Математические основы защиты информации
Лабораторная работа №3
Алгебраические уравнения в кольце вычетов

БГУ, ММФ, каф. ДУиСА,
доцент Чергинец Д.Н.

Китайская теорема об остатках

Пусть $n_1, n_2, \ldots, n_k \in \mathbb{N}$, НОД$(n_i, n_j) = 1$ при $i \neq j$,
$b_1, b_2, \ldots, b_k \in \mathbb{Z}$.
Тогда система

$$
\begin{aligned}
x &\equiv b_1 \pmod{n_1},\\
x &\equiv b_2 \pmod{n_2}, \quad (1)\\
&\ldots\\
x &\equiv b_k \pmod{n_k},
\end{aligned}
$$

имеет в $\mathbb{Z}_n^*$ единственное решение, причем его можно вычислить по формуле

$$x = \sum_{i=1}^{k} N_i C_i b_i \pmod{n}, \quad (2)$$

где

$n := n_1 \ldots n_k$,
$N_i := \frac{n}{n_i}$,
$C_i := N_i^{-1} \pmod{n_i}$.

Задание 1.

Реализовать алгоритм, который находит решение системы (1) при помощи формулы (2). Сколько арифметических операций выполняется во время

работы данного алгоритма? Является ли он полиномиальным?

Алгоритм Гарнера

- **Вход:** $b_1, \ldots, b_k \in \mathbb{Z}$,
  $n_1, \ldots, n_k \in \mathbb{N}$ – взаимно простые.
- **Выход:** x, $0 \le x < n_1 \ldots n_k$, – решение (2).
- 1. Задаем начальные значения переменных:
  $N := 1$,
  $x := b_1 \pmod{n_1}$.
- 2. Для $i := 2, \ldots, k$ последовательно вычисляем:
  $N := N n_{i-1}$,
  $C := N^{-1} \pmod{n_i}$,
  $y := C(b_i - x) \pmod{n_i}$,
  $x := Ny + x$.
- 3. Выдаем результат: x.

Задание 2.

Реализовать алгоритм Гарнера, который находит решение системы (1). Сколько арифметических операций выполняется во время работы данного алгоритма? Является ли он полиномиальным? При помощи функции Timing сравнить время вычисления алгоритма Гарнера со временем вычисления алгоритма из задания 1.

Квадратичные вычеты

Пусть задано натуральное число n. Вычет $a \in \mathbb{Z}_n^*$ называется квадратичным вычетом по модулю n, если уравнение

$$x^2 \equiv a \pmod{n} \qquad (3)$$

имеет решение. В противном случае a называется квадратичным невычетом по модулю n.

Квадратичные вычеты по простому модулю

Теорема. Пусть p нечетное простое, $a \in \mathbb{Z}_p^*$. Тогда

1) уравнение (3) имеет либо два корня, либо ни одного;

2) кольцо $\mathbb{Z}_p$ имеет $\frac{p-1}{2}$ квадратичных вычетов и столько же квадратичных невычетов;

3) a является квадратичным вычетом тогда и только тогда, когда $a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$;

4) a является квадратичным невычетом в том и только том случае, когда $a^{\frac{p-1}{2}} \equiv -1 \pmod{p}$.

Для нахождения корней уравнения (3) в случае когда p – простое, на данный момент имеются лишь вероятностные (т.е. не детерминированные) полиномиальные алгоритмы. Ниже приведен один из них.

Алгоритм Шенкса решения уравнения $x^2 \equiv a \pmod{p}$.

Вход: p — простое, a — квадратичный вычет по модулю p.

Выход: Корень уравнения $x^2 \equiv a \pmod{p}$.

1. Методом пробных делений на 2 находим такие s, $t \in \mathbb{N}$, t — нечетное, что $p - 1 = t\,2^s$.
2. Случайным образом выбираем невычет n числа p при помощи условия $n^{\frac{p-1}{2}} \equiv -1 \pmod{p}$.
3. Вычисляем $b := n^t \pmod{p}$, $r := a^{\frac{t+1}{2}} \pmod{p}$.
4. Задаем начальные значения параметров $d := 0$, $f := a^t \pmod{p}$, $c := b$.
5. Для $i := 1, \ldots, s-1$, последовательно выполняем шаги 5.1, 5.2.
   5.1. Вычисляем $c := c^2 \pmod{p}$.
   5.2. Если $f^{2^{s-1-i}} \not\equiv 1 \pmod{p}$, то $d := d + 2^i$, $f := f\,c \pmod{p}$.
6. Выдаем результат $x = r\,b^{d/2} \pmod{p}$.

Задание 3.

Реализовать алгоритм Шенкса. Сколько корней может иметь уравнение (3) для простого p?

Символы Лежандра и Якоби

Символом Лежандра $\left(\frac{a}{p}\right)$, где p – нечетное простое, $a \in \mathbb{Z}$, называется функция

$$\left(\frac{a}{p}\right) := \begin{cases} -1, & \text{если } a - \text{ квадратичный невычет по модулю } p; \\ 0, & \text{если НОД}(a, p) = p; \\ 1, & \text{если } a - \text{квадратичный вычет по модулю } p\,. \end{cases}$$

Нетрудно доказать, что символ Лежандра можно вычислять по формуле

$$\left(\frac{a}{p}\right) := a^{\frac{p-1}{2}} \pmod p.$$

Обобщением символа Лежандра является символ Якоби. Пусть n – нечетное, большее единицы число, разложение на простые множители которого имеет вид

$n = p_1 p_2 \ldots p_k$, где среди простых чисел p_i могут быть одинаковые.

Символом Якоби произвольного целого числа a называется число

$$\left(\frac{a}{n}\right) := \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right)\ldots\left(\frac{a}{p_k}\right),$$

где $\left(\frac{a}{p_i}\right)$ – символ Лежандра. Когда n простое, символ Якоби совпадает с символом Лежандра, поэтому символ Якоби имеет такое же обозначение. Значение символа Якоби для составных чисел уже не зависит от разрешимости квадратного уравнения как в случае символа Лежандра. Если вычислять символ Якоби исходя из определения, то данный алгоритм будет экспоненциальным, однако, благодаря свойствам символа Якоби он вычисляется за полиномиальное время и даже символ Лежандра вычисляется не по определению, а как символ Якоби, что и иллюстрируют следующие вычисления

```
In[ ]:= p1 = 16 395 155 176 313 139 322 021;
        p2 = 1 362 766 142 559 093 439 411 703;
        n = p1 p2;
        FactorInteger[n] // Timing
        факторизовать целое ч··· затраченное врем
        JacobiSymbol[11, n] // Timing
        символ Якоби              затраченное вр
Out[ ]= {1.32601, {{16 395 155 176 313 139 322 021, 1}, {1 362 766 142 559 093 439 411 703, 1}}}
Out[ ]= {0., -1}
```

In[ ]:=
```
p =
  12 298 120 408 548 381 921 949 835 559 130 363 125 529 753 678 612 793 629 866 092 168 341 157 677 ↘
   004 548 909 890 194 322 320 188 611 624 608 811 733 931 456 088 225 443 388 533 288 329 565 083 ↘
   193 503 675 166 129 710 355 993 503 285 460 782 162 687 449 001 932 607 888 044 856 843 834 448 ↘
   046 669 642 562 579 539 878 983 216 911 024 659 511 203 267 436 306 425 169 766 384 977 812 745 ↘
   329 180 607 883 735 997 471 881 725 271 290 829 988 189 123 070 515 621 275 520 208 366 410 857 ↘
   284 732 676 256 969 330 943 202 597 868 406 283 019 471 093 419 998 714 880 651 628 035 574 273 ↘
   203 133 347 881 501 552 355 055 036 863 559 577 444 579 332 767 110 609 641 162 352 730 908 905 ↘
   341 755 140 840 817 688 237 650 074 856 938 925 705 023 786 037 242 333 779 882 084 378 507 742 ↘
   099 650 835 194 579 364 530 134 715 691 360 486 004 055 929 437 282 386 138 610 160 667 839 748 ↘
   759 682 356 185 444 266 407 034 990 930 818 776 631 796 519 392 735 335 398 537 603 853 614 696 ↘
   448 286 566 938 691 953 402 253 898 131 437 650 538 537 150 455 169 352 842 653 301 272 066 180 ↘
   275 693 357 706 780 245 624 218 430 092 628 955 784 730 114 249 729 402 847 990 445 984 529 364 ↘
   256 240 643 230 072 104 677 467 341 673 637 006 440 427 969 179 876 300 573 682 502 400 130 753 ↘
   541 300 429 553 804 552 084 654 561 152 802 138 223 932 661 655 478 035 386 502 339;
PrimeQ[p]
простое число?
a = p - 3;
PowerMod[a, (p - 1)/2, p] // AbsoluteTiming
степень по модулю                длительность по настенным часам

JacobiSymbol[a, p] // AbsoluteTiming
символ Якоби                  длительность по настенным часам
```

Out[ ]=
```
True
```

Out[ ]=
```
{0.0450639, 1}
```

Out[ ]=
```
{0.000015086, 1}
```

Вычисление символа Якоби

Вход: a — целое, n - нечетное натуральное.

Выход: $\left(\frac{a}{n}\right)$ — символ Якоби.

1. Инициализация $J := 1$.
2. Если $n = 1$, то выдаем результат: J, конец алгоритма.
3. Если $a < 0$, то $a := -a$, $J = (-1)^{(n-1)/2} J$.
4. Если $n = 1$, то выдаем результат: J, конец алгоритма.
5. Если $a = 0$, выдаем результат: 0, конец алгоритма.
6. Методом пробных делений представляем число a в виде

   $a = 2^s t$,

   где t — нечетное.
7. Если s — нечетное и $n \equiv \pm 3 \pmod 8$, то $J := -J$.
8. Если $n \equiv 3 \pmod 4$ и $t \equiv 3 \pmod 4$, то $J := -J$.
9. Вычисляем $a := n \pmod t$, $n := t$, переходим к шагу 4.

Задание 4.

Реализовать алгоритм вычисления символа Якоби.

Квадратичный вычет по составному модулю

Пусть в уравнении (3) модуль является составным $n = p\,q$, p, q – простые, a – квадратичный вычет по модулю n, следовательно, a – квадратичный вычет по модулю p и по модулю q. Решая уравнения
$x^2 \equiv a \pmod p$, $x^2 \equiv a \pmod q$,
например, алгоритмом Шенкса, находим соответствующие уравнениям корни $\pm x_p, \pm x_q$. Далее, при помощи Китайской теоремы об остатках решаем четыре системы
1) $x \equiv x_p \pmod p$, $x \equiv x_q \pmod q$;
2) $x \equiv x_p \pmod p$, $x \equiv -x_q \pmod q$;
3) $x \equiv -x_p \pmod p$, $x \equiv x_q \pmod q$;
4) $x \equiv -x_p \pmod p$, $x \equiv -x_q \pmod q$
и получаем четыре корня уравнения (3). Аналогично делается в случае, когда n раскладывается более чем на два множителя.

Задание 5.

Известно, что n является произведением двух простых чисел. Вычислить символы Якоби $\left(\frac{c_i}{n}\right)$. Указать, где это возможно, сколько корней имеет уравнение $x^2 \equiv c \pmod n$. Найти хотя бы один из корней, где это просто сделать, вычислить символы Якоби $\left(\frac{x_i}{n}\right)$.

Вариант 1.

```
n =
 17 559 915 661 908 927 049 443 260 850 969 472 791 412 728 369 123 148 669 201 206 622 434 539 228 ↘
  709 717 808 393 545 371 189 408 773;
c1 = 9;
c2 = 121;
c3 =
 10 338 834 675 177 952 044 709 093 355 496 749 039 468 503 256 208 100 344 402 818 453 532 708 354 ↘
  274 744 654 461 334 970 081 074 489;
c4 =
 16 645 372 732 324 343 017 042 368 294 925 431 318 530 480 201 746 250 593 929 618 073 557 605 696 ↘
  684 409 426 766 731 807 609 858 685;
c5 = 11;
```

Вариант 2.

```
n =
 23 380 654 653 089 016 997 193 753 825 884 205 762 888 936 330 644 662 943 970 136 817 584 977 269 ↖
  064 138 574 418 248 745 386 424 849;
c1 = 25;
c2 = 169;
c3 =
 16 792 944 963 232 729 027 968 193 397 393 078 542 490 598 120 364 877 651 833 462 496 096 807 297 ↖
  997 760 245 892 257 102 552 737 569;
c4 =
 9 637 236 784 133 025 587 656 227 313 999 539 148 014 738 552 269 775 984 699 964 363 941 015 367 715 ↖
  532 710 821 558 208 926 592 625;
c5 = 13;
```

Вариант 3.

```
n =
 29 954 257 713 341 844 882 726 300 212 966 880 336 533 960 633 968 103 018 144 945 331 133 333 436 ↖
  371 206 699 195 734 174 880 801 349;
c1 = 49;
c2 = 4;
c3 =
 29 330 039 703 069 592 573 912 994 857 286 914 210 924 413 792 661 006 744 025 887 306 232 993 847 ↖
  934 692 302 536 653 638 905 430 689;
c4 =
 25 233 429 919 459 802 239 726 263 188 269 030 718 225 203 863 882 357 883 976 924 180 074 984 218 ↖
  686 730 911 220 223 061 682 236 019;
c5 = 2;
```

Вариант 4.

```
n =
 15 385 737 706 298 210 211 608 039 651 990 685 040 590 993 419 685 247 022 435 984 427 100 279 096 ↖
  537 722 064 100 042 120 038 314 509;
c1 = 121;
c2 = 9;
c3 =
 13 450 613 961 549 157 116 398 603 010 296 320 633 656 008 529 609 959 829 585 193 663 852 121 752 ↖
  481 645 881 900 559 666 977 133 089;
c4 =
 793 594 020 936 352 794 643 531 124 129 065 238 806 318 206 375 898 228 575 728 744 945 059 828 150 ↖
  026 314 672 331 193 627 143 603;
c5 = 3;
```

Вариант 5.

```
n =
 23 199 213 419 116 623 305 818 452 403 049 906 355 952 920 437 262 284 787 713 675 638 737 817 072 ↖
  134 509 955 970 401 665 953 327 657;
c1 = 169;
c2 = 25;
c3 =
 16 410 753 774 568 544 424 568 143 835 769 395 902 782 046 707 718 687 179 089 151 247 686 478 564 ↖
  914 181 799 896 908 006 608 289 641;
c4 =
 18 613 619 286 584 370 777 415 495 616 453 837 232 066 292 510 430 129 373 035 370 465 745 218 719 ↖
  418 480 894 119 016 784 124 605 985;
c5 = 5;
```

Вариант 6.

```
n =
 27 905 338 062 091 695 929 048 760 242 076 000 051 423 471 176 255 612 418 502 965 293 739 002 999 ↖
  443 746 451 619 655 865 481 345 593;
c1 = 4;
c2 = 49;
c3 =
 25 411 655 426 537 145 318 139 253 979 729 322 151 887 024 959 219 653 350 587 250 908 758 090 243 ↖
  839 784 588 266 334 102 176 995 241;
c4 =
 20 375 944 918 445 254 983 634 681 803 643 453 482 587 407 481 202 415 142 103 930 348 260 768 985 ↖
  038 744 482 562 093 765 711 348 138;
c5 = 7;
```

Вариант 7.

```
n =
 14 647 214 105 231 950 169 092 785 355 714 589 928 164 940 843 090 731 952 783 166 266 660 832 926 ↖
  805 210 272 423 553 224 530 799 661;
c1 = 9;
c2 = 121;
c3 =
 10 059 527 742 231 445 633 805 141 230 657 997 413 549 995 595 403 627 530 936 241 237 635 253 531 ↖
  773 512 061 879 143 812 524 352 761;
c4 =
 9 504 398 570 218 869 165 034 780 201 325 827 583 703 320 196 212 221 286 038 715 904 964 866 891 682 ↖
  619 542 244 904 244 606 712 529;
c5 = 11;
```

Вариант 8.

```
n =
 22 076 901 423 378 204 827 490 777 488 966 544 891 584 987 921 656 780 084 810 939 365 036 980 243 ↖
  875 615 255 891 616 928 137 406 769;
c1 = 25;
c2 = 169;
c3 =
 16 110 615 126 123 235 271 068 942 673 835 912 544 002 116 465 033 979 645 414 977 323 943 895 507 ↖
  331 806 480 925 520 145 612 218 761;
c4 =
 16 501 032 651 576 737 845 559 203 417 418 378 268 105 999 235 001 822 973 906 049 811 225 131 822 ↖
  577 372 213 240 129 030 815 675 468;
c5 = 13;
```

Вариант 9.

```
n =
 22 623 583 936 570 714 681 197 020 034 449 650 541 277 608 417 381 640 108 134 849 405 414 714 157 ↖
  692 782 134 600 543 552 395 634 517;
c1 = 49;
c2 = 4;
c3 =
 14 856 044 860 088 136 744 952 607 072 877 913 518 993 924 268 300 660 455 262 330 422 471 735 692 ↖
  049 350 833 310 539 795 510 792 729;
c4 =
 679 817 103 033 725 930 618 386 725 039 539 227 801 677 415 916 079 386 128 792 516 778 084 510 862 ↖
  102 991 037 904 543 206 362 264;
c5 = 2;
```

Вариант 10.

```
n =
 30 926 233 369 645 029 229 059 151 176 640 181 799 302 331 045 127 098 307 889 987 031 498 145 199 ↖
  771 386 792 016 479 155 529 472 393;
c1 = 121;
c2 = 9;
c3 =
 25 233 403 114 836 402 619 493 235 251 332 592 064 779 047 303 209 714 397 576 538 525 987 759 630 ↖
  300 116 391 309 878 095 466 258 561;
c4 =
 24 781 289 396 055 981 905 002 345 132 073 327 831 522 686 751 355 141 232 758 829 001 464 609 862 ↖
  542 847 154 527 514 093 372 394 011;
c5 = 3;
```

Вариант 11.

```
n =
 20 752 377 193 257 105 013 047 715 523 773 694 683 584 689 293 383 488 982 437 262 624 986 254 984 ↖
  935 633 898 832 090 781 280 383 997;
c1 = 169;
c2 = 25;
c3 =
 11 775 207 344 158 093 603 942 493 980 439 043 329 555 427 350 526 640 194 075 368 618 173 774 729 ↖
  199 215 795 895 769 106 127 833 729;
c4 =
 591 940 789 690 992 279 386 068 485 849 494 064 803 682 989 399 497 753 358 382 092 126 962 598 040 ↖
  740 914 540 242 394 040 709 804;
c5 = 5;
```

Вариант 12.

```
n =
 18 114 430 062 009 755 246 299 720 064 649 708 252 216 264 498 432 177 880 020 058 122 331 592 257 ↖
  885 222 861 950 475 937 046 798 137;
c1 = 4;
c2 = 49;
c3 =
 16 010 854 751 103 551 231 491 082 938 419 326 398 830 095 516 865 948 695 665 882 028 732 079 748 ↖
  687 599 950 903 719 504 452 452 089;
c4 =
 7 883 087 906 680 481 289 607 095 624 176 841 677 929 686 972 200 978 806 864 311 980 297 052 316 852 ↖
  957 258 820 194 189 923 797 258;
c5 = 7;
```

Вариант 13.

```
n =
 21 524 937 431 620 261 470 628 927 403 073 153 103 564 737 617 035 159 057 879 617 671 748 856 820 ↖
  043 660 000 479 284 880 104 871 209;
c1 = 9;
c2 = 121;
c3 =
 19 344 223 121 500 254 674 189 400 542 670 647 289 458 431 116 262 829 956 689 487 718 498 868 782 ↖
  755 899 291 549 577 287 158 303 689;
c4 =
 11 965 166 327 402 005 073 406 355 552 714 308 019 906 492 751 081 908 625 398 154 847 882 802 885 ↖
  359 391 746 900 287 306 675 898 623;
c5 = 11;
```

Вариант 14.

```
n =
 18 288 897 613 562 037 093 402 920 841 759 305 617 087 900 447 635 019 352 755 154 115 557 604 898 ↖
   783 856 846 762 557 998 753 825 109;
c1 = 25;
c2 = 169;
c3 =
 11 128 169 581 007 940 534 369 671 456 311 810 578 698 888 272 784 079 069 108 058 239 186 274 678 ↖
   291 637 225 783 659 888 905 022 249;
c4 =
 6 493 512 807 740 424 214 596 825 210 048 332 238 924 775 202 878 176 126 534 091 829 010 555 779 448 ↖
   265 586 430 667 007 282 384 985;
c5 = 13;
```

Вариант 15.

```
n =
 18 467 680 766 845 112 208 423 280 472 770 456 384 891 751 456 028 462 719 641 438 769 077 724 195 ↖
   421 814 262 684 632 755 824 089 757;
c1 = 49;
c2 = 4;
c3 =
 12 987 656 766 859 902 007 098 587 547 750 668 798 678 742 122 664 847 341 184 744 591 769 596 719 ↖
   696 645 116 003 292 324 228 464 089;
c4 =
 8 654 171 836 727 560 494 717 611 043 020 196 504 892 845 760 796 677 892 380 734 724 284 154 186 679 ↖
   948 941 985 464 921 481 063 704;
c5 = 2;
```

Вариант 16.

```
n =
 22 795 348 854 187 733 447 360 207 548 418 799 260 664 788 353 398 534 522 719 316 976 907 784 810 ↖
   952 513 198 691 903 627 020 676 861;
c1 = 121;
c2 = 9;
c3 =
 16 625 654 273 999 059 570 933 465 754 652 405 716 651 961 993 803 566 926 448 273 548 330 477 645 ↖
   133 175 202 339 679 043 857 704 721;
c4 =
 16 845 790 700 335 365 955 430 443 966 247 316 869 367 386 554 624 409 446 183 174 609 512 304 033 ↖
   891 783 267 648 769 357 067 148 511;
c5 = 3;
```

Вариант 17.

```
n =
 17 517 292 915 524 013 964 900 463 544 669 815 867 135 707 241 637 445 208 083 875 799 032 014 138 ↖
   426 646 510 362 095 079 577 697 957;
c1 = 169;
c2 = 25;
c3 =
 16 994 174 547 415 766 916 928 793 950 553 001 972 987 801 201 262 025 262 957 976 712 770 984 336 ↖
   149 661 511 069 552 385 948 794 569;
c4 =
 5 631 569 253 707 230 515 449 345 843 949 265 305 624 596 186 198 847 169 827 776 945 761 321 528 312 ↖
   154 105 082 745 339 572 965 254;
c5 = 5;
```

Вариант 18.

```
n =
 36 720 677 508 362 889 887 596 162 430 779 828 456 747 703 334 178 105 372 333 678 013 400 108 066 ↖
   162 134 218 026 276 509 712 195 809;
c1 = 4;
c2 = 49;
c3 =
 34 595 308 300 636 607 233 001 530 984 497 873 145 931 684 753 105 195 141 870 804 777 490 483 500 ↖
   009 613 948 366 546 981 341 584 441;
c4 =
 32 223 683 752 610 787 456 587 086 887 102 565 787 240 382 651 436 373 755 089 713 252 196 648 115 ↖
   358 675 435 072 232 487 847 985 121;
c5 = 7;
```

Вариант 19.

```
n =
 19 773 017 437 002 435 480 211 810 761 535 528 384 886 155 631 752 055 498 201 123 984 693 260 382 ↖
   835 588 723 222 829 743 119 402 809;
c1 = 9;
c2 = 121;
c3 =
 11 707 021 153 036 791 404 751 201 150 317 008 657 201 623 917 521 671 010 021 326 673 593 911 456 ↖
   577 624 441 295 679 429 676 277 129;
c4 =
 9 939 697 867 546 554 279 455 205 439 567 210 590 125 123 247 877 161 714 948 289 387 840 958 540 026 ↖
   783 853 983 068 678 522 531 671;
c5 = 11;
```

Вариант 20.

```
n =
 23 234 320 048 443 537 153 028 447 037 472 060 624 521 995 426 974 665 208 754 577 492 893 812 407 ↖
  025 273 355 187 322 729 561 425 461;
c1 = 25;
c2 = 169;
c3 =
 15 815 756 843 053 577 953 074 837 108 948 369 557 261 943 212 983 757 520 847 362 415 663 896 552 ↖
  076 675 032 443 204 569 380 994 321;
c4 =
 16 129 204 717 461 975 260 882 599 723 454 500 280 386 226 843 947 789 100 578 947 564 852 993 455 ↖
  947 742 559 306 968 463 230 812 244;
c5 = 13;
```

Вариант 21.

```
n =
 12 873 013 639 055 180 876 104 109 976 989 482 321 945 041 931 661 780 864 120 451 912 871 522 426 ↖
  568 241 088 496 768 813 013 529 397;
c1 = 49;
c2 = 4;
c3 =
 12 137 348 649 653 427 365 928 469 722 291 472 755 464 095 957 563 599 917 812 821 852 578 764 301 ↖
  689 013 538 597 972 124 413 074 729;
c4 =
 3 261 275 703 260 750 407 227 555 349 948 931 425 496 159 310 012 960 503 043 840 596 739 784 408 063 ↖
  371 720 673 794 946 245 003 283;
c5 = 2;
```

Вариант 22.

```
n =
 12 273 732 623 763 403 529 532 892 122 115 044 181 466 128 295 424 762 854 539 338 502 091 090 905 ↖
  974 791 367 845 577 920 822 832 353;
c1 = 121;
c2 = 9;
c3 =
 10 228 689 805 119 457 713 767 312 670 953 915 070 025 150 396 491 454 182 971 146 998 604 344 695 ↖
  298 153 520 429 930 675 850 719 081;
c4 =
 4 534 464 407 158 926 995 921 621 649 587 435 575 364 987 315 542 844 164 320 596 186 750 673 324 883 ↖
  305 824 258 321 326 022 310 358;
c5 = 3;
```

Вариант 23.

```
n =
 20 259 881 106 457 430 170 757 875 772 465 690 211 886 653 624 076 484 575 599 739 305 458 006 616 ↖
   588 632 776 013 695 516 322 517 577;
c1 = 169;
c2 = 25;
c3 =
 18 240 736 375 782 600 613 809 158 003 999 385 874 232 769 820 092 568 174 413 209 091 340 075 990 ↖
   580 351 427 672 143 257 634 362 561;
c4 =
 19 794 475 729 641 183 572 276 916 284 887 914 678 800 051 311 102 007 705 124 984 565 181 607 009 ↖
   502 352 040 704 029 891 929 272 102;
c5 = 5;
```

Вариант 24.

```
n =
 38 091 130 343 477 039 616 830 862 930 519 028 905 811 361 801 095 280 611 939 899 777 922 249 305 ↖
   218 372 217 892 417 051 327 374 841;
c1 = 4;
c2 = 49;
c3 =
 36 722 622 158 753 848 911 756 643 203 835 177 930 681 138 021 057 534 415 852 082 577 386 547 832 ↖
   783 424 189 716 011 407 604 861 609;
c4 =
 9 448 416 422 007 409 051 262 957 549 754 014 736 325 622 308 678 122 820 455 363 554 641 915 219 908 ↖
   063 111 508 069 200 713 607 596;
c5 = 7;
```

Вариант 25.

```
n =
 17 277 822 756 753 827 807 953 021 976 025 275 553 057 031 213 450 979 225 141 076 738 310 940 439 ↖
   611 422 712 983 473 023 338 490 521;
c1 = 9;
c2 = 121;
c3 =
 13 072 449 718 412 732 067 430 124 546 065 339 121 888 272 678 328 815 831 330 051 573 400 237 195 ↖
   063 853 365 128 632 183 421 963 369;
c4 =
 4 566 738 381 850 541 079 366 671 867 713 708 223 083 265 587 493 433 841 869 113 170 267 323 062 330 ↖
   331 864 717 856 069 103 187 388;
c5 = 11;
```

Вариант 26.

```
n =
 24 500 646 450 627 336 517 109 254 792 353 469 798 507 579 241 398 513 818 766 284 982 779 714 965 ↖
  018 433 821 218 470 081 645 910 709;
c1 = 25;
c2 = 169;
c3 =
 16 181 773 013 038 003 758 361 468 222 971 592 396 259 674 665 802 162 593 690 857 360 175 473 943 ↖
  763 602 848 912 081 897 375 102 689;
c4 =
 11 219 357 156 369 691 807 446 240 992 082 436 812 818 460 154 508 100 822 772 848 001 533 599 541 ↖
  160 836 048 316 764 970 561 057 802;
c5 = 13;
```

Вариант 27.

```
n =
 27 681 134 381 988 397 390 012 409 160 538 949 551 603 471 442 790 571 379 002 810 420 634 874 682 ↖
  916 769 316 132 294 661 251 505 741;
c1 = 49;
c2 = 4;
c3 =
 27 393 390 309 476 430 206 149 912 713 137 047 931 287 936 683 444 281 122 509 038 957 609 476 786 ↖
  373 869 789 874 771 435 513 933 929;
c4 =
 3 599 694 457 769 342 174 669 722 111 792 423 071 033 989 226 462 264 393 953 268 691 108 860 755 424 ↖
  790 311 242 332 412 728 416 563;
c5 = 2;
```

Вариант 28.

```
n =
 28 607 783 694 753 616 876 983 188 824 077 917 612 605 236 276 076 486 920 579 179 359 802 231 014 ↖
  131 654 492 389 138 436 093 915 161;
c1 = 121;
c2 = 9;
c3 =
 26 266 413 519 187 856 090 251 758 079 780 882 928 662 266 732 725 678 795 541 476 168 618 074 280 ↖
  026 012 027 464 903 190 624 193 281;
c4 =
 20 016 548 444 564 078 604 944 521 898 107 132 557 548 133 012 263 493 819 830 576 873 647 960 557 ↖
  653 184 154 779 610 678 983 414 513;
c5 = 3;
```

Вариант 29.

```
n =
 26 596 863 219 180 302 897 994 373 911 568 365 979 465 278 339 764 883 397 144 698 554 119 600 941 ↘
  429 572 999 000 084 264 894 116 717;
c1 = 169;
c2 = 25;
c3 =
 20 702 098 676 565 851 986 889 389 345 901 888 775 337 320 780 596 543 518 573 923 117 736 890 214 ↘
  280 144 997 320 340 236 136 341 041;
c4 =
 23 163 589 870 899 625 534 945 737 784 378 464 716 294 292 631 055 071 205 210 275 602 108 709 981 ↘
  672 307 287 317 251 574 423 728 393;
c5 = 5;
```

Вариант 30.

```
n =
 31 305 329 412 189 441 126 433 763 310 575 398 271 108 305 575 952 446 258 952 905 949 909 538 563 ↘
  619 082 509 841 760 984 491 887 993;
c1 = 4;
c2 = 49;
c3 =
 30 795 106 491 903 228 563 700 661 867 424 059 230 656 745 423 110 965 333 446 637 022 632 131 092 ↘
  346 426 059 263 794 665 584 459 921;
c4 =
 5 093 939 371 378 636 120 052 718 396 833 313 747 798 090 551 471 278 742 420 367 381 432 273 654 044 ↘
  990 196 377 034 854 514 300 161;
c5 = 7;
```