1	1	1	1	1	1	1	1
2	0	-1	0	1	0	1	0	-1	0
3	-1	0	1	-1	0	1	-1	0	1
4	0	1	0	1	0	1	0	1	0