

Hexi — Waves Exam 3

April 29, 2025

This exam tests your fluency with the core of the Wolfram Language, as it was presented in *An Elementary Introduction to the Wolfram Language, 3rd Edition (EIWL3)*, Sections 25-34 and 38-41. There is one problem with two or three parts corresponding to each section. **Tip: all of them are meant to be quick. If you get bogged down, move on.**

Directions:

After downloading this notebook, rename it with your first name in the filename. E.g., *Eli-Exam3.nb*, *Harper-Exam3.nb*, *Hexi-Exam3.nb*, *Jeremy-Exam3.nb*, *Rania-Exam3.nb*, *Tahm-Exam3.nb*, or *Walker-Exam3.nb*.

Then disconnect from the wifi and work the exam. Save your notebook early and often so that you don't lose work in progress.

Your answers always go into the Wolfram Language Input cells that begin with a comment, e.g.,

```
(* 1a *) foobar /@ Plus[Array]
```

All your answers should execute and re-execute without warnings or error messages.

You may refer to your downloaded copies of *EIWL3*, and anything else we developed in the course (like your cheat sheets!), but not to any web resources.

When you are done, save your notebook one last time, re-join the wifi, and then email it to me.

This exam was designed to require about 45 minutes, but if you need a full hour, that is ok. Everyone will stop at the one-hour mark.

1. Applying Functions (*EIWL3* Section 25)

(a)

Use **Map** with a *level-spec* to put a frame around each individual number in the array **Array[Plus, {10, 10}]** (we don't want frames around already-framed things — just one level of frames around the individual numbers).

In[162]:=

Map[Framed, Array[Plus, {10, 10}], 1]

Out[162]=

```
{ {2, 3, 4, 5, 6, 7, 8, 9, 10, 11}, {3, 4, 5, 6, 7, 8, 9, 10, 11, 12},
  {4, 5, 6, 7, 8, 9, 10, 11, 12, 13}, {5, 6, 7, 8, 9, 10, 11, 12, 13, 14},
  {6, 7, 8, 9, 10, 11, 12, 13, 14, 15}, {7, 8, 9, 10, 11, 12, 13, 14, 15, 16},
  {8, 9, 10, 11, 12, 13, 14, 15, 16, 17}, {9, 10, 11, 12, 13, 14, 15, 16, 17, 18},
  {10, 11, 12, 13, 14, 15, 16, 17, 18, 19}, {11, 12, 13, 14, 15, 16, 17, 18, 19, 20} }
```

(b)

Copy what you did in (a), but for this part, also turn the result into a grid using **Grid** and the “as an afterthought” syntax:

In[163]:=

{Map[Framed, Array[Plus, {10, 10}], 1]} // Grid

Out[163]=

| | | | | | | | | | |
|--------|-------|--------|-------|--------|-------|--------|-----|------|------|
| {2, 3, | {3, | {4, 5, | {5, | {6, 7, | {7, | {8, 9, | {9, | {10, | {11, |
| 4, 5, | 4, 5, | 6, 7, | 6, 7, | 8, 9, | 8, 9, | 10, | 10, | 11, | 12, |
| 6, 7, | 6, 7, | 8, 9, | 8, 9, | 10, | 10, | 11, | 11, | 12, | 13, |
| 8, 9, | 8, 9, | 10, | 10, | 11, | 11, | 12, | 12, | 13, | 14, |
| 10, | 10, | 11, | 11, | 12, | 12, | 13, | 13, | 14, | 15, |
| 11} | 11, | 12, | 12, | 13, | 13, | 14, | 14, | 15, | 16, |
| | 12} | 13} | 13, | 14, | 14, | 15, | 15, | 16, | 17, |
| | | | 14} | 15} | 15, | 16, | 16, | 17, | 18, |
| | | | | | 16} | 17} | 17, | 18, | 19, |
| | | | | | | | 18} | 19} | 20} |

2. Pure Anonymous Functions (EIWL3 Section 26)

(a)

Use the **#** and **&** notation to create an anonymous function that cubes whatever is given it, and then use **/@** to apply it to every member of the list **{1, 2, 3, 4, 5}**.

In[164]:=

#^3 & /@ {1, 2, 3, 4, 5}

Out[164]=

{1, 8, 27, 64, 125}

(b)

Use the **#1**, **#2**, and **&** notation to create an anonymous function that divides its first argument by its

second argument. Combine this with **Apply** and a *levelspec* to apply the function to $\{\{1,2\}, \{2,3\}, \{3,4\}, \{4,5\}\}$. Once you have this right, you will get $\{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}\}$.

```
In[165]:=
Apply[#1 / #2 &, {{1,2},{2,3},{3,4},{4,5}}, 1]

Out[165]=
{1/2, 2/3, 3/4, 4/5}
```

3. Applying Functions Repeatedly (EIWL3 Section 27)

(a)

Use **Nest** to apply **Factorial** twice to $\{1,2,3,4\}$. If you have this right, 620,448,401,733,239,439,360,000 will be one of the elements of your answer.

```
In[166]:=
Nest[Factorial, {1,2,3,4}, 2]

Out[166]=
{1, 2, 720, 620 448 401 733 239 439 360 000}
```

(b)

Use **NestList** to apply **Factorial** three times to $\{1,2,3\}$, as well as showing the results of doing it 0, 1, and 2 times. If you have this right, you will have an insanely large result at the third step. Do not go any higher, or I do not know what will happen to your computer.

```
In[167]:=
NestList[Factorial, {1,2,3}, 32]

Out[167]=
{{1, 2, 3}, {1, 2, 6}, {1, 2, 720}, {1, 2,
2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 785 727 541 837 850 835 \
631 156 947 382 240 678 577 958 130 457 082 619 920 575 892 247 259 536 641 565 162 052 015 \
873 791 984 587 740 832 529 105 244 690 388 811 884 123 764 341 191 951 045 505 346 658 616 \
243 271 940 197 113 909 845 536 727 278 537 099 345 629 855 586 719 369 774 070 003 700 430 \
783 758 997 420 676 784 016 967 207 846 280 629 229 032 107 161 669 867 260 548 988 445 514 \
257 193 985 499 448 939 594 496 064 045 132 362 140 265 986 193 073 249 369 770 477 606 067 \
680 670 176 491 669 403 034 819 961 881 455 625 195 592 566 918 830 825 514 942 947 596 537 \
274 845 624 628 824 234 526 597 789 737 740 896 466 553 992 435 928 786 212 515 967 483 220 \
976 029 505 696 699 927 284 670 563 747 137 533 019 248 313 587 076 125 412 683 415 860 129 \
447 566 011 455 420 749 589 952 563 543 068 288 634 631 084 965 650 682 771 552 996 256 790 \
845 235 702 552 186 222 358 130 016 700 834 523 443 236 821 935 793 184 701 956 510 729 781 \
804 354 173 890 560 727 428 048 583 995 919 729 021 726 612 291 298 420 516 067 579 036 232 \
337 699 453 964 191 475 175 567 557 695 392 233 803 056 825 308 599 977 441 675 784 352 815 \
913 461 340 394 604 901 269 542 028 838 347 101 363 733 824 484 506 660 093 348 484 440 711 \
931 292 537 694 657 354 337 375 724 772 230 181 534 032 647 177 531 984 537 341 478 674 327 \
048 457 983 786 618 703 257 405 938 924 215 709 695 994 630 557 521 063 203 263 493 209 220 \}}
```

738 320 923 356 309 923 267 504 401 701 760 572 026 010 829 288 042 335 606 643 089 888 710
297 380 797 578 013 056 049 576 342 838 683 057 190 662 205 291 174 822 510 536 697 756 603
029 574 043 387 983 471 518 552 602 805 333 866 357 139 101 046 336 419 769 097 397 432 285
994 219 837 046 979 109 956 303 389 604 675 889 865 795 711 176 566 670 039 156 748 153 115
943 980 043 625 399 399 731 203 066 490 601 325 311 304 719 028 898 491 856 203 766 669 164
468 791 125 249 193 754 425 845 895 000 311 561 682 974 304 641 142 538 074 897 281 723 375
955 380 661 719 801 404 677 935 614 793 635 266 265 683 339 509 760 000 000 000 000 000 000
000
000
000 000 000 000 000}, {1, 2,

2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 785 727 541 837 850 835
631 156 947 382 240 678 577 958 130 457 082 619 920 575 892 247 259 536 641 565 162 052 015
873 791 984 587 740 832 529 105 244 690 388 811 884 123 764 341 191 951 045 505 346 658 616
243 271 940 197 113 909 845 536 727 278 537 099 345 629 855 586 719 369 774 070 003 700 430
783 758 997 420 676 784 016 967 207 846 280 629 229 032 107 161 669 867 260 548 988 445 514
257 193 985 499 448 939 594 496 064 045 132 362 140 265 986 193 073 249 369 770 477 606 067
680 670 176 491 669 403 034 819 961 881 455 625 195 592 566 918 830 825 514 942 947 596 537
274 845 624 628 824 234 526 597 789 737 740 896 466 553 992 435 928 786 212 515 967 483 220
976 029 505 696 699 927 284 670 563 747 137 533 019 248 313 587 076 125 412 683 415 860 129
447 566 011 455 420 749 589 952 563 543 068 288 634 631 084 965 650 682 771 552 996 256 790
845 235 702 552 186 222 358 130 016 700 834 523 443 236 821 935 793 184 701 956 510 729 781
804 354 173 890 560 727 428 048 583 995 919 729 021 726 612 291 298 420 516 067 579 036 232
337 699 453 964 191 475 175 567 557 695 392 233 803 056 825 308 599 977 441 675 784 352 815
913 461 340 394 604 901 269 542 028 838 347 101 363 733 824 484 506 660 093 348 484 440 711
931 292 537 694 657 354 337 375 724 772 230 181 534 032 647 177 531 984 537 341 478 674 327
048 457 983 786 618 703 257 405 938 924 215 709 695 994 630 557 521 063 203 263 493 209 220
738 320 923 356 309 923 267 504 401 701 760 572 026 010 829 288 042 335 606 643 089 888 710
297 380 797 578 013 056 049 576 342 838 683 057 190 662 205 291 174 822 510 536 697 756 603
029 574 043 387 983 471 518 552 602 805 333 866 357 139 101 046 336 419 769 097 397 432 285
994 219 837 046 979 109 956 303 389 604 675 889 865 795 711 176 566 670 039 156 748 153 115
943 980 043 625 399 399 731 203 066 490 601 325 311 304 719 028 898 491 856 203 766 669 164
468 791 125 249 193 754 425 845 895 000 311 561 682 974 304 641 142 538 074 897 281 723 375
955 380 661 719 801 404 677 935 614 793 635 266 265 683 339 509 760 000 000 000 000 000 000
000
000
000 000 000 000 000 000 !}, {1, 2,

(2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 785 727 541 837 850 835
631 156 947 382 240 678 577 958 130 457 082 619 920 575 892 247 259 536 641 565 162 052 015
873 791 984 587 740 832 529 105 244 690 388 811 884 123 764 341 191 951 045 505 346 658 616
243 271 940 197 113 909 845 536 727 278 537 099 345 629 855 586 719 369 774 070 003 700 430
783 758 997 420 676 784 016 967 207 846 280 629 229 032 107 161 669 867 260 548 988 445 514
257 193 985 499 448 939 594 496 064 045 132 362 140 265 986 193 073 249 369 770 477 606 067
680 670 176 491 669 403 034 819 961 881 455 625 195 592 566 918 830 825 514 942 947 596 537
274 845 624 628 824 234 526 597 789 737 740 896 466 553 992 435 928 786 212 515 967 483 220

[illegible]

```

000 000 000 000 000 000 000 000 000 ! ) ! } , { 1 , 2 ,
(( (2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 785 727 541 837 850 \
835 631 156 947 382 240 678 577 958 130 457 082 619 920 575 892 247 259 536 641 565 \
162 052 015 873 791 984 587 740 832 529 105 244 690 388 811 884 123 764 341 191 951 \
045 505 346 658 616 243 271 940 197 113 909 845 536 727 278 537 099 345 629 855 586 \
719 369 774 070 003 700 430 783 758 997 420 676 784 016 967 207 846 280 629 229 032 \
107 161 669 867 260 548 988 445 514 257 193 985 499 448 939 594 496 064 045 132 362 \
140 265 986 193 073 249 369 770 477 606 067 680 670 176 491 669 403 034 819 961 881 \
455 625 195 592 566 918 830 825 514 942 947 596 537 274 845 624 628 824 234 526 597 \
789 737 740 896 466 553 992 435 928 786 212 515 967 483 220 976 029 505 696 699 927 \
284 670 563 747 137 533 019 248 313 587 076 125 412 683 415 860 129 447 566 011 455 \
420 749 589 952 563 543 068 288 634 631 084 965 650 682 771 552 996 256 790 845 235 \
702 552 186 222 358 130 016 700 834 523 443 236 821 935 793 184 701 956 510 729 781 \
804 354 173 890 560 727 428 048 583 995 919 729 021 726 612 291 298 420 516 067 579 \
036 232 337 699 453 964 191 475 175 567 557 695 392 233 803 056 825 308 599 977 441 \
675 784 352 815 913 461 340 394 604 901 269 542 028 838 347 101 363 733 824 484 506 \
660 093 348 484 440 711 931 292 537 694 657 354 337 375 724 772 230 181 534 032 647 \
177 531 984 537 341 478 674 327 048 457 983 786 618 703 257 405 938 924 215 709 695 \
994 630 557 521 063 203 263 493 209 220 738 320 923 356 309 923 267 504 401 701 760 \
572 026 010 829 288 042 335 606 643 089 888 710 297 380 797 578 013 056 049 576 342 \
838 683 057 190 662 205 291 174 822 510 536 697 756 603 029 574 043 387 983 471 518 \
552 602 805 333 866 357 139 101 046 336 419 769 097 397 432 285 994 219 837 046 979 \
109 956 303 389 604 675 889 865 795 711 176 566 670 039 156 748 153 115 943 980 043 \
625 399 399 731 203 066 490 601 325 311 304 719 028 898 491 856 203 766 669 164 468 \
791 125 249 193 754 425 845 895 000 311 561 682 974 304 641 142 538 074 897 281 723 \
375 955 380 661 719 801 404 677 935 614 793 635 266 265 683 339 509 760 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 ! ) ! } , { 1 , 2 ,
((( (2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 785 727 541 837 850 \
835 631 156 947 382 240 678 577 958 130 457 082 619 920 575 892 247 259 536 641 565 \
162 052 015 873 791 984 587 740 832 529 105 244 690 388 811 884 123 764 341 191 951 \
045 505 346 658 616 243 271 940 197 113 909 845 536 727 278 537 099 345 629 855 586 \
719 369 774 070 003 700 430 783 758 997 420 676 784 016 967 207 846 280 629 229 032 \
107 161 669 867 260 548 988 445 514 257 193 985 499 448 939 594 496 064 045 132 362 \
140 265 986 193 073 249 369 770 477 606 067 680 670 176 491 669 403 034 819 961 881 \
455 625 195 592 566 918 830 825 514 942 947 596 537 274 845 624 628 824 234 526 597 \
789 737 740 896 466 553 992 435 928 786 212 515 967 483 220 976 029 505 696 699 927 \
284 670 563 747 137 533 019 248 313 587 076 125 412 683 415 860 129 447 566 011 455 \
420 749 589 952 563 543 068 288 634 631 084 965 650 682 771 552 996 256 790 845 235 \
702 552 186 222 358 130 016 700 834 523 443 236 821 935 793 184 701 956 510 729 781 \
804 354 173 890 560 727 428 048 583 995 919 729 021 726 612 291 298 420 516 067 579 \
036 232 337 699 453 964 191 475 175 567 557 695 392 233 803 056 825 308 599 977 441 \
675 784 352 815 913 461 340 394 604 901 269 542 028 838 347 101 363 733 824 484 506 \

```

```
660 093 348 484 440 711 931 292 537 694 657 354 337 375 724 772 230 181 534 032 647 :  
177 531 984 537 341 478 674 327 048 457 983 786 618 703 257 405 938 924 215 709 695 :  
994 630 557 521 063 203 263 493 209 220 738 320 923 356 309 923 267 504 401 701 760 :  
572 026 010 829 288 042 335 606 643 089 888 710 297 380 797 578 013 056 049 576 342 :  
838 683 057 190 662 205 291 174 822 510 536 697 756 603 029 574 043 387 983 471 518 :  
552 602 805 333 866 357 139 101 046 336 419 769 097 397 432 285 994 219 837 046 979 :  
109 956 303 389 604 675 889 865 795 711 176 566 670 039 156 748 153 115 943 980 043 :  
625 399 399 731 203 066 490 601 325 311 304 719 028 898 491 856 203 766 669 164 468 :  
791 125 249 193 754 425 845 895 000 311 561 682 974 304 641 142 538 074 897 281 723 :  
375 955 380 661 719 801 404 677 935 614 793 635 266 265 683 339 509 760 000 000 000 :  
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 :  
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 :  
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 ! ) ! ) ! } ,
```

 $\{1,$

2,

[illegible]

$\{1, 2,$

420 749 589 952 563 543 068 288 634 631 084 965 650 682 771 552 996 256 790 :
845 235 702 552 186 222 358 130 016 700 834 523 443 236 821 935 793 184 701 :
956 510 729 781 804 354 173 890 560 727 428 048 583 995 919 729 021 726 612 :
291 298 420 516 067 579 036 232 337 699 453 964 191 475 175 567 557 695 392 :
233 803 056 825 308 599 977 441 675 784 352 815 913 461 340 394 604 901 269 :
542 028 838 347 101 363 733 824 484 506 660 093 348 484 440 711 931 292 537 :
694 657 354 337 375 724 772 230 181 534 032 647 177 531 984 537 341 478 674 :
327 048 457 983 786 618 703 257 405 938 924 215 709 695 994 630 557 521 063 :
203 263 493 209 220 738 320 923 356 309 923 267 504 401 701 760 572 026 010 :
829 288 042 335 606 643 089 888 710 297 380 797 578 013 056 049 576 342 838 :
683 057 190 662 205 291 174 822 510 536 697 756 603 029 574 043 387 983 471 :
518 552 602 805 333 866 357 139 101 046 336 419 769 097 397 432 285 994 219 :
837 046 979 109 956 303 389 604 675 889 865 795 711 176 566 670 039 156 748 :
153 115 943 980 043 625 399 399 731 203 066 490 601 325 311 304 719 028 898 :
491 856 203 766 669 164 468 791 125 249 193 754 425 845 895 000 311 561 682 :
974 304 641 142 538 074 897 281 723 375 955 380 661 719 801 404 677 935 614 :
793 635 266 265 683 339 509 760 000 000 000 000 000 000 000 000 000 000 :
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 :
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 :
000 000 000 000 000 000 000 000 000 !) !) !) !) !) ,

 $\{1,$

2,

```
(((((2601218943565795100204903227081043611191521875016945785727541837
850835631156947382240678577958130457082619920575892247
259536641565162052015873791984587740832529105244690388
811884123764341191951045505346658616243271940197113909
845536727278537099345629855586719369774070003700430783
758997420676784016967207846280629229032107161669867260
548988445514257193985499448939594496064045132362140265
986193073249369770477606067680670176491669403034819961
881455625195592566918830825514942947596537274845624628
824234526597789737740896466553992435928786212515967483
220976029505696699927284670563747137533019248313587076
125412683415860129447566011455420749589952563543068288
634631084965650682771552996256790845235702552186222358
130016700834523443236821935793184701956510729781804354
173890560727428048583995919729021726612291298420516067
579036232337699453964191475175567557695392233803056825
308599977441675784352815913461340394604901269542028838
347101363733824484506660093348484440711931292537694657
354337375724772230181534032647177531984537341478674327
048457983786618703257405938924215709695994630557521063
203263493209220738320923356309923267504401701760572026
010829288042335606643089888710297380797578013056049576
```

```

342 838 683 057 190 662 205 291 174 822 510 536 697 756 603 029 574 043 \
387 983 471 518 552 602 805 333 866 357 139 101 046 336 419 769 097 397 \
432 285 994 219 837 046 979 109 956 303 389 604 675 889 865 795 711 176 \
566 670 039 156 748 153 115 943 980 043 625 399 399 731 203 066 490 601 \
325 311 304 719 028 898 491 856 203 766 669 164 468 791 125 249 193 754 \
425 845 895 000 311 561 682 974 304 641 142 538 074 897 281 723 375 955 \
380 661 719 801 404 677 935 614 793 635 266 265 683 339 509 760 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 ! ) ! ) ! ) ! ) ! ) ! } , { 1 , 2 ,
((((((((2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 785 727 541 \
837 850 835 631 156 947 382 240 678 577 958 130 457 082 619 920 575 \
892 247 259 536 641 565 162 052 015 873 791 984 587 740 832 529 105 \
244 690 388 811 884 123 764 341 191 951 045 505 346 658 616 243 271 \
940 197 113 909 845 536 727 278 537 099 345 629 855 586 719 369 774 \
070 003 700 430 783 758 997 420 676 784 016 967 207 846 280 629 229 \
032 107 161 669 867 260 548 988 445 514 257 193 985 499 448 939 594 \
496 064 045 132 362 140 265 986 193 073 249 369 770 477 606 067 680 \
670 176 491 669 403 034 819 961 881 455 625 195 592 566 918 830 825 \
514 942 947 596 537 274 845 624 628 824 234 526 597 789 737 740 896 \
466 553 992 435 928 786 212 515 967 483 220 976 029 505 696 699 927 \
284 670 563 747 137 533 019 248 313 587 076 125 412 683 415 860 129 \
447 566 011 455 420 749 589 952 563 543 068 288 634 631 084 965 650 \
682 771 552 996 256 790 845 235 702 552 186 222 358 130 016 700 834 \
523 443 236 821 935 793 184 701 956 510 729 781 804 354 173 890 560 \
727 428 048 583 995 919 729 021 726 612 291 298 420 516 067 579 036 \
232 337 699 453 964 191 475 175 567 557 695 392 233 803 056 825 308 \
599 977 441 675 784 352 815 913 461 340 394 604 901 269 542 028 838 \
347 101 363 733 824 484 506 660 093 348 484 440 711 931 292 537 694 \
657 354 337 375 724 772 230 181 534 032 647 177 531 984 537 341 478 \
674 327 048 457 983 786 618 703 257 405 938 924 215 709 695 994 630 \
557 521 063 203 263 493 209 220 738 320 923 356 309 923 267 504 401 \
701 760 572 026 010 829 288 042 335 606 643 089 888 710 297 380 797 \
578 013 056 049 576 342 838 683 057 190 662 205 291 174 822 510 536 \
697 756 603 029 574 043 387 983 471 518 552 602 805 333 866 357 139 \
101 046 336 419 769 097 397 432 285 994 219 837 046 979 109 956 303 \
389 604 675 889 865 795 711 176 566 670 039 156 748 153 115 943 980 \
043 625 399 399 731 203 066 490 601 325 311 304 719 028 898 491 856 \
203 766 669 164 468 791 125 249 193 754 425 845 895 000 311 561 682 \
974 304 641 142 538 074 897 281 723 375 955 380 661 719 801 404 677 \
935 614 793 635 266 265 683 339 509 760 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000

```

[illegible]

000 000 000 000 000 000 000 !) !) !) !) !) !) !) !) !) ! } ,

 $\{1,$

2,

```
(((((2601218943565795100204903227081043611191521875016945785727 \
541837850835631156947382240678577958130457082 \
619920575892247259536641565162052015873791984 \
587740832529105244690388811884123764341191951 \
045505346658616243271940197113909845536727278 \
537099345629855586719369774070003700430783758 \
997420676784016967207846280629229032107161669 \
867260548988445514257193985499448939594496064 \
045132362140265986193073249369770477606067680 \
670176491669403034819961881455625195592566918 \
830825514942947596537274845624628824234526597 \
789737740896466553992435928786212515967483220 \
976029505696699927284670563747137533019248313 \
587076125412683415860129447566011455420749589 \
952563543068288634631084965650682771552996256 \
```

Hexi-Exam3.nb

790 845 235 702 552 186 222 358 130 016 700 834 523 443 236
821 935 793 184 701 956 510 729 781 804 354 173 890 560 727
428 048 583 995 919 729 021 726 612 291 298 420 516 067 579
036 232 337 699 453 964 191 475 175 567 557 695 392 233 803
056 825 308 599 977 441 675 784 352 815 913 461 340 394 604
901 269 542 028 838 347 101 363 733 824 484 506 660 093 348
484 440 711 931 292 537 694 657 354 337 375 724 772 230 181
534 032 647 177 531 984 537 341 478 674 327 048 457 983 786
618 703 257 405 938 924 215 709 695 994 630 557 521 063 203
263 493 209 220 738 320 923 356 309 923 267 504 401 701 760
572 026 010 829 288 042 335 606 643 089 888 710 297 380 797
578 013 056 049 576 342 838 683 057 190 662 205 291 174 822
510 536 697 756 603 029 574 043 387 983 471 518 552 602 805
333 866 357 139 101 046 336 419 769 097 397 432 285 994 219
837 046 979 109 956 303 389 604 675 889 865 795 711 176 566
670 039 156 748 153 115 943 980 043 625 399 399 731 203 066
490 601 325 311 304 719 028 898 491 856 203 766 669 164 468
791 125 249 193 754 425 845 895 000 311 561 682 974 304 641
142 538 074 897 281 723 375 955 380 661 719 801 404 677 935
614 793 635 266 265 683 339 509 760 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000

$$) !) ! \} ,$$
 $\{1,$

2,

```
(((((2601218943565795100204903227081043611191521875016945785727\
541837850835631156947382240678577958130457\
082619920575892247259536641565162052015873\
791984587740832529105244690388811884123764\
341191951045505346658616243271940197113909\
845536727278537099345629855586719369774070\
003700430783758997420676784016967207846280\
629229032107161669867260548988445514257193\
985499448939594496064045132362140265986193\
073249369770477606067680670176491669403034\
819961881455625195592566918830825514942947\
596537274845624628824234526597789737740896\
466553992435928786212515967483220976029505\
696699927284670563747137533019248313587076\
125412683415860129447566011455420749589952\
563543068288634631084965650682771552996256\
790845235702552186222358130016700834523443\
```

```

236 821 935 793 184 701 956 510 729 781 804 354 173 890 \
560 727 428 048 583 995 919 729 021 726 612 291 298 420 \
516 067 579 036 232 337 699 453 964 191 475 175 567 557 \
695 392 233 803 056 825 308 599 977 441 675 784 352 815 \
913 461 340 394 604 901 269 542 028 838 347 101 363 733 \
824 484 506 660 093 348 484 440 711 931 292 537 694 657 \
354 337 375 724 772 230 181 534 032 647 177 531 984 537 \
341 478 674 327 048 457 983 786 618 703 257 405 938 924 \
215 709 695 994 630 557 521 063 203 263 493 209 220 738 \
320 923 356 309 923 267 504 401 701 760 572 026 010 829 \
288 042 335 606 643 089 888 710 297 380 797 578 013 056 \
049 576 342 838 683 057 190 662 205 291 174 822 510 536 \
697 756 603 029 574 043 387 983 471 518 552 602 805 333 \
866 357 139 101 046 336 419 769 097 397 432 285 994 219 \
837 046 979 109 956 303 389 604 675 889 865 795 711 176 \
566 670 039 156 748 153 115 943 980 043 625 399 399 731 \
203 066 490 601 325 311 304 719 028 898 491 856 203 766 \
669 164 468 791 125 249 193 754 425 845 895 000 311 561 \
682 974 304 641 142 538 074 897 281 723 375 955 380 661 \
719 801 404 677 935 614 793 635 266 265 683 339 509 760 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! } ,

```

{1,

2,

```

(((((((((((((((2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 785 727 \
541 837 850 835 631 156 947 382 240 678 577 958 130 \
457 082 619 920 575 892 247 259 536 641 565 162 052 \
015 873 791 984 587 740 832 529 105 244 690 388 811 \
884 123 764 341 191 951 045 505 346 658 616 243 271 \
940 197 113 909 845 536 727 278 537 099 345 629 855 \
586 719 369 774 070 003 700 430 783 758 997 420 676 \
784 016 967 207 846 280 629 229 032 107 161 669 867 \
260 548 988 445 514 257 193 985 499 448 939 594 496 \
064 045 132 362 140 265 986 193 073 249 369 770 477 \
606 067 680 670 176 491 669 403 034 819 961 881 455 \
625 195 592 566 918 830 825 514 942 947 596 537 274 \
845 624 628 824 234 526 597 789 737 740 896 466 553 \
992 435 928 786 212 515 967 483 220 976 029 505 696 \
699 927 284 670 563 747 137 533 019 248 313 587 076 \
125 412 683 415 860 129 447 566 011 455 420 749 589 \
952 563 543 068 288 634 631 084 965 650 682 771 552 \

```



```

553 992 435 928 786 212 515 967 483 220 976 029 505 \
696 699 927 284 670 563 747 137 533 019 248 313 587 \
076 125 412 683 415 860 129 447 566 011 455 420 749 \
589 952 563 543 068 288 634 631 084 965 650 682 771 \
552 996 256 790 845 235 702 552 186 222 358 130 016 \
700 834 523 443 236 821 935 793 184 701 956 510 729 \
781 804 354 173 890 560 727 428 048 583 995 919 729 \
021 726 612 291 298 420 516 067 579 036 232 337 699 \
453 964 191 475 175 567 557 695 392 233 803 056 825 \
308 599 977 441 675 784 352 815 913 461 340 394 604 \
901 269 542 028 838 347 101 363 733 824 484 506 660 \
093 348 484 440 711 931 292 537 694 657 354 337 375 \
724 772 230 181 534 032 647 177 531 984 537 341 478 \
674 327 048 457 983 786 618 703 257 405 938 924 215 \
709 695 994 630 557 521 063 203 263 493 209 220 738 \
320 923 356 309 923 267 504 401 701 760 572 026 010 \
829 288 042 335 606 643 089 888 710 297 380 797 578 \
013 056 049 576 342 838 683 057 190 662 205 291 174 \
822 510 536 697 756 603 029 574 043 387 983 471 518 \
552 602 805 333 866 357 139 101 046 336 419 769 097 \
397 432 285 994 219 837 046 979 109 956 303 389 604 \
675 889 865 795 711 176 566 670 039 156 748 153 115 \
943 980 043 625 399 399 731 203 066 490 601 325 311 \
304 719 028 898 491 856 203 766 669 164 468 791 125 \
249 193 754 425 845 895 000 311 561 682 974 304 641 \
142 538 074 897 281 723 375 955 380 661 719 801 404 \
677 935 614 793 635 266 265 683 339 509 760 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! )

```

```
! ) ! ) ! ) ! } ,
```

```
{1,
```

```
2,
```

```

((((((((((((((((((2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 785 \
727 541 837 850 835 631 156 947 382 240 678 577 \
958 130 457 082 619 920 575 892 247 259 536 641 \
565 162 052 015 873 791 984 587 740 832 529 105 \
244 690 388 811 884 123 764 341 191 951 045 505 \
346 658 616 243 271 940 197 113 909 845 536 727 \
278 537 099 345 629 855 586 719 369 774 070 003 \
700 430 783 758 997 420 676 784 016 967 207 846 \
280 629 229 032 107 161 669 867 260 548 988 445 \

```

```

514 257 193 985 499 448 939 594 496 064 045 132 \
362 140 265 986 193 073 249 369 770 477 606 067 \
680 670 176 491 669 403 034 819 961 881 455 625 \
195 592 566 918 830 825 514 942 947 596 537 274 \
845 624 628 824 234 526 597 789 737 740 896 466 \
553 992 435 928 786 212 515 967 483 220 976 029 \
505 696 699 927 284 670 563 747 137 533 019 248 \
313 587 076 125 412 683 415 860 129 447 566 011 \
455 420 749 589 952 563 543 068 288 634 631 084 \
965 650 682 771 552 996 256 790 845 235 702 552 \
186 222 358 130 016 700 834 523 443 236 821 935 \
793 184 701 956 510 729 781 804 354 173 890 560 \
727 428 048 583 995 919 729 021 726 612 291 298 \
420 516 067 579 036 232 337 699 453 964 191 475 \
175 567 557 695 392 233 803 056 825 308 599 977 \
441 675 784 352 815 913 461 340 394 604 901 269 \
542 028 838 347 101 363 733 824 484 506 660 093 \
348 484 440 711 931 292 537 694 657 354 337 375 \
724 772 230 181 534 032 647 177 531 984 537 341 \
478 674 327 048 457 983 786 618 703 257 405 938 \
924 215 709 695 994 630 557 521 063 203 263 493 \
209 220 738 320 923 356 309 923 267 504 401 701 \
760 572 026 010 829 288 042 335 606 643 089 888 \
710 297 380 797 578 013 056 049 576 342 838 683 \
057 190 662 205 291 174 822 510 536 697 756 603 \
029 574 043 387 983 471 518 552 602 805 333 866 \
357 139 101 046 336 419 769 097 397 432 285 994 \
219 837 046 979 109 956 303 389 604 675 889 865 \
795 711 176 566 670 039 156 748 153 115 943 980 \
043 625 399 399 731 203 066 490 601 325 311 304 \
719 028 898 491 856 203 766 669 164 468 791 125 \
249 193 754 425 845 895 000 311 561 682 974 304 \
641 142 538 074 897 281 723 375 955 380 661 719 \
801 404 677 935 614 793 635 266 265 683 339 509 \
760 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \

```

```

) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! } ,

```

```

{1, 2,

```

```

((((((((((((((((((2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 785 \
727 541 837 850 835 631 156 947 382 240 678 577 \
958 130 457 082 619 920 575 892 247 259 536 641 \

```

565 162 052 015 873 791 984 587 740 832 529 105 %
 244 690 388 811 884 123 764 341 191 951 045 505 %
 346 658 616 243 271 940 197 113 909 845 536 727 %
 278 537 099 345 629 855 586 719 369 774 070 003 %
 700 430 783 758 997 420 676 784 016 967 207 846 %
 280 629 229 032 107 161 669 867 260 548 988 445 %
 514 257 193 985 499 448 939 594 496 064 045 132 %
 362 140 265 986 193 073 249 369 770 477 606 067 %
 680 670 176 491 669 403 034 819 961 881 455 625 %
 195 592 566 918 830 825 514 942 947 596 537 274 %
 845 624 628 824 234 526 597 789 737 740 896 466 %
 553 992 435 928 786 212 515 967 483 220 976 029 %
 505 696 699 927 284 670 563 747 137 533 019 248 %
 313 587 076 125 412 683 415 860 129 447 566 011 %
 455 420 749 589 952 563 543 068 288 634 631 084 %
 965 650 682 771 552 996 256 790 845 235 702 552 %
 186 222 358 130 016 700 834 523 443 236 821 935 %
 793 184 701 956 510 729 781 804 354 173 890 560 %
 727 428 048 583 995 919 729 021 726 612 291 298 %
 420 516 067 579 036 232 337 699 453 964 191 475 %
 175 567 557 695 392 233 803 056 825 308 599 977 %
 441 675 784 352 815 913 461 340 394 604 901 269 %
 542 028 838 347 101 363 733 824 484 506 660 093 %
 348 484 440 711 931 292 537 694 657 354 337 375 %
 724 772 230 181 534 032 647 177 531 984 537 341 %
 478 674 327 048 457 983 786 618 703 257 405 938 %
 924 215 709 695 994 630 557 521 063 203 263 493 %
 209 220 738 320 923 356 309 923 267 504 401 701 %
 760 572 026 010 829 288 042 335 606 643 089 888 %
 710 297 380 797 578 013 056 049 576 342 838 683 %
 057 190 662 205 291 174 822 510 536 697 756 603 %
 029 574 043 387 983 471 518 552 602 805 333 866 %
 357 139 101 046 336 419 769 097 397 432 285 994 %
 219 837 046 979 109 956 303 389 604 675 889 865 %
 795 711 176 566 670 039 156 748 153 115 943 980 %
 043 625 399 399 731 203 066 490 601 325 311 304 %
 719 028 898 491 856 203 766 669 164 468 791 125 %
 249 193 754 425 845 895 000 311 561 682 974 304 %
 641 142 538 074 897 281 723 375 955 380 661 719 %
 801 404 677 935 614 793 635 266 265 683 339 509 %
 760 000 000 000 000 000 000 000 000 000 000 000 %
 000 000 000 000 000 000 000 000 000 000 000 000 %
 000 000 000 000 000 000 000 000 000 000 000 000 %
 000 000 000 000 000 000 000 000 000 000 000 000 %


```

304 641 142 538 074 897 281 723 375 955 380 661 \
719 801 404 677 935 614 793 635 266 265 683 339 \
509 760 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! )

```

```

! ) ! ) ! } , { 1 , 2 ,

```

```

( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( 2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 \
785 727 541 837 850 835 631 156 947 382 240 678 \
577 958 130 457 082 619 920 575 892 247 259 536 \
641 565 162 052 015 873 791 984 587 740 832 529 \
105 244 690 388 811 884 123 764 341 191 951 045 \
505 346 658 616 243 271 940 197 113 909 845 536 \
727 278 537 099 345 629 855 586 719 369 774 070 \
003 700 430 783 758 997 420 676 784 016 967 207 \
846 280 629 229 032 107 161 669 867 260 548 988 \
445 514 257 193 985 499 448 939 594 496 064 045 \
132 362 140 265 986 193 073 249 369 770 477 606 \
067 680 670 176 491 669 403 034 819 961 881 455 \
625 195 592 566 918 830 825 514 942 947 596 537 \
274 845 624 628 824 234 526 597 789 737 740 896 \
466 553 992 435 928 786 212 515 967 483 220 976 \
029 505 696 699 927 284 670 563 747 137 533 019 \
248 313 587 076 125 412 683 415 860 129 447 566 \
011 455 420 749 589 952 563 543 068 288 634 631 \
084 965 650 682 771 552 996 256 790 845 235 702 \
552 186 222 358 130 016 700 834 523 443 236 821 \
935 793 184 701 956 510 729 781 804 354 173 890 \
560 727 428 048 583 995 919 729 021 726 612 291 \
298 420 516 067 579 036 232 337 699 453 964 191 \
475 175 567 557 695 392 233 803 056 825 308 599 \
977 441 675 784 352 815 913 461 340 394 604 901 \
269 542 028 838 347 101 363 733 824 484 506 660 \
093 348 484 440 711 931 292 537 694 657 354 337 \
375 724 772 230 181 534 032 647 177 531 984 537 \
341 478 674 327 048 457 983 786 618 703 257 405 \
938 924 215 709 695 994 630 557 521 063 203 263 \
493 209 220 738 320 923 356 309 923 267 504 401 \
701 760 572 026 010 829 288 042 335 606 643 089 \
888 710 297 380 797 578 013 056 049 576 342 838 \
683 057 190 662 205 291 174 822 510 536 697 756 \
603 029 574 043 387 983 471 518 552 602 805 333 \

```


938 924 215 710 695 994 630 557 521 063 203 263 !
493 209 220 738 320 923 356 309 923 267 504 401 !
701 760 572 026 010 829 288 042 335 606 643 089 !
888 710 297 380 797 578 013 056 049 576 342 838 !
683 057 190 662 205 291 174 822 510 536 697 756 !
603 029 574 043 387 983 471 518 552 602 805 333 !
866 357 139 101 046 336 419 769 097 397 432 285 !
994 219 837 046 979 109 956 303 389 604 675 889 !
865 795 711 176 566 670 039 156 748 153 115 943 !
980 043 625 399 399 731 203 066 490 601 325 311 !
304 719 028 898 491 856 203 766 669 164 468 791 !
125 249 193 754 425 845 895 000 311 561 682 974 !
304 641 142 538 074 897 281 723 375 955 380 661 !
719 801 404 677 935 614 793 635 266 265 683 339 !
509 760 000 000 000 000 000 000 000 000 000 000 !
000 000 000 000 000 000 000 000 000 000 000 000 !
000 000 000 000 000 000 000 000 000 000 000 000 !
000 000 000 000 000 000 000 000 000 000 000 000 !
000 000 000 000 000 000 000 000 000 000 000 000 !
000 000 000 000 000 000 000 000 000 000 000 000 !
000 !) !) !) !) !) !) !) !) !) !) !) !) !

$$!) !) !) !) !\}, \{1, 2,$$

(((((2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 945 785 727 541 837 850 835 631 156 947 382 240 678 577 958 130 457 082 619 920 575 892 247 259 536 641 565 162 052 015 873 791 984 587 740 832 529 105 244 690 388 811 884 123 764 341 191 951 045 505 346 658 616 243 271 940 197 113 909 845 536 727 278 537 099 345 629 855 586 719 369 774 070 003 700 430 783 758 997 420 676 784 016 967 207 846 280 629 229 032 107 161 669 867 260 548 988 445 514 257 193 985 499 448 939 594 496 064 045 132 362 140 265 986 193 073 249 369 770 477 606 067 680 670 176 491 669 403 034 819 961 881 455 625 195 592 566 918 830 825 514 942 947 596 537 274 845 624 628 824 234 526 597 789 737 740 896 466 553 992 435 928 786 212 515 967 483 220 976 029 505 696 699 927 284 670 563 747 137 533 019 248 313 587 076 125 412 683 415 860 129 447 566 011 455 420 749 589 952 563 543 068 288 634 631 084 965 650 682 771 552 996 256 790 845 235 702 552 186 222 358 130 016 700 834 523 443 236 821 935 793 184 701 956 510 729 781 804 354 173 890 560 727 428 048 583 995 919 729 021 726 612 291 298 420 516 067 579 036 232 337 699 453 964 191


```

566 011 455 420 749 589 952 563 543 068 288 634 \
631 084 965 650 682 771 552 996 256 790 845 235 \
702 552 186 222 358 130 016 700 834 523 443 236 \
821 935 793 184 701 956 510 729 781 804 354 173 \
890 560 727 428 048 583 995 919 729 021 726 612 \
291 298 420 516 067 579 036 232 337 699 453 964 \
191 475 175 567 557 695 392 233 803 056 825 308 \
599 977 441 675 784 352 815 913 461 340 394 604 \
901 269 542 028 838 347 101 363 733 824 484 506 \
660 093 348 484 440 711 931 292 537 694 657 354 \
337 375 724 772 230 181 534 032 647 177 531 984 \
537 341 478 674 327 048 457 983 786 618 703 257 \
405 938 924 215 709 695 994 630 557 521 063 203 \
263 493 209 220 738 320 923 356 309 923 267 504 \
401 701 760 572 026 010 829 288 042 335 606 643 \
089 888 710 297 380 797 578 013 056 049 576 342 \
838 683 057 190 662 205 291 174 822 510 536 697 \
756 603 029 574 043 387 983 471 518 552 602 805 \
333 866 357 139 101 046 336 419 769 097 397 432 \
285 994 219 837 046 979 109 956 303 389 604 675 \
889 865 795 711 176 566 670 039 156 748 153 115 \
943 980 043 625 399 399 731 203 066 490 601 325 \
311 304 719 028 898 491 856 203 766 669 164 468 \
791 125 249 193 754 425 845 895 000 311 561 682 \
974 304 641 142 538 074 897 281 723 375 955 380 \
661 719 801 404 677 935 614 793 635 266 265 683 \
339 509 760 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) !
) ! ) ! ) ! ) ! ) ! ) ! } , { 1 , 2 ,
((((((((((((((((((((((((((((((((((((2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 016 \
945 785 727 541 837 850 835 631 156 947 382 240 \
678 577 958 130 457 082 619 920 575 892 247 259 \
536 641 565 162 052 015 873 791 984 587 740 832 \
529 105 244 690 388 811 884 123 764 341 191 951 \
045 505 346 658 616 243 271 940 197 113 909 845 \
536 727 278 537 099 345 629 855 586 719 369 774 \
070 003 700 430 783 758 997 420 676 784 016 967 \
207 846 280 629 229 032 107 161 669 867 260 548 \
988 445 514 257 193 985 499 448 939 594 496 064 \
045 132 362 140 265 986 193 073 249 369 770 477 \

```


000 000 !) !) !) !) !) !) !) !) !) !) !) !) !

```

) ! ) ! ) ! ) ! ) ! ) ! } , { 1 , 2 ,
((((((((((((((((((((((((((((((((((((((((2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 \
016 945 785 727 541 837 850 835 631 156 947 382 \
240 678 577 958 130 457 082 619 920 575 892 247 \
259 536 641 565 162 052 015 873 791 984 587 740 \
832 529 105 244 690 388 811 884 123 764 341 191 \
951 045 505 346 658 616 243 271 940 197 113 909 \
845 536 727 278 537 099 345 629 855 586 719 369 \
774 070 003 700 430 783 758 997 420 676 784 016 \
967 207 846 280 629 229 032 107 161 669 867 260 \
548 988 445 514 257 193 985 499 448 939 594 496 \
064 045 132 362 140 265 986 193 073 249 369 770 \
477 606 067 680 670 176 491 669 403 034 819 961 \
881 455 625 195 592 566 918 830 825 514 942 947 \
596 537 274 845 624 628 824 234 526 597 789 737 \
740 896 466 553 992 435 928 786 212 515 967 483 \
220 976 029 505 696 699 927 284 670 563 747 137 \
533 019 248 313 587 076 125 412 683 415 860 129 \
447 566 011 455 420 749 589 952 563 543 068 288 \
634 631 084 965 650 682 771 552 996 256 790 845 \
235 702 552 186 222 358 130 016 700 834 523 443 \
236 821 935 793 184 701 956 510 729 781 804 354 \
173 890 560 727 428 048 583 995 919 729 021 726 \
612 291 298 420 516 067 579 036 232 337 699 453 \
964 191 475 175 567 557 695 392 233 803 056 825 \
308 599 977 441 675 784 352 815 913 461 340 394 \
604 901 269 542 028 838 347 101 363 733 824 484 \
506 660 093 348 484 440 711 931 292 537 694 657 \
354 337 375 724 772 230 181 534 032 647 177 531 \
984 537 341 478 674 327 048 457 983 786 618 703 \
257 405 938 924 215 709 695 994 630 557 521 063 \
203 263 493 209 220 738 320 923 356 309 923 267 \
504 401 701 760 572 026 010 829 288 042 335 606 \
643 089 888 710 297 380 797 578 013 056 049 576 \
342 838 683 057 190 662 205 291 174 822 510 536 \
697 756 603 029 574 043 387 983 471 518 552 602 \
805 333 866 357 139 101 046 336 419 769 097 397 \
432 285 994 219 837 046 979 109 956 303 389 604 \
675 889 865 795 711 176 566 670 039 156 748 153 \
115 943 980 043 625 399 399 731 203 066 490 601 \
325 311 304 719 028 898 491 856 203 766 669 164 \
468 791 125 249 193 754 425 845 895 000 311 561 \
682 974 304 641 142 538 074 897 281 723 375 955 \
380 661 719 801 404 677 935 614 793 635 266 265 \

```

!) !) !) !) !) !) !) !) !) !) !) !) !} ,

(((((2601218943565795100204903227081043611191521875
016945785727541837850835631156947382
240678577958130457082619920575892247
259536641565162052015873791984587740
832529105244690388811884123764341191
951045505346658616243271940197113909
845536727278537099345629855586719369
774070003700430783758997420676784016
967207846280629229032107161669867260
548988445514257193985499448939594496
064045132362140265986193073249369770
477606067680670176491669403034819961
881455625195592566918830825514942947
596537274845624628824234526597789737
740896466553992435928786212515967483
220976029505696699927284670563747137
533019248313587076125412683415860129
447566011455420749589952563543068288
634631084965650682771552996256790845
235702552186222358130016700834523443
236821935793184701956510729781804354
173890560727428048583995919729021726
612291298420516067579036232337699453
964191475175567557695392233803056825
308599977441675784352815913461340394
604901269542028838347101363733824484
506660093348484440711931292537694657
354337375724772230181534032647177531
984537341478674327048457983786618703
257405938924215709695994630557521063
203263493209220738320923356309923267
504401701760572026010829288042335606
643089888710297380797578013056049576
342838683057190662205291174822510536
697756603029574043387983471518552602
805333866357139101046336419769097397

```

432 285 994 219 837 046 979 109 956 303 389 604 \
675 889 865 795 711 176 566 670 039 156 748 153 \
115 943 980 043 625 399 399 731 203 066 490 601 \
325 311 304 719 028 898 491 856 203 766 669 164 \
468 791 125 249 193 754 425 845 895 000 311 561 \
682 974 304 641 142 538 074 897 281 723 375 955 \
380 661 719 801 404 677 935 614 793 635 266 265 \
683 339 509 760 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! )

```

```

! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! } ,

```

```

{1, 2,

```

```

((((((((((((((((((((((((((((((((((((((((2 601 218 943 565 795 100 204 903 227 081 043 611 191 521 875 \
016 945 785 727 541 837 850 835 631 156 947 382 \
240 678 577 958 130 457 082 619 920 575 892 247 \
259 536 641 565 162 052 015 873 791 984 587 740 \
832 529 105 244 690 388 811 884 123 764 341 191 \
951 045 505 346 658 616 243 271 940 197 113 909 \
845 536 727 278 537 099 345 629 855 586 719 369 \
774 070 003 700 430 783 758 997 420 676 784 016 \
967 207 846 280 629 229 032 107 161 669 867 260 \
548 988 445 514 257 193 985 499 448 939 594 496 \
064 045 132 362 140 265 986 193 073 249 369 770 \
477 606 067 680 670 176 491 669 403 034 819 961 \
881 455 625 195 592 566 918 830 825 514 942 947 \
596 537 274 845 624 628 824 234 526 597 789 737 \
740 896 466 553 992 435 928 786 212 515 967 483 \
220 976 029 505 696 699 927 284 670 563 747 137 \
533 019 248 313 587 076 125 412 683 415 860 129 \
447 566 011 455 420 749 589 952 563 543 068 288 \
634 631 084 965 650 682 771 552 996 256 790 845 \
235 702 552 186 222 358 130 016 700 834 523 443 \
236 821 935 793 184 701 956 510 729 781 804 354 \
173 890 560 727 428 048 583 995 919 729 021 726 \
612 291 298 420 516 067 579 036 232 337 699 453 \
964 191 475 175 567 557 695 392 233 803 056 825 \
308 599 977 441 675 784 352 815 913 461 340 394 \
604 901 269 542 028 838 347 101 363 733 824 484 \
506 660 093 348 484 440 711 931 292 537 694 657 \
354 337 375 724 772 230 181 534 032 647 177 531 \
984 537 341 478 674 327 048 457 983 786 618 703 \

```

```

257 405 938 924 215 709 695 994 630 557 521 063 \
203 263 493 209 220 738 320 923 356 309 923 267 \
504 401 701 760 572 026 010 829 288 042 335 606 \
643 089 888 710 297 380 797 578 013 056 049 576 \
342 838 683 057 190 662 205 291 174 822 510 536 \
697 756 603 029 574 043 387 983 471 518 552 602 \
805 333 866 357 139 101 046 336 419 769 097 397 \
432 285 994 219 837 046 979 109 956 303 389 604 \
675 889 865 795 711 176 566 670 039 156 748 153 \
115 943 980 043 625 399 399 731 203 066 490 601 \
325 311 304 719 028 898 491 856 203 766 669 164 \
468 791 125 249 193 754 425 845 895 000 311 561 \
682 974 304 641 142 538 074 897 281 723 375 955 \
380 661 719 801 404 677 935 614 793 635 266 265 \
683 339 509 760 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 000 000 000 000 000 000 000 000 000 \
000 000 000 ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! )
! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! ) ! }

```

4. Tests and Conditionals (EIWL3 Section 28)

(a)

Use **PrimeQ** and **/@** to generate a **True** or **False** list that is twenty elements long expressing which numbers in **Range[20]** are prime.

```

In[168]:= PrimeQ /@ Range[20]
Out[168]= {False, True, True, False, True, False, True, False, False, False,
          True, False, True, False, False, False, True, False, True, False}

```

(b)

Combine **PrimeQ** with **Select** to only list the numbers in **Range[20]** that are prime.

```

In[169]:= Select[Range[20], PrimeQ]
Out[169]= {2, 3, 5, 7, 11, 13, 17, 19}

```

5. More About Pure Functions (EIWL3 Section 29)

(a)

Accomplish exactly the same thing as `Table[n*(n-1)/2, {n, 6}]` using `Array` and a pure function.

```
In[170]:=
Array[#*(#-1)/2 &, 6]
Out[170]=
{0, 1, 3, 6, 10, 15}
```

(b)

Make some modifications to `FoldList[Plus, {1, 2, 3, 4, 5}]` so that it produces a list of the first 10 factorials. Instead of hand-coding the list up to 10, begin by first changing `{1, 2, 3, 4, 5}` to `Range[10]`.

```
In[171]:=
FoldList[Times, Range[10]]
Out[171]=
{1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800}
```

6. Rearranging Lists (EIWL3 Section 30)

(a)

Use `Transpose` and one of the *levelspec* options to turn `{{{1, uno}, {2, dos}, {3, tres}}, {{4, cuatro}, {5, cinco}, {6, seis}}}` into `{{{1, 2, 3}, {uno, dos, tres}}, {{4, 5, 6}, {cuatro, cinco, seis}}}`

```
In[172]:=
Transpose[{{{1, uno}, {2, dos}, {3, tres}}, {{4, cuatro}, {5, cinco}, {6, seis}}}, 2 ↔ 3]
Out[172]=
{{{1, 2, 3}, {uno, dos, tres}}, {{4, 5, 6}, {cuatro, cinco, seis}}}
```

(b)

Use `Flatten` and a *levelspec* option to turn `{{{1, uno}, {2, dos}, {3, tres}}, {{4, cuatro}, {5, cinco}, {6, seis}}}` into `{{1, uno}, {2, dos}, {3, tres}, {4, cuatro}, {5, cinco}, {6, seis}}`

```
In[173]:=
Flatten[
  Transpose[{{{1, uno}, {2, dos}, {3, tres}}, {{4, cuatro}, {5, cinco}, {6, seis}}}, 2 ↔ 3], 1]
Out[173]=
{1, 2, 3}, {uno, dos, tres}, {4, 5, 6}, {cuatro, cinco, seis}
```


7. Parts of Lists (EIWL3 Section 31)

(a)

Use the magical **All** position (you will need to use **All** more than once) to turn

`{{{Eli, Lerner},{Harper,Yonago},{Hexi,Jin}},{{Jeremy,Choy},{Rania,Zaki},{Tahm,Loyd},{Walker,Harris}}}` into
`{{Eli,Harper,Hexi},{Jeremy,Rania,Tahm,Walker}}`

```
In[174]:=
list1=Take[#,All,1]&/@{{{Eli, Lerner},{Harper,Yonago},{Hexi,Jin}},
      {{Jeremy,Choy},{Rania,Zaki},{Tahm,Loyd},{Walker,Harris}}};
list2=Flatten[#]&/@ list1;
list2
```

```
Out[176]=
{{Eli, Harper, Hexi}, {Jeremy, Rania, Tahm, Walker}}
```

(b)

Use a magical *negative positional argument* to extract `{Jeremy,Rania,Tahm,Walker}` from
`{{Eli,Harper,Hexi},{Jeremy,Rania,Tahm,Walker}}` and combine that with **Take** with a
different magical *negative* argument to extract `{Tahm,Walker}`.

```
In[177]:=
Take[list2[[2]], -2]
```

```
Out[177]=
{Tahm, Walker}
```

8. Patterns (EIWL3 Section 32)

(a)

Use **Cases** to choose the lists that begin and end with the same letter in this list of lists (but look ahead to part (b) before you solve part (a)):

```
{
  {"a", "l", "u", "l", "a"},
  {"a", "l", "o", "h", "a"},
  {"a", "r", "a", "r", "a"},
  {"b", "o", "n", "u", "s"},
  {"c", "i", "v", "i", "c"},
  {"d", "e", "b", "e", "d"},
  {"e", "l", "b", "o", "w"},
  {"z", "a"},
}
```

```
{“z”, “z”}
}
```

In[178]:=

```
Cases[{
  {"a", "l", "u", "l", "a"},
  {"a", "l", "o", "h", "a"},
  {"a", "r", "a", "r", "a"},
  {"b", "o", "n", "u", "s"},
  {"c", "i", "v", "i", "c"},
  {"d", "e", "b", "e", "d"},
  {"e", "l", "b", "o", "w"},
  {"z", "a"},
  {"z", "z"}
}, {x_, __, x_}]
```

Out[178]=

```
{{a, l, u, l, a}, {a, l, o, h, a}, {a, r, a, r, a}, {c, i, v, i, c}, {d, e, b, e, d}}
```

(b)

The pattern **BlankNullSequence** has the shorthand `___`. Use `___` to improve the pattern you used in Part (a) so that the two-letter list `{z, z}` is also included in your result.

In[179]:=

```
Cases[{
  {"a", "l", "u", "l", "a"},
  {"a", "l", "o", "h", "a"},
  {"a", "r", "a", "r", "a"},
  {"b", "o", "n", "u", "s"},
  {"c", "i", "v", "i", "c"},
  {"d", "e", "b", "e", "d"},
  {"e", "l", "b", "o", "w"},
  {"z", "a"},
  {"z", "z"}
}, {x_, ___, x_}]
```

Out[179]=

```
{{a, l, u, l, a}, {a, l, o, h, a}, {a, r, a, r, a}, {c, i, v, i, c}, {d, e, b, e, d}, {z, z}}
```

9. Assigning Names to Things (EIWL3 Section 38)

(a)

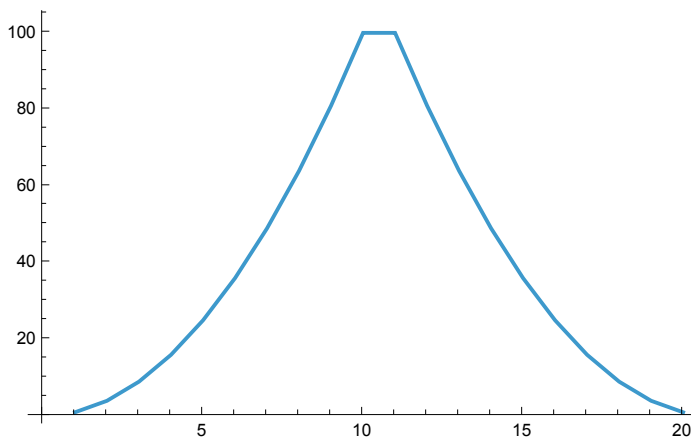
Use **Module** to compute `x=Factorial[10]`, and then produce `{x,x^2,x^3}`.

```
In[180]:=
Module[{x = Factorial[10]}, {x, x^2, x^3}]
Out[180]=
{3 628 800, 13 168 189 440 000, 47 784 725 839 872 000 000}
```

(b)

Inside **Module**, let `rangeSquared=Range[10]^2`, and then produce a list line plot of `rangeSquared` joined with `Reverse[rangeSquared]`.

```
In[181]:=
Module[{rangeSquared=Range[10]^2},
  ListLinePlot[Join[rangeSquared, Reverse[rangeSquared]]]]
Out[181]=
```



10. Immediate and Delayed Values (EIWL3 Section 39)

(a)

Make a *one-character change* to this expression,

`Module[{x:=RandomInteger[10]}, {x, x^2, x^3, x^4}]`, so that it produces four different powers of the same random number instead of four different powers of different random numbers.

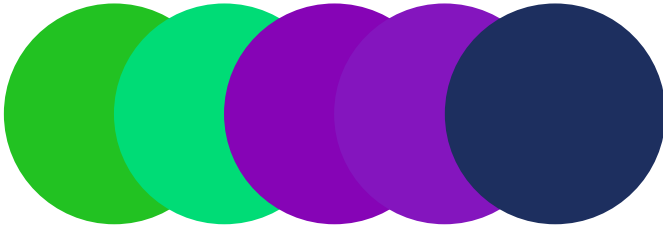
```
In[182]:=
Module[{x=RandomInteger[10]}, {x, x^2, x^3, x^4}]
Out[182]=
{1, 1, 1, 1}
```

(b)

Make a *one-character change* to this expression,

`Module[{color=RandomColor[]}, Graphics[Table[Style[Disk[{i,0}], color], {i, 5}]]]`, so that it produces five different-color disks.

```
In[183]:=
Module[{color:=RandomColor[]},Graphics[Table[Style[Disk[{i,0}],color],{i,5}]]]
Out[183]=
```



11. Defining Your Own Functions (EIWL3 Section 40)

(a)

Define a function **f** that takes a list of three elements and out of them makes a list of lists that contains all six possible orderings. Using **Permutations** will make this easy.

Include a test of your function as **f[1,2,3]** and make sure it gets
 $\{\{1,2,3\}, \{1,3,2\}, \{2,1,3\}, \{2,3,1\}, \{3,1,2\}, \{3,2,1\}\}$.

```
In[184]:=
f[{x_, y_, z_}] := Permutations[{x, y, z}]
f[1, 2, 3]
Out[185]=
{{1, 2, 3}, {1, 3, 2}, {2, 1, 3}, {2, 3, 1}, {3, 1, 2}, {3, 2, 1}}
```

(b)

Define a function **g** that gives 1 for **g[0]**, and gives **n*g[n-1]** for any integer **n** greater than 0, *but don't use an If statement!* Include a test of your function as **g[6]** and make sure it gets 720.

```
In[186]:=
g[n_] := n * g[n - 1] /; (n > 0, n = 0 -> 1);
g[10]
```

12. More About Patterns (EIWL3 Section 41)

(a)

Use the replacement rule notation — e.g., **/.** and **->** — to exchange the first and last element in any list containing two or more elements and test your replacement using the list
 $\{\text{alpha}, \text{beta}, \text{gamma}, \text{delta}, \text{epsilon}\}$.

```
In[186]:=
(* 12a *)
```

(b)

Starting with `Characters/@RomanNumeral[Range[100]]`, select all the sequences corresponding to the Roman numerals that have XXX in them.

```
In[187]:=
Select[Characters/@RomanNumeral[Range[100]], MatchQ[#, {___, X, X, X, ___}] &]

Out[187]=
{}

In[188]:=
Select[{{a, a}, {b, a}, {a, b, c}, {b, b}, {c, a}, {b, b, b}}, MatchQ[#, {b, _}] &]

Out[188]=
{{b, a}, {b, b}}
```

```
In[189]:=
Characters /@ RomanNumeral [Range[100]]

Out[189]=
{{I}, {I, I}, {I, I, I}, {I, V}, {V}, {V, I}, {V, I, I}, {V, I, I, I}, {I, X}, {X}, {X, I},
 {X, I, I}, {X, I, I, I}, {X, I, V}, {X, V}, {X, V, I}, {X, V, I, I}, {X, V, I, I, I},
 {X, I, X}, {X, X}, {X, X, I}, {X, X, I, I}, {X, X, I, I, I}, {X, X, I, V}, {X, X, V},
 {X, X, V, I}, {X, X, V, I, I}, {X, X, V, I, I, I}, {X, X, I, X}, {X, X, X},
 {X, X, X, I}, {X, X, X, I, I}, {X, X, X, I, I, I}, {X, X, X, I, V}, {X, X, X, V},
 {X, X, X, V, I}, {X, X, X, V, I, I}, {X, X, X, V, I, I, I}, {X, X, X, I, X}, {X, L},
 {X, L, I}, {X, L, I, I}, {X, L, I, I, I}, {X, L, I, V}, {X, L, V}, {X, L, V, I},
 {X, L, V, I, I}, {X, L, V, I, I, I}, {X, L, I, X}, {L}, {L, I}, {L, I, I}, {L, I, I, I},
 {L, I, V}, {L, V}, {L, V, I}, {L, V, I, I}, {L, V, I, I, I}, {L, I, X}, {L, X},
 {L, X, I}, {L, X, I, I}, {L, X, I, I, I}, {L, X, I, V}, {L, X, V}, {L, X, V, I},
 {L, X, V, I, I}, {L, X, V, I, I, I}, {L, X, I, X}, {L, X, X}, {L, X, X, I},
 {L, X, X, I, I}, {L, X, X, I, I, I}, {L, X, X, I, V}, {L, X, X, V}, {L, X, X, V, I},
 {L, X, X, V, I, I}, {L, X, X, V, I, I, I}, {L, X, X, I, X}, {L, X, X, X},
 {L, X, X, X, I}, {L, X, X, X, I, I}, {L, X, X, X, I, I, I}, {L, X, X, X, I, V},
 {L, X, X, X, V}, {L, X, X, X, V, I}, {L, X, X, X, V, I, I}, {L, X, X, X, V, I, I, I},
 {L, X, X, X, I, X}, {X, C}, {X, C, I}, {X, C, I, I}, {X, C, I, I, I}, {X, C, I, V},
 {X, C, V}, {X, C, V, I}, {X, C, V, I, I}, {X, C, V, I, I, I}, {X, C, I, X}, {C}}
```

(c)

Use `StringJoin` to turn what you got in 12(b) into

`{XXX, XXXI, XXXII, XXXIII, XXXIV, XXXV, XXXVI, XXXVII, XXXVIII, XXXIX, LXXX, LXXXI, LXXXII, LXXXIII, LXXXIV, LXXXV, LXXXVI, LXXXVII, LXXXVIII, LXXXIX}`.

```
In[190]:=
StringJoin[
  Select[Characters/@RomanNumeral[Range[100]], MatchQ[#, {___, X, X, X, ___}] &]]

Out[190]=
```