In[1]:=
```
RationalQ[x_] := (Head[x] === Rational || IntegerQ[x]);
```

In[146]:=
```
tribonacci = RecurrenceTable[{a[k] == a[k - 1] + a[k - 2] + a[k - 3],
    a[0] == 0, a[1] == 0, a[2] == 1}, a, {k, 3, 1000}];
```

In[147]:=
```
Table[tribonacci[[n]], {n, 1, 10}]
```

Out[147]=
```
{1, 2, 4, 7, 13, 24, 44, 81, 149, 274}
```

In[148]:=
```
listTribonacciSequence = Flatten[Table[
    Table[Take[tribonacci, {n, n + 4 * k - 1}], {n, 1, 100}], {k, 1, 10, 1}], 1];
```

In[149]:=
```
sumTribonacciSequence = Total /@ listTribonacciSequence;
```

In[150]:=
```
Table[sumTribonacciSequence[[n]], {n, 100}];
```

In[151]:=
```
tribonacci[[998]];
```

In[152]:=
```
Table[divideSequence[[n]], {n, 1000}];
```

In[153]:=
```
divideSequence = Flatten[Table[sumTribonacciSequence[[n]] / tribonacci[[k]],
   {n, 1, Length[sumTribonacciSequence]}, {k, 1, Length[tribonacci]}]]
```

Out[153]=
```
{14, 7, 7/2, ··· 997 994 ···, 1 762 776 824 ··· 15 ··· 42 750 639 384 / 5 128 983 763 ··· 242 ··· 9 450 609 001,
 8 285 051 075 563 961 161 044 720 409 280 051 048 / 44 338 257 949 022 630 766 198 624 158 427 000 925 635 523 642 ··· 169 ··· 472 265 666 010 074 477 859 199 789 932 765 503 984 125 240 893,
 1 035 631 384 445 495 145 130 590 051 160 006 381 /
  (101 938 463 243 632 901 876 579 371 717 190 048 762 630 927 705 007 454 272 817 044 369 276 793 156 189 749 051 ×
     479 152 377 237 075 801 586 464 248 011 390 224 ··· 23 ···
    7 240 625 540 185 427 997 955 868 026 793 112 032 095 904 831 658 727 915 618 917 540 818 814 206 589 053 500 593 ×
     829 351 875 943 061 563 847 973 703 215 713 923)}
```

Size in memory: 156.7 MB | + Show more | Show all | Iconize | Store full expression in notebook

In[154]:=
```
integerCheck = IntegerQ /@ divideSequence
```

Out[154]=
```
{True, True, False, True, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, ··· 997 914 ···, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False, False}
```

Size in memory: 8 MB | + Show more | Show all | Iconize | Store full expression in notebook

In[155]:=
```
Count[integerCheck, True]
```
Out[155]=
```
3485
```
In[156]:=
```
positionCheck = Position[integerCheck, True] // Flatten;
```
In[163]:=
```
ListIntgers = Table[divideSequence[[n]], {n, positionCheck}];
```
In[184]:=
```
data = Table[ListIntgers[[n]], {n, 1, 1000}];
```
In[185]:=
```
groups = {};
currentGroup = {};

Do[AppendTo[currentGroup, num];
 If[num == 2, AppendTo[groups, currentGroup];
  currentGroup = {}], {num, data}]
groups
```
Out[188]=
```
{{14, 7, 2}, {26, 13, 2}, {48, 24, 12, 2}, {88, 44, 22, 2}, {162, 81, 2},
 {298, 149, 2}, {548, 274, 137, 2}, {1008, 504, 252, 144, 42, 2}, {1854, 927, 2},
 {3410, 1705, 2}, {6272, 3136, 1568, 896, 2}, {11 536, 5768, 2884, 1648, 2},
 {21 218, 10 609, 2}, {39 026, 19 513, 3002, 2}, {71 780, 35 890, 17 945, 2},
 {132 024, 66 012, 33 006, 5501, 2}, {242 830, 121 415, 34 690, 2},
 {446 634, 223 317, 5514, 2}, {821 488, 410 744, 205 372, 2},
 {1 510 952, 755 476, 377 738, 2}, {2 779 074, 1 389 537, 2},
 {5 111 514, 2 555 757, 2}, {9 401 540, 4 700 770, 2 350 385, 2},
 {17 292 128, 8 646 064, 4 323 032, 2 470 304, 2}, {31 805 182, 15 902 591, 2},
 {58 498 850, 29 249 425, 2}, {107 596 160, 53 798 080, 26 899 040, 15 370 880, 34 310, 2},
 {197 900 192, 98 950 096, 49 475 048, 28 271 456, 2},
 {363 995 202, 181 997 601, 2}, {669 491 554, 334 745 777, 63 106, 2},
 {1 231 386 948, 615 693 474, 307 846 737, 27 986 067, 15 202 308, 2},
 {2 264 873 704, 1 132 436 852, 566 218 426, 2},
 {4 165 752 206, 2 082 876 103, 595 107 458, 2}, {7 662 012 858, 3 831 006 429, 2},
 {14 092 638 768, 7 046 319 384, 3 523 159 692, 1 084 049 136, 587 193 282, 2},
 {25 920 403 832, 12 960 201 916, 6 480 100 958, 2}, {47 675 055 458, 23 837 527 729, 2},
 {87 688 098 058, 43 844 049 029, 2}, {161 283 557 348, 80 641 778 674, 40 320 889 337, 2},
 {296 646 710 864, 148 323 355 432, 74 161 677 716, 42 378 101 552, 2},
 {545 618 366 270, 272 809 183 135, 3 661 868 230, 2},
 {1 003 548 634 482, 501 774 317 241, 2}, {1 845 813 711 616, 922 906 855 808,
  461 453 427 904, 263 687 673 088, 588 588 556, 320 009 312, 2},
 {3 394 980 712 368, 1 697 490 356 184, 848 745 178 092, 484 997 244 624, 141 457 529 682,
  41 913 342 128, 6 736 072 842, 2}, {6 244 343 058 466, 3 122 171 529 233, 2},
 {11 485 137 482 450, 5 742 568 741 225, 2}, {21 124 461 253 284, 10 562 230 626 642,
  5 281 115 313 321, 22 787 984 092, 1 991 183 076, 2}, {38 853 941 794 200,
  19 426 970 897 100, 9 713 485 448 550, 2 988 764 753 400, 1 618 914 241 425, 2},
 {71 463 540 529 934, 35 731 770 264 967, 10 209 077 218 562, 2},
 {131 441 943 577 418, 65 720 971 788 709, 2}, {241 759 425 901 552,
```

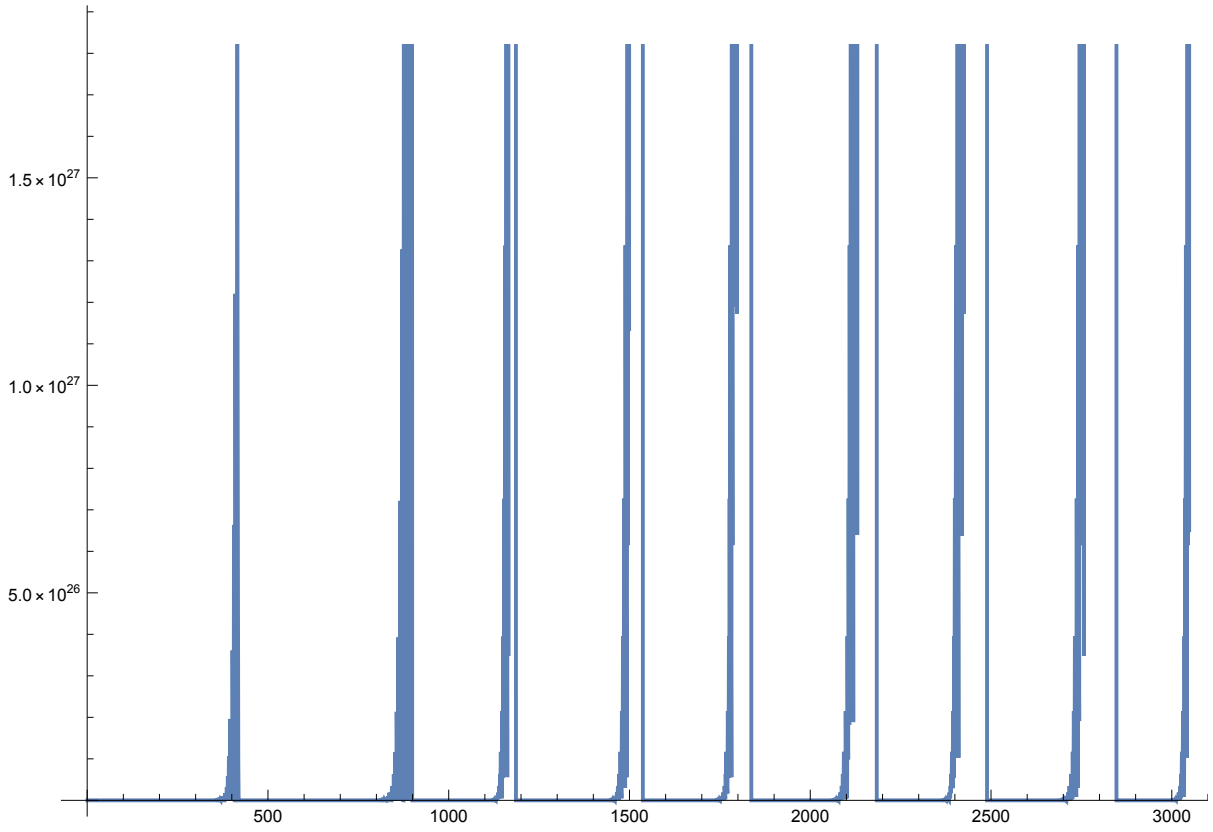
```
 120 879 712 950 776, 60 439 856 475 388, 18 596 878 915 504, 12 389 659 504, 2},
{444 664 910 008 904, 222 332 455 004 452, 111 166 227 502 226, 34 204 993 077 608, 2},
{817 866 279 487 874, 408 933 139 743 937, 2},
{1 504 290 615 398 330, 752 145 307 699 165, 2},
{2 766 821 804 895 108, 1 383 410 902 447 554, 691 705 451 223 777, 2},
{5 088 978 699 781 312, 2 544 489 349 890 656, 1 272 244 674 945 328, 726 996 957 111 616,
 2}, {9 360 091 120 074 750, 4 680 045 560 037 375, 115 556 680 494 750, 2},
{17 215 891 624 751 170, 8 607 945 812 375 585, 1 324 299 355 750 090, 2},
{31 664 961 444 607 232, 15 832 480 722 303 616,
 7 916 240 361 151 808, 4 523 565 920 658 176, 10 097 245 358 612, 2},
{58 240 944 189 433 152, 29 120 472 094 716 576, 14 560 236 047 358 288,
 8 320 134 884 204 736, 2 426 706 007 893 048, 115 557 428 947 288, 62 827 340 010 176,
 10 097 251 073 064, 2}, {107 121 797 258 791 554, 53 560 898 629 395 777, 2},
{197 027 702 892 831 938, 98 513 851 446 415 969, 2},
{362 390 444 341 056 644, 181 195 222 170 528 322, 90 597 611 085 264 161, 2},
{666 539 944 492 680 136, 333 269 972 246 340 068,
 166 634 986 123 170 034, 62 827 782 495 304, 2},
{1 225 958 091 726 568 718, 612 979 045 863 284 359, 175 136 870 246 652 674, 2},
{2 254 888 480 560 305 498, 1 127 444 240 280 152 749, 2},
{4 147 386 516 779 554 352, 2 073 693 258 389 777 176, 1 036 846 629 194 888 588, 2},
{7 628 233 089 066 428 568, 3 814 116 544 533 214 284,
 1 907 058 272 266 607 142, 317 843 045 377 767 857, 2},
{14 030 508 086 406 288 418, 7 015 254 043 203 144 209, 2}, {25 806 127 692 252 271 338,
 12 903 063 846 126 135 669, 1 985 086 745 557 867 026, 318 594 169 040 151 498, 2},
{47 464 868 867 724 988 324, 23 732 434 433 862 494 162, 11 866 217 216 931 247 081, 2},
{87 301 504 646 383 548 080, 43 650 752 323 191 774 040,
 21 825 376 161 595 887 020, 12 471 643 520 911 935 440, 2},
{160 572 501 206 360 807 742, 80 286 250 603 180 403 871, 2},
{295 338 874 720 469 344 146, 147 669 437 360 234 672 073, 2},
{543 212 880 573 213 699 968, 271 606 440 286 606 849 984, 135 803 220 143 303 424 992,
 77 601 840 081 887 671 424, 173 218 393 039 927 838, 2}, {999 124 256 500 043 851 856,
 499 562 128 250 021 925 928, 249 781 064 125 010 962 964, 142 732 036 642 863 407 408, 2},
{1 837 676 011 793 726 895 970, 918 838 005 896 863 447 985, 2},
{3 380 013 148 866 984 447 794, 1 690 006 574 433 492 223 897, 2},
{6 216 813 417 160 755 195 620, 3 108 406 708 580 377 597 810,
 1 554 203 354 290 188 798 905, 2}, {11 434 502 577 821 466 539 384,
 5 717 251 288 910 733 269 692, 2 858 625 644 455 366 634 846, 2},
{21 031 329 143 849 206 182 798, 10 515 664 571 924 603 091 399,
 3 004 475 591 978 458 026 114, 1 982 404 481 463 776 622, 2},
{38 682 645 138 831 427 917 802, 19 341 322 569 415 713 958 901, 2},
{71 148 476 860 502 100 639 984, 35 574 238 430 251 050 319 992,
 17 787 119 215 125 525 159 996, 2 964 519 869 187 587 526 666,
 878 376 257 537 062 970 864, 2}, {130 862 451 143 182 734 740 584,
 65 431 225 571 591 367 370 292, 32 715 612 785 795 683 685 146, 2},
{240 693 573 142 516 263 298 370, 120 346 786 571 258 131 649 185, 2},
{442 704 501 146 201 098 678 938, 221 352 250 573 100 549 339 469, 2},
{814 260 525 431 900 096 717 892,
```

```
 407 130 262 715 950 048 358 946, 203 565 131 357 975 024 179 473, 2},
{1 497 658 599 720 617 458 695 200, 748 829 299 860 308 729 347 600,
 374 414 649 930 154 364 673 800, 213 951 228 531 516 779 813 600, 2},
{2 754 623 626 298 718 654 092 030, 1 377 311 813 149 359 327 046 015, 2},
{5 066 542 751 451 236 209 505 122, 2 533 271 375 725 618 104 752 561, 2},
{9 318 824 977 470 572 322 292 352, 4 659 412 488 735 286 161 146 176,
 2 329 706 244 367 643 080 573 088, 1 331 260 711 067 224 617 470 336,
 716 832 690 574 659 409 407 104, 2 971 564 087 203 626 378 282, 2},
{17 139 991 355 220 527 185 889 504, 8 569 995 677 610 263 592 944 752,
 4 284 997 838 805 131 796 472 376, 2 448 570 193 602 932 455 127 072, 2},
{31 525 359 084 142 335 717 686 978, 15 762 679 542 071 167 858 843 489, 2},
{57 984 175 416 833 435 225 868 834, 28 992 087 708 416 717 612 934 417, 2},
{106 649 525 856 196 298 129 445 316, 53 324 762 928 098 149 064 722 658,
 26 662 381 464 049 074 532 361 329, 2}, {196 159 060 357 172 069 073 001 128,
 98 079 530 178 586 034 536 500 564, 49 039 765 089 293 017 268 250 282,
 8 173 294 181 548 836 211 375 047, 2 421 716 794 532 988 507 074 088, 2},
{360 792 761 630 201 802 428 315 278, 180 396 380 815 100 901 214 157 639,
 51 541 823 090 028 828 918 330 754, 2}, {663 601 347 843 570 169 630 761 722,
 331 800 673 921 785 084 815 380 861, 62 550 791 577 299 478 709 658, 2},
{1 220 553 169 830 944 041 132 078 128, 610 276 584 915 472 020 566 039 064,
 305 138 292 457 736 010 283 019 532, 50 856 382 076 289 335 047 169 922, 2},
{2 244 947 279 304 716 013 191 155 128, 1 122 473 639 652 358 006 595 577 564,
 561 236 819 826 179 003 297 788 782, 93 539 469 971 029 833 882 964 797, 2}}
```

In[162]:=

```
ListLinePlot[ Table[divideSequence[[n]], {n, positionCheck}]]
```

Out[162]=



In[158]:=
```
reverseDivideSequence = Flatten[Table[tribonacci[[k]] / sumTribonacciSequence[[n]],
    {n, 1, Length[sumTribonacciSequence]}, {k, 1, Length[tribonacci]}]]
```

Out[158]=

$$\left\{\frac{1}{14}, \frac{1}{7}, \frac{2}{7}, \text{··· 997 994 ···}, \frac{5\,128\,983\,763\ \text{··· 242 ···}\ 9\,450\,609\,001}{1\,762\,776\,824\ \text{··· 15 ···}\ 42\,750\,639\,384}, \right.$$

$$\frac{44\,338\,257\,949\,022\,630\,766\,198\,624\,158\,427\,000\,925\,635\,523\,642\ \text{··· 170 ···}\ 72\,265\,666\,010\,074\,477\,859\,199\,789\,932\,765\,503\,984\,125\,240\,893}{8\,285\,051\,075\,563\,961\,161\,044\,720\,409\,280\,051\,048},$$

$$\left(101\,938\,463\,243\,632\,901\,876\,579\,371\,717\,190\,048\,762\,630\,927\,705\,007\,454\,272\,817\,044\,369\,276\,793\,156\,189\,749\,051\,479 \right.$$
$$152\,377\,237\,075\,801\,586\,464\,248\,011\,390\,224\ \text{··· 23 ···}$$
$$\left.7\,240\,625\,540\,185\,427\,997\,955\,868\,026\,793\,112\,032\,095\,904\,831\,658\,727\,915\,618\,917\,540\,818\,814\,206\,589\,053\,500\,593 \right.$$
$$\left.\left.829\,351\,875\,943\,061\,563\,847\,973\,703\,215\,713\,923\right) / 1\,035\,631\,384\,445\,495\,145\,130\,590\,051\,160\,006\,381\right\}$$

Size in memory: 156.8 MB | + Show more | Show all | Iconize | Store full expression in notebook

In[159]:=
```
integerCheck2 = IntegerQ /@ reverseDivideSequence
```

Out[159]=

```
{False, False, False, False, False, False, False, False, False, False, True, False, False,
 True, True, False, False, False, False, False, False, False, False, False, False, False,
 True, False, False, True, True, False, False, False, False, False, False, False, False,
 False, False, False, True, ··· 997 914 ···, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False}
```

Size in memory: 8 MB | + Show more | Show all | Iconize | Store full expression in notebook

In[160]:=
```
Count[integerCheck2, True]
```

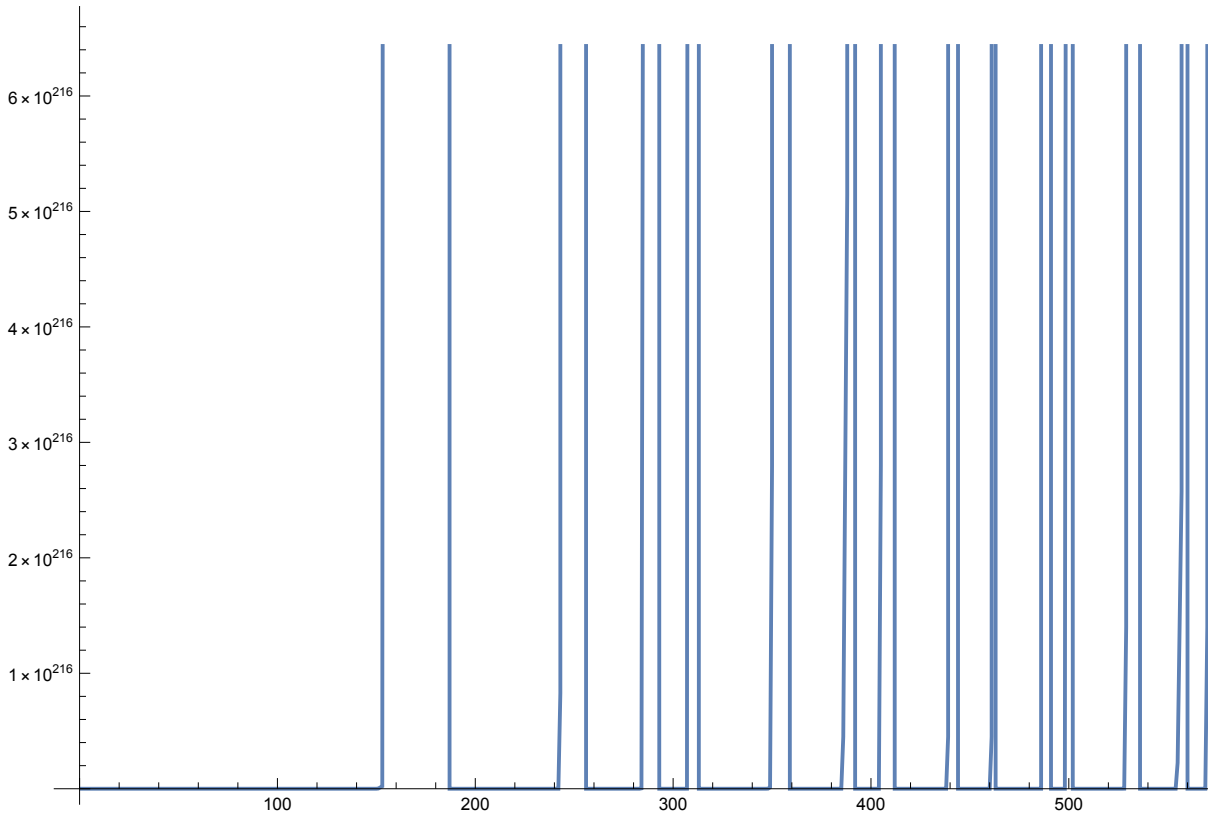
Out[160]=
```
627
```

In[118]:=
```
positionCheck2 = Position[integerCheck2, True] // Flatten;
```

In[133]:=
```
listIntegers1 = Table[reverseDivideSequence[[n]], {n, positionCheck2}];
```

In[134]:=
```
ListLinePlot[listIntegers1]
```

Out[134]=



In[132]:=
```
partition = Partition[positionCheck2, 3];
```

In[70]:=
```
reverseDivideSequence2 =
 Flatten[Table[tribonacci⟦k⟧ / sumTribonacciSequence⟦n⟧,
   {k, 1, Length[tribonacci]}, {n, 1, Length[sumTribonacciSequence]}]]
```

Out[70]=

$$\left\{\frac{1}{14}, \frac{1}{26}, \frac{1}{48}, \text{… 997 994 …}, \frac{203\,876\,926\ \text{…}\ 6\,431\,427\,846}{6\,122\,605\,986\ \text{…}\ 8\,446\,671\,509}, \right.$$

$$\frac{2\,038\,769\,264\,872\,658\,037\,531\,587\,434\,343\,800\,975\,252\,618\,554\ \text{…}\ 78\,107\,001\,187\,658\,703\,751\,886\,123\,127\,695\,947\,406\,431\,427\,846}{1\,126\,122\,809\,844\,198\,847\,564\,947\,518\,346\,763\,521},$$

$$\left(101\,938\,463\,243\,632\,901\,876\,579\,371\,717\,190\,048\,762\,630\,927\,705\,007\,454\,272\,817\,044\,369\,276\,793\,156\,189\,749\,051\,479\right.$$
$$152\,377\,237\,075\,801\,586\,464\,248\,011\,390\,224\ \text{… 23 …}$$
$$\left.7\,240\,625\,540\,185\,427\,997\,955\,868\,026\,793\,112\,032\,095\,904\,831\,658\,727\,915\,618\,917\,540\,818\,814\,206\,589\,053\,500\,593\right.$$
$$\left.829\,351\,875\,943\,061\,563\,847\,973\,703\,215\,713\,923\right) / 1\,035\,631\,384\,445\,495\,145\,130\,590\,051\,160\,006\,381\Big\}$$

Size in memory: 156.8 MB | + Show more | Show all | Iconize | Store full expression in notebook

In[85]:=
```
integerCheck3 = IntegerQ /@ reverseDivideSequence2
```

Out[85]=

In[86]:=
```
Tally[integerCheck3]
```

Out[86]=
```
{{False, 997 373}, {True, 627}}
```

In[119]:=
```
positionCheck3 = Position[integerCheck3, True] // Flatten;
```

In[131]:=
```
partitionCheck3 = Partition[positionCheck3, 3];
```

In[135]:=
```
listIntegers2 = Table[reverseDivideSequence2[[n]], {n, positionCheck3}];
```

In[136]:=

```
ListLinePlot[listIntegers2]
```

Out[136]=