

Status	Finished
Started	Monday, 18 November 2024, 10:40 AM
Completed	Friday, 22 November 2024, 8:31 AM
Duration	3 days 21 hours
Marks	100.00/100.00
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 10.00 out of 10.00

[TwoDimensionArrayQueries]

Cho ma trận kích thước $m \times n$ chứa các số nguyên, các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n . Có Q câu hỏi, mỗi câu hỏi đưa ra 2 số i, j . Nhiệm vụ của bạn là in ra giá trị của phần tử ở hàng thứ i và cột thứ j của ma trận.

Đầu vào

Đầu vào từ bàn phím gồm $m + Q + 1$ dòng.

- Dòng đầu tiên chứa 3 số nguyên n, m, Q ($m, n, Q \leq 500$).
- m dòng tiếp theo mỗi dòng chứa n số nguyên biểu diễn ma trận đã cho, các số liên tiếp trên một dòng cách nhau bởi một dấu cách.
- Q dòng tiếp theo, mỗi dòng là một câu hỏi chứa 2 số nguyên i, j cách nhau bởi một dấu cách.

Đầu ra

In ra màn hình Q dòng, mỗi dòng tương ứng với câu trả lời của một câu hỏi.

For example:

Input	Result
3 3 2	5
1 3 5	4
2 4 7	
3 4 2	
1 3	
2 2	

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7
8  int main()
9  {
10     ios_base::sync_with_stdio(0);
11     cin.tie(0);
12     int m,n,q; cin>>m>>n>>q;
13     int a[m][n];
14     for(int i=0;i<m;i++)
15     {
16         for(int j=0;j<n;j++)
17         {
18             cin>>a[i][j];
19         }
20     }
21     while(q-->0)
22     {
23         int x,y; cin>>x>>y;
24         cout<<a[x-1][y-1]<<el;
25     }
26     return 0;
27 }
28
29

```

	Input	Expected	Got	
✓	3 3 2 1 3 5 2 4 7 3 4 2 1 3 2 2	5 4	5 4	✓
✓	10 20 15 610 132 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 515 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 185 381 120 695 333 985 387 360 615 369 154 966 934 341 352 948 690 516 26 953 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142 4 6 10 13 9 18 3 19 2 3 10 13 3 4 5 11 1 10 7 7 10 3 2 19 3 8 5 13 10 17	299 276 882 571 515 276 96 312 26 215 47 489 938 250 895	299 276 882 571 515 276 96 312 26 215 47 489 938 250 895	✓

	Input	Expected	Got	
✓	100 132 231 936 421 503 78 36 246 708 774 11 131 685 511 461 310 775 748 740 639 609 745 355 982 282 263 184 18 949 198 95 263 837 32 684 340 462 72 938 170 846 301 301 883 813 762 194 940 511 286 579 472 383 934 454 666 549 991 684 498 189 779 761 26 811 797 718 626 869 8 796 68 661 450 303 826 212 849 766 75 136 697 547 871 983 2 537 532 345 221 30 534 353 791 912 516 940 982 142 162 342 291 230 3 741 885 182 953 735 948 381 871 998 928 742 981 282 632 514 979 205 544 513 558 688 777 427 628 759 569 790 101 860 372 457 601 258 639 907 345 939 288 216 937 568 310 919 203 294 433 182 500 329 48 58 369 825 485 350 585 407 492 38 267 217 495 221 827 486 480 172 778 768 388 715 688 50 986 243 697 419 778 197 101 826 607 822 651 445 172 588 852 665 979 471 234 826 692 61 313 524 233 91 292 973 158 333 23 145 576 720 916 354 269 17 532 877 192 184 322 364 124 174 381 103 997 615 930 42 676 595 566 909 38 211 882 196 544 258 341 472 330 258 179 600 627 711 477 819 247 151 536 372 677 269 827 674 885 757 68 913 704 635 823 742 846 57 939 742 667 632 214 998 890 393 950 870 105 427 41 352 930 577 76 607 847 904 633 84 13 702 997 718 337 172 812 535 230 751 277 249 384 843 247 626 589 197 496 46 976 890 750 906 467 827 865 666 83 499 750 96 553 100 166 242 272 979 777 854 82 406 104 466 601 703 445 190 901 293 236 229 183 339 488 3 166 353 669 601 204 772 49 109 872 216 351 144 547 480 351 629 886 807 448 488 510 245 30 763 538 619 993 722 958 481 725 476 186 746 77 743 518 126 852 390 694 204 887 593 684 590 223 923 397 23 763 907 268 145 671 158 764 664 232 722 497 957 550 683 704 627 426 574 106 631 965 800 835 204 394 871 794 969 146 191 344 261 450 964 407 473 122 171 137 355 246 986 664 796 670 720 424 448 295 530 79 612 682 266 168 76 490 962 397 636 505 741 898 307 57 305 781 180 828 270 887 74 257 903 871 279 624 647 79 271 529 159 235 211 777 403 640 619 365 37 608 222 131 506 529 188 163 310 720 343 581 607 418 190 511 641 469 487 288 548 758 817 59 993 380 837 748 20 808 465 410 416 687 541 274 568 81 789 231 802 133 164 761 903 354 624 544 823 111 184 723 221 353 783 566 733 972 314 106 780 131 516 197 818 409 471 739 490 261 970 644 746 486 406 649 840 382 545 15 846 81 738 67 434 873 986 519 197 652 625 978 784 493 527 954 254 350 45 97 963 367 741 709 853 499 710 693 882 255 60 728 336 151 147 122 24 485 994 222 490 971 552 274 465 431 580 71 781 626 168 97 993 910 158 847 409 869 892 643 476 305 723 165 456 871 639 480 356 633 54 846 605 958 472 422 389 405 493 523 383 662 620 376 924 778 575 685 999 820 329 828 125 52 345 581 275 984 413 632 970 820 830 575 778 655 997 168 60 842 691 443 504 663 171 780 793 99 466 145 271 795 973 396 199 670 329 475 6 94 459 976 914 289 551 693 296 900 213 708 95 256 503 599 271 27 732 64 126 198 209 397 993 534 145 192 556 826 667 563 920 478 539 187 768 443 232 416 343 797 125 438 53 980 390 324 7 122 740 133 672 302 882 665 188 379 209 745 557 229 308 830 707 847 17 827 290 601 596 986 398 721 776 451 701 518 127 61 992 219 194 664 521 429 681 710 160 891 455 70 472 763 900 531 962 269 359 605 870 955 591 268 28 367 71 81 238 550 142 230 769 689 895 643 470 928 353 982 819 160 52 643 275 952 175 589 573 886 194 443 193 785 63 221 505 486 654 743 36 797 325 158 838 572 153 308 501 506 642 672 18 695 316 293 999 491 234 573 729 429 368 922 214 432 495 71 918 149 166 307 298 844 817 488 416 970 148 917 828 791 590 198 838 906 843 837 749 77 410 830 506 131 752 73 563 247 144 833 748 663 492 399 507 661 887 923 631 388 193 811 179 783 9 17 41 852 854 142 930 617 972 436 100 76 861 663 675 358 848 423 21 693 174 880 354 414 155 338 802 348 501 981 483 511 350 876 715 556 18 645 525 342 434 625 418 295 640 93 653 489 869 26 182 395 258 888 809 414 226 611 762 728 944 246 591 294 474 306 851 493 304 376 187 738 2 606 385 642 51 391 483 272 769 17 668 28 906 477 442 484 441 556 564 385 154 507 680 629 166 883 474 470 259 661 560 613 619 297 608 671 40 91 295 810 461 963 190 719 793 9 ...snip... 40 618 765 406 208 401 4 475 744 237 601 805 603 500 911 760 925 735 76 617 653 973 790 513 445 434 459 130 561 83 649 853 53 766 260 261 167 264 88 263 854 42 69 809 542 332 921 467 67 997 436 72	547 447 122 835 95 479 44 509 449 755 934 621 939 315 959 635 404 210 289 66 211 580 580 498 314 969 76 372 220 662 843 9 157 232 609 126 183 248 582 236 780 494 550 16 64 969 126 945 592 772	547 447 122 835 95 479 44 509 449 755 934 621 939 315 959 635 404 210 289 66 211 580 580 498 314 969 76 372 220 662 843 9 157 232 609 126 183 248 582 236 780 494 550 16 64 969 126 945 592 772	✓

	Input	Expected	Got
	971 226 585 768 12 396 898 926 831 548 779 884 666 39 498 834 656 586 449 510 628 870 671 522	47	47
	202 592 341 269 942 777 341 913 4 278 681 368 27 932 294 858 480 74 743 498 113 241 684 121 827	73	73
	134 983 808 4 654 682 207 247 24 828 189 801 522 454 157 800 487 526 827 419 820 686 251 246	881	881
	781 750 360 22 434 481 201 920 465 361	448	448
	925 471 44 484 70 68 312 611 221 834 65 379 987 905 905 814 676 77 852 280 324 633 382 36 7 816	590	590
	869 209 89 686 922 366 158 966 850 228 386 514 192 608 349 257 339 688 514 596 502 191 673 707	806	806
	471 349 340 853 385 700 21 255 261 462 293 183 828 451 502 678 32 240 193 224 200 894 833 539	951	951
	582 700 135 436 891 161 495 714 510 188 919 248 888 940 855 149 755 148 684 583 952 186 262 336	188	188
	427 807 912 627 53 97 519 635 797 6 423 688 167 919 754 30 107 673 278 347 966 485 496 721 633	290	290
	180 304 937 719 918 273 146 725 537	307	307
	125 778 635 996 765 432 3 541 473 522 812 227 552 271 253 182 618 219 19 466 292 5 998 948 942	521	521
	717 867 568 215 944 105 693 75 740 689 192 173 692 733 646 567 897 873 471 168 478 6 786 697 25	665	665
	604 341 30 603 290 973 320 157 541 888 101 646 581 528 387 270 721 912 315 454 558 882 352 783	264	264
	705 872 262 711 11 311 737 615 653 767 570 943 92 243 452 985 131 905 984 712 434 371 334 155	923	923
	635 1 961 193 235 665 328 941 538 590 652 549 254 741 516 907 861 87 202 305 330 654 291 813	609	609
	559 275 525 993 998 211 500 633 565 462	238	238
	178 800 479 506 93 369 449 98 270 703 839 787 962 52 226 516 358 556 170 649 369 81 276 246 427	104	104
	626 809 927 259 374 741 437 527 221 295 972 590 744 70 213 799 262 0 113 314 226 629 672 134	982	982
	151 673 855 585 301 453 12 927 262 291 186 989 33 975 868 606 623 840 548 719 911 761 519 173	579	579
	113 984 839 339 966 864 825 117 537 680 702 191 133 66 118 748 358 657 737 743 984 957 349 607	723	723
	797 249 327 60 363 198 585 476 182 425 168 148 289 993 618 178 26 672 369 511 739 840 259 449	422	422
	497 348 192 481 305 893 89 103 494 416 515	420	420
	857 966 101 334 148 878 854 649 167 847 267 697 225 291 419 737 382 259 996 831 108 697 375 589	57	57
	2 620 30 457 115 798 973 972 764 426 658 913 304 512 914 823 360 533 520 937 824 939 674 559	868	868
	550 23 742 658 72 118 600 74 90 630 884 205 781 857 178 545 283 836 810 939 349 724 762 61 609	523	523
	282 998 434 574 25 345 124 400 87 135 472 557 735 898 648 717 782 853 498 991 383 44 274 220	771	771
	206 213 921 931 975 334 540 610 332 326 184 709 671 660 461 111 795 933 668 882 832 668 600 614	711	711
	522 98 606 905 142 232 477 349 798 750 632	827	827
	85 14	951	951
	88 41	380	380
	4 34	608	608
	31 16	633	633
	85 79	183	183
	38 93	983	983
	26 92	551	551
	15 7	894	894
	27 110	829	829
	77 63	333	333
	65 121	353	353
	59 27	916	916
	19 52	256	256
	61 112	801	801
	39 41	986	986
	95 77	291	291
	69 87	393	393
	44 22	821	821
	6 36	948	948
	99 87	165	165
	67 38	618	618
	25 132	572	572
	46 64	624	624

Input	Expected	Got
54 39	382	382
5 13	179	179
73 10	607	607
64 74	303	303
81 51	745	745
71 6	679	679
51 105	287	287
85 45	46	46
83 97	338	338
78 29	227	227
77 115	733	733
1 19	451	451
28 111	567	567
62 47	781	781
62 49	40	40
43 44	696	696
32 15	115	115
44 52	212	212
69 120	826	826
39 123	24	24
41 113	280	280
41 69	78	78
69 39	771	771
39 129	591	591
29 107	688	688
77 116	875	875
3 85	705	705
31 89	687	687
47 54	500	500
34 101	269	269
4 45	294	294
75 53	891	891
24 67	58	58
42 15	76	76
29 92	903	903
44 24	746	746
40 129	952	952
63 20	433	433
60 68	334	334
59 101	148	148
69 36	325	325
93 63	861	861
68 83	739	739
52 115	184	184
67 18	344	344
74 102	149	149
62 94	872	872
56 117	554	554
19 57	932	932
32 78	97	97
29 131	66	66

Input	Expected	Got
20 83	325	325
10 121	260	260
28 30	970	970
83 113	58	58
56 30	295	295
77 79	602	602
73 68	396	396
75 108	209	209
9 28	257	257
13 129	957	957
15 119	531	531
35 14	903	903
62 114	495	495
19 33	327	327
3 81	617	617
19 88	831	831
20 26	130	130
56 128	791	791
48 63	433	433
37 118	18	18
40 124	452	452
55 49	560	560
60 24	552	552
91 105	741	741
22 80	286	286
54 44	354	354
29 124	233	233
5 32	736	736
44 59	910	910
39 28	604	604
75 27	686	686
72 102	852	852
89 58	793	793
92 103	498	498
16 5	444	444
92 12	430	430
44 15	314	314
22 50	840	840
75 15	125	125
23 85	789	789
68 124	220	220
83 88	172	172
68 48	925	925
42 83	422	422
14 115	661	661
13 109	768	768
90 24	325	325
33 121	92	92
63 38	459	459
14 1	160	160
8 43	635	635

	Input	Expected	Got
	89 101	78	78
	51 68	919	919
	58 91	925	925
	75 105	629	629
	24 40	114	114
	60 34	662	662
	89 74	354	354
	44 111	18	18
	41 119	77	77
	63 66	354	354
	49 98	586	586
	42 36	455	455
	29 102	711	711
	62 108	688	688
	86 47	601	601
	99 109	263	263
	72 7	572	572
	37 97	970	970
	29 122	427	427
	13 3	429	429
	4 28	614	614
	9 20	21	21
	50 48	131	131
	74 57	741	741
	14 23	99	99
	48 124	394	394
	39 10	932	932
	47 5	992	992
	96 32		
	62 76		
	29 130		
	8 24		
	65 92		
	33 96		
	6 53		
	97 42		
	17 129		
	52 45		
	9 123		
	61 74		
	13 86		
	36 64		
	100 20		
	71 70		
	90 47		
	28 60		
	47 62		
	21 107		
	82 67		
	56 102		
	99 31		

	Input	Expected	Got
	95 64		
	8 3		
	85 3		
	24 73		
	90 69		
	56 68		
	91 49		
	59 131		
	54 69		
	96 64		
	69 17		
	20 11		
	5 13		
	94 125		
	88 124		
	63 56		
	11 114		
	32 27		
	91 115		
	44 55		
	43 120		
	57 115		
	58 9		
	98 87		
	66 23		
	43 5		
	87 74		
	74 94		
	67 88		
	55 64		
	11 106		
	39 66		
	92 109		
	68 3		
	77 122		
	13 7		
	61 43		
	56 9		
	87 30		
	80 129		
	97 44		
	89 23		
	40 80		
	31 3		
	34 66		
	11 45		
	6 116		
	69 58		
	72 1		
	30 112		
	98 112		

	Input	Expected	Got	
	71 27			
	15 102			
	80 74			
	30 62			

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 2

Correct

Mark 10.00 out of 10.00

[InsertColumn]

Cho ma trận A kích thước $m \times n$ chứa các số nguyên, các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n .

Viết chương trình nhập vào 2 số nguyên i và x . Sau đó, chèn thêm vào ma trận A một cột ở vị trí thứ i với tất cả các phần tử có cột đó bằng x .

Đầu vào

Đầu vào từ bàn phím gồm $m + 2$ dòng:

- Dòng đầu tiên chứa 2 số nguyên n, m ($m, n \leq 1000$).
- m dòng tiếp theo mỗi dòng chứa n số nguyên biểu diễn ma trận đã cho, các số liên tiếp nhau trên một dòng cách nhau bởi một dấu cách.
- Dòng cuối chứa 2 số nguyên i, x .

Đầu ra

In ra màn hình m dòng biểu diễn ma trận sau khi đã thêm cột ở vị trí thứ i , các số liên tiếp nhau trên một dòng cách nhau bởi một dấu cách.

For example:

Input	Result
3 3	1 3 2 3
1 2 3	4 3 5 6
4 5 6	2 3 3 4
2 3 4	
2 3	

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7
8  int main()
9  {
10     ios_base::sync_with_stdio(0);
11     cin.tie(0);
12     int m,n,id,x; cin>>m>>n;
13     int a[m][n];
14     for(int i=0;i<m;i++)
15     {
16         for(int j=0;j<n;j++)
17         {
18             cin>>a[i][j];
19         }
20     }
21     cin>>id>>x;
22     for(int i=0;i<m;i++)
23     {
24         for(int j=0;j<n;j++)
25         {
26             if(j==id-1) cout<<x<<" ";
27             cout<<a[i][j]<<" ";
28         }

```

```
29         cout<<endl;
30     }
31     return 0;
32 }
33
34
```

	Input	Expected	Got	
✓	3 3 1 2 3 4 5 6 2 3 4 2 3	1 3 2 3 4 3 5 6 2 3 3 4	1 3 2 3 4 3 5 6 2 3 3 4	✓
✓	4 5 1 3	3 1 1 1 1 1 3 1 1 1 1 1 3 1 1 1 1 1 3 1 1 1 1 1	3 1 1 1 1 1 3 1 1 1 1 1 3 1 1 1 1 1 3 1 1 1 1 1	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.



Question 3

Correct

Mark 10.00 out of 10.00

[InsertRow]

Cho ma trận A kích thước $m \times n$ chứa các số nguyên, các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n .

Viết chương trình nhập vào 2 số nguyên i và x . Sau đó, chèn thêm vào ma trận A một hàng ở vị trí thứ i với tất cả các phần tử có hàng đó bằng x .

Đầu vào

Đầu vào từ bàn phím gồm $m + 2$ dòng:

- Dòng đầu tiên chứa 2 số nguyên m, n ($m, n \leq 1000$).
- m dòng tiếp theo mỗi dòng chứa n số nguyên biểu diễn ma trận đã cho, các số liên tiếp nhau trên một dòng cách nhau bởi một dấu cách.
- Dòng tiếp theo chứa 2 số nguyên i, x .

Đầu ra

In ra màn hình $m + 1$ dòng biểu diễn ma trận sau khi đã thêm hàng ở vị trí thứ i , các số liên tiếp nhau trên một dòng cách nhau bởi một dấu cách.

For example:

Input	Result
3 3	1 2 3
1 2 3	3 3 3
4 5 6	4 5 6
2 3 4	2 3 4
2 3	

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7
8  int main()
9  {
10     ios_base::sync_with_stdio(0);
11     cin.tie(0);
12     int m,n,id,x; cin>>m>>n;
13     int a[m][n];
14     for(int i=0;i<m;i++)
15     {
16         for(int j=0;j<n;j++)
17         {
18             cin>>a[i][j];
19         }
20     }
21     cin>>id>>x;
22     for(int i=0;i<m;i++)
23     {
24         if(i==id-1)
25         {
26             for(int j =0;j<n;j++)
27             {
28                 cout<<x<<" ";
29             }
30             cout<<el;
31         }
32         else
33             for(int j=0;j<n;j++)
34                 cout<<a[i][j]<<" ";
35         cout<<el;
36     }
37 }
```

```
29         }
30         cout<<el;
31     }
32     for(int j=0;j<n;j++)
33     {
34         cout<<a[i][j]<<" ";
35     }
36     cout<<el;
37 }
38 return 0;
39 }
40
41
```

	Input	Expected	Got	
✓	3 3 1 2 3 4 5 6 2 3 4 2 3	1 2 3 3 3 3 4 5 6 2 3 4	1 2 3 3 3 3 4 5 6 2 3 4	✓
✓	4 5 1 3	3 3 3 3 3 1	3 3 3 3 3 1	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.



Question 4

Correct

Mark 10.00 out of 10.00

[SumPerRow]

Cho một ma trận kích thước $m \times n$ chứa các số nguyên. Viết chương trình tính tổng của các phần tử trên cùng một hàng của ma trận đã cho.

Đầu vào

Đầu vào từ bàn phím gồm $m + 1$ dòng:

- Dòng đầu tiên chứa số nguyên m, n ($m, n \leq 1000$).
- m dòng tiếp theo, mỗi dòng chứa n số, cách nhau bởi một dấu cách, biểu diễn ma trận đã cho.

Đầu ra

In ra màn hình m số nguyên trên một dòng, số nguyên thứ i tương ứng với tổng các phần tử trên hàng thứ i của ma trận.

For example:

Input	Result
2 3 1 2 3 3 2 1	6 6

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7
8  int main()
9  {
10     ios_base::sync_with_stdio(0);
11     cin.tie(0);
12     int m,n; cin>>m>>n;
13     int a[m][n];
14     for(int i=0;i<m;i++)
15     {
16         for(int j=0;j<n;j++)
17         {
18             cin>>a[i][j];
19         }
20     }
21     for(int i=0;i<m;i++)
22     {
23         ll ans=0;
24         for(int j=0;j<n;j++)
25         {
26             ans+=a[i][j];
27         }
28         cout<<ans<<" ";
29     }
30     return 0;
31 }
32
33

```

	Input	Expected	Got	
✓	2 3 1 2 3 3 2 1	6 6	6 6	✓
✓	10 10 223 704 720 256 698 641 528 109 436 695 150 302 750 373 683 525 47 65 961 479 396 660 487 32 144 866 967 659 831 492 679 406 548 400 662 246 393 542 355 829 589 505 132 691 230 167 216 277 584 530 109 980 190 596 364 334 463 332 993 646 824 673 404 372 425 418 970 170 960 677 352 902 183 836 593 765 3 162 395 587 692 504 567 234 100 932 568 915 616 913 913 792 938 317 516 715 87 486 886 400	5010 4335 5534 5060 3921 5007 5893 4778 6041 6050	5010 4335 5534 5060 3921 5007 5893 4778 6041 6050	✓

	Input	Expected	Got	
✓	100 200 80 606 738 305 701 357 431 91 74 948 798 550 873 690 459 486 242 286 723 152 777 436 230 38 70 292 44 344 661 520 362 741 126 452 46 179 809 829 270 884 778 421 786 3 111 597 841 705 884 564 857 661 353 439 52 423 732 96 767 393 968 482 486 95 934 533 626 96 362 897 332 492 670 470 848 781 68 689 838 952 606 696 965 959 487 369 734 571 466 854 316 786 336 803 881 622 688 860 718 50 109 50 895 779 521 95 912 589 784 750 893 390 798 210 701 286 580 436 209 398 290 526 184 626 329 418 248 17 630 319 419 739 369 666 518 242 761 430 183 546 532 428 288 331 991 990 969 571 778 178 321 68 56 857 46 385 627 294 754 257 613 174 996 335 840 514 929 602 296 465 500 829 893 140 512 884 482 833 807 260 363 480 680 420 338 726 157 965 21 912 575 986 86 571 673 278 438 603 232 86 68 84 267 313 577 779 550 59 964 357 672 328 190 352 748 528 79 905 845 452 169 420 790 607 344 464 886 782 419 470 868 839 907 136 152 484 267 54 895 232 412 919 912 602 272 12 482 703 269 679 507 439 452 297 46 796 113 284 930 532 107 150 723 14 638 228 498 906 282 745 490 46 665 754 0 937 766 834 992 387 514 851 826 966 148 225 114 614 861 44 498 968 546 222 982 185 450 832 443 84 578 285 483 243 39 483 532 157 318 524 544 184 375 723 502 875 948 616 489 809 12 988 778 558 562 112 743 364 945 186 448 523 471 931 766 862 767 298 19 437	102609 102399 98275 103928 100224 102424 100709 89391 103860 97132 100306 103270 102579 99189 97815 99639 98766 99146 97927 105487 102292 104126 100726 99115 103578 100016 103906 102806 95763 99839 101827 93533 95866 102251 99796 96690 101940 96853 104025 103883 100272 100973 91097 103037 102182 99549 95840 95762 100895 101125 90067 108151 103269 99815 94645 100632 106092 106686 107995 100153 98076 98572 108123 98964 94525 92732 99191 101665 99706 95673 98330 99561 96492 107497 105511 96027 109114 100118 104593 99053 103102 98571 104117 89790 101459 98262 97626 101924 95944 98835 97789 96354 101287 105527 98360 96453 99097 105150 99793 99603	102609 102399 98275 103928 100224 102424 100709 89391 103860 97132 100306 103270 102579 99189 97815 99639 98766 99146 97927 105487 102292 104126 100726 99115 103578 100016 103906 102806 95763 99839 101827 93533 95866 102251 99796 96690 101940 96853 104025 103883 100272 100973 91097 103037 102182 99549 95840 95762 100895 101125 90067 108151 103269 99815 94645 100632 106092 106686 107995 100153 98076 98572 108123 98964 94525 92732 99191 101665 99706 95673 98330 99561 96492 107497 105511 96027 109114 100118 104593 99053 103102 98571 104117 89790 101459 98262 97626 101924 95944 98835 97789 96354 101287 105527 98360 96453 99097 105150 99793 99603	✓

	Input	Expected	Got	
	174 564 621 901 639 123 128 587 91 618 748 103 958 878 661 872 991 405 236 936 943 36 811 767 320 577 629 87 227 649 876 753 565 497 654 556 972 782 495 63 752 243 166 62 474 179 286 817 584 522 753 880 911 916 647 231 493 276 670 72 277 546 825 194 395 831 750 367 965 245 782 70 489 948 132 315 127 771 132 64 293 237 944 204 153 591 787 998 219 457 70 497 355 247 43 750 78 794 117 43 39 899 113 880 199 598 195 679 369 679 743 14 916 39 571 421 982 710 419 201 168 841 50 523 88 94 626 518 240 95 914 279 995 379 512 546 977 707 225 698 739 320 65 655 711 988 77 693 698 848 247 866 690 649 742 130 95 720 1 335 815 915 967 162 294 479 709 624 538 286 322 277 959 739 285 670 727 714 364 426 562 963 644 604 964 738 87 60 810 88 395 626 3 714 788 297 193 849 921 84 136 596 361 447 687 646 117 415 712 833 193 275 796 189 231 761 280 318 173 90 758 920 68 761 635 857 411 180 58 332 264 546 280 626 993 968 624 111 735 337 296 280 964 445 469 195 558 101 866 731 544 624 651 612 738 638 821 149 819 880 833 435 426 114 413 420 82 390 883 169 727 179 449 43 624 270 590 182 372 456 265 916 433 269 528 171 907 350 320 78 582 153 866 8 619 279 428 53 669 663 222 748 843 671 791 819 942 382 354 666 190 619 934 623 240 814 794 500 516 466 578 98 972 796 107 943 76 887 997 745 551 571 494 746 595 637 565 889 19 271 555 210 891 489 185 483 303 332 983 820 798			

	Input	Expected	Got	
	914 918 122 710 377 66 786 265 415 884 168 986 378 914 581 367 831 822 387 103 377 949 346 218 134 829 522 818 165 694 969 79 964 91 789 342 509 928 959 276 812 479 263 542 393 196 909 576 19 648 679 748 597 377 967 84 559 841 254 724 535 223 803 851 667 944 545 528 872 504 805 684 983 420 578 728 968 488 305 339 136 336 88 86 66 407 522 625 248 776 701 135 0 504 338 19 800 884 547 673 740 352 709 76 772 640 804 93 128 461 432 616 150 520 54 216 279 576 193 879 353 894 14 705 750 705 724 902 589 623 927 329 976 637 405 100 277 562 193 757 375 626 725 525 498 780 93 778 356 638 9 61 884 24 118 986 729 194 889 670 818 816 351 146 805 109 246 434 671 792 191 398 770 917 276 268 697 369 398 405 360 408 467 244 784 937 231 865 132 472 887 950 288 238 448 446 347 46 880 370 838 424 769 608 693 45 229 390 766 979 147 126 387 966 723 523 904 306 740 36 778 627 338 418 866 138 864 213 184 745 584 23 169 353 983 862 750 212 604 516 192 103 643 931 70 718 455 974 24 547 362 802 175 52 572 41 190 437 606 726 534 542 749 55 247 733 917 997 297 873 514 489 976 509 773 398 227 580 724 251 127 438 405 302 842 977 695 384 766 302 111 300 196 860 355 444 945 624 793 243 849 307 84 178 168 857 576 395 437 653 998 565 91 755 219 934 733 915 318 851 569 781 152 765 994 859 561 939 484 355 534 685 14 619 ...snip... 293 744 796 170 890 38 266 502 165 199 393 553 474 315			

	Input	Expected	Got	
	118 458 921 942 841 8 775			
	639 563 803 178 16			
	118 202 309 901 988 603 998			
	137 773 240 527 39 742 692			
	238 136 245 65 803 715 523			
	725 658 717 85 785 708 649			
	588 239 17 58 441 326 312			
	781 281 310 270 54 550 797			
	446 645 490 36 781 735 101			
	936 451 977 13 461 694 99			
	246 754 100 187 993 469 245			
	434 147 557 568 429 219 838			
	483 122 636 929 767 478 318			
	900 565 419 188 368 396 202			
	829 90 653 76 197 753 615			
	190 574 212 977 721 122 545			
	150 341 735 986 815 371 267			
	934 201 585 186 767 5 375			
	135 401 577 317 844 230 745			
	393 335 360 583 261 572 912			
	982 694 457 133 388 193 471			
	555 916 738 490 470 676 28			
	237 681 403 724 434 980 41			
	630 562 786 23 249 146 959			
	510 71 871 493 117 681 978			
	857 226 801 413 494 891 255			
	964 567 283 201 600 687 926			
	387 19 319 369 934 106 393			
	183 604 704 46 675 927 891			
	145 608 221 2 834 22 767			
	681 913 22 645			
	833 658 199 433 345 477 820			
	716 796 190 2 902 935 538			
	507 991 584 534 918 475 679			
	527 48 682 713 70 449 746			
	335 824 392 168 482 591 954			
	179 68 126 247 216 668 250			
	119 603 788 978 946 372 512			
	865 199 192 744 247 226 809			
	669 27 556 4 203 300 173			
	685 891 479 216 311 605 464			
	527 274 714 998 229 502 976			
	176 226 841 393 777 33 137			
	376 611 298 45 990 854 49			
	194 154 574 879 397 53 448			
	60 11 912 940 285 978 938			
	866 832 267 42 58 108 787			
	187 493 924 563 104 575 608			
	94 429 9 640 936 936 872			
	685 341 320 746 352 232 38			
	989 210 328 856 394 595 250			

	Input	Expected	Got	
	452 703 38 991 548 962 554 652 537 162 99 319 523 91 607 459 963 292 801 283 390 153 867 428 143 429 757 351 823 352 601 627 408 639 618 956 954 172 961 843 686 412 162 210 503 121 21 467 414 822 102 804 328 970 585 823 751 694 174 575 46 775 554 806 767 173 115 721 697 428 564 384 840 79 946 695 552 967 162 966 790 617 123 470 939 60 293 690 754 819 617 152 946 172 959 713 345 426 786 42 854 703 778 694 134 724 389 686 692 904 653 834 521 128 656 812 188 949 502 942 120 472 446 418 644 405 132 341 831 270 735 685 973 514 731 107 238 473 794 282 377 799 468 250 927 124 414 467 425 916 761 545 388 207 964 384 965 448 725 148 718 813 186 44 327 917 151 917 742 297 552 119 96 20 721 375 497 487 194 922 404 955 820 144 515 136 881 480 584 958 628 654 771 166 50 450 84 202 368 178 851 272 650 948 292 371 675 789 859 870 64 615 825 884 111 692 20 344 172 956 303 153 962 74 319 13 525 755 567 245 934 418 517 584 718 161 307 394 303 518 616 367 485 793 603 949 486 623 293 10 931 596 163 893 23 835 258 900 942 825 497 228 596 14 164 314 527 824 60 830 694 676 549 180 470 152 481 308 127 774 318 58 723 482 952 746 669 210 998 963 388 495 192 984 861 708 650 740 532 711 571 579 739 472 111 561 625 592 869 752 366 188 163 441 22 115 539 43 677 537 6 65 384 550 401 245 259 52 986 143 115 909 722 854 733 185 416 358 777 285 463 496 825 626 937 847 93 477 890 122 366 897			

	Input	Expected	Got	
	188 751 799 941 348 410 993 686 554 108 947 628 963 681 814 731 391 943 368 206 439 194 184 377 41 277 206 284 400 572 533 940 323 332 881 24 743 875 710 649 335 10 629 650 691 443 381 434 387 750 641 178 296 825 555 689 103 113 973 855 686 858 795 361 543 28 385 286 255 448 287 591 458 916 241 501 712 975 935 99 77 576 629 373 402 185 62 857 298 388 64 336 246 211 698 789 239 435 427 847 883 714 438 693 983 31 194 695 6 130 146 83 58 775 808 812 312 871 21 611 611 85 299 857 648 349 999 888 785 778 735 20 493 525 66 476 908 260 523 915 742 669 350 801 444 159 965 109 382 987 72 345 424 371 202 73 721 553 313 506 332 400 878 177 925 944 5 833 557 528 100 299 549 803 452 993 314 418 454 696 757 526 41 181 898 595 606 619 501 919 477 833 319 707 362 596 652 367 782 209 247 882 860 796 685 313 141 351 83 596 47 192 122 440 725 20 388 332 991 889 251 820 74 923 880 788 519 532 155 301 93 402 536 953 198 573 618 339 925 701 935 324 893 410 117 971 782 857 303 126 98 906 298 524 829 178 312 701 710 467 2 155 221 890 109 771 816 727 110 741 781 398 417 26 160 534 997 942 743 300 68 841 207 719 365 36 897 677 737 960 496 92 115 717 334 224 488 150 304 951 243 85 701 661 111 861 547 109			

	Input	Expected	Got	
✓	500 700	348205 343182 364617 338524 353213 346023	348205 343182 364617 338524 353213	✓
	371 376 877 289 749 525 818	367195 341369 359015 343997 355514 355805	346023 367195 341369 359015 343997	
	772 11 860 66 90 447 278	326242 345197 355918 348951 350043 351001	355514 355805 326242 345197 355918	
	586 724 608 642 973 606 178	352918 344319 351589 343019 357420 344118	348951 350043 351001 352918 344319	
	179 347 351 47 493 232 908	348784 352977 347694 349334 351497 349248	351589 343019 357420 344118 348784	
	437 426 205 808 154 435 97	353412 339516 345907 361597 350974 350991	352977 347694 349334 351497 349248	
	903 960 268 27 323 128 446	353247 359342 353762 343796 346032 359261	353412 339516 345907 361597 350974	
	414 927 724 352 651 685 994	348311 360898 351601 369393 350732 360065	350991 353247 359342 353762 343796	
	976 291 524 507 990 228 554	338645 345139 351729 349748 339167 352741	346032 359261 348311 360898 351601	
	483 460 462 921 239 20 81	339344 345099 346155 351173 366376 354026	369393 350732 360065 338645 345139	
	393 455 531 649 767 799 28	361093 353131 339484 362003 349839 348589	351729 349748 339167 352741 339344	
	443 927 474 209 206 551 561	348302 356072 360936 351059 341056 343614	345099 346155 351173 366376 354026	
	857 236 908 185 879 784 45	354838 336539 354361 351588 375123 344280	361093 353131 339484 362003 349839	
	221 12 599 705 825 414 978	364826 352055 354457 351179 335620 344699	348589 348302 356072 360936 351059	
	64 786 411 809 593 942 810	347216 348372 349590 359651 351149 355393	341056 343614 354838 336539 354361	
	712 741 839 507 20 665 716	350432 357010 343549 346884 332024 356829	351588 375123 344280 364826 352055	
	226 568 630 435 156 890 973	348606 355023 346410 354552 337504 342557	354457 351179 335620 344699 347216	
	35 674 18 257 39 969 314	330717 354157 343367 343013 349340 353371	348372 349590 359651 351149 355393	
	216 735 644 280 521 55 441	366408 345692 353006 336632 355632 345830	350432 357010 343549 346884 332024	
	466 350 604 179 91 443 686	339352 339983 343234 348738 354165 344157	356829 348606 355023 346410 354552	
	464 460 755 690 29 385 478	346923 353535 349901 355839 348482 342700	337504 342557 330717 354157 343367	
	185 275 451 573 301 469 182	351615 355087 339629 339354 350189 348940	343013 349340 353371 366408 345692	
	692 438 848 908 174 844 540	349439 350923 348361 357024 337824 349281	353006 336632 355632 345830 339352	
	47 251 334 514 601 938 45	355516 354845 343571 342636 334586 358872	339983 343234 348738 354165 344157	
	45 733 83 509 193 838 551	346603 354345 342452 350527 345861 347581	346923 353535 349901 355839 348482	
	574 575 29 112 202 832 37	349271 350789 337087 330643 354164 341464	342700 351615 355087 339629 339354	
	504 301 219 548 92 419 809	360399 342594 344308 341400 347468 346815	350189 348940 349439 350923 348361	
	618 263 701 665 514 35 531	340881 335941 351329 361307 356249 351620	357024 337824 349281 355516 354845	
	116 973 576 513 706 12 22	349659 358520 344578 351232 355328 357457	343571 342636 334586 358872 346603	
	252 850 925 178 778 955 642	351503 339777 351100 357639 352213 356462	354345 342452 350527 345861 347581	
	980 787 679 484 441 250 385	348179 344965 353328 350843 359674 358559	349271 350789 337087 330643 354164	
	533 21 194 151 284 895 168	341684 346513 346472 358614 355193 356944	341464 360399 342594 344308 341400	
	151 283 700 619 256 628 132	356269 350132 361961 335235 354018 334509	347468 346815 340881 335941 351329	
	315 992 506 919 843 431 449	354170 350954 340918 351654 363075 345286	361307 356249 351620 349659 358520	
	973 738 92 953 526 123 790	353682 349578 349687 349691 359615 345163	344578 351232 355328 357457 351503	
	319 726 175 852 747 721 355	340830 352346 339929 343343 334216 347844	339777 351100 357639 352213 356462	
	384 616 875 535 899 927 506	335052 354420 353395 353937 347748 353283	348179 344965 353328 350843 359674	
	508 556 638 175 900 144 446	365014 335807 339227 358904 370567 338622	358559 341684 346513 346472 358614	
	95 927 895 68 666 339 22	360488 337842 356180 364071 347464 341584	355193 356944 356269 350132 361961	
	544 463 812 863 541 987 67	355825 344829 356603 349596 350161 357598	335235 354018 334509 354170 350954	
	288 708 422 672 676 649 559	357323 349649 364546 353729 339670 340609	340918 351654 363075 345286 353682	
	928 577 65 788 485 55 963	357564 347173 350376 349799 347048 348562	349578 349687 349691 359615 345163	
	385 551 409 481 479 656 549	357595 358376 347015 351969 369779 358170	340830 352346 339929 343343 334216	
	145 996 923 41 811 735 904	347719 346731 348102 347531 359791 354965	347844 335052 354420 353395 353937	
	352 74 971 992 782 745 17	354438 352525 337845 344516 348756 369015	347748 353283 365014 335807 339227	
	811 394 576 739 323 994 879	350114 345809 343066 340717 348791 362777	358904 370567 338622 360488 337842	
	808 401 842 546 953 603 379	358631 336438 354724 349981 340522 343719	356180 364071 347464 341584 355825	
	784 259 280 929 607 204 322	350429 351921 350866 355788 351001 339999	344829 356603 349596 350161 357598	
	770 939 226 122 14 197 467	348320 344956 344280 355819 344537 348234	357323 349649 364546 353729 339670	
	148 294 484 959 40 412 50	351178 358631 338467 346051 343426 352500	340609 357564 347173 350376 349799	

	Input	Expected	Got	
	364 406 281 524 808 123 422	343748 350759 361913 354012 355999 357306	347048 348562 357595 358376 347015	
	113 78 801 897 690 82 178	354230 354876 346870 342617 358881 343410	351969 369779 358170 347719 346731	
	297 286 500 68 577 726 542	357084 353682 349399 353960 342869 344273	348102 347531 359791 354965 354438	
	591 275 9 92 921 493 403	346162 342307 342985 352729 353353 363867	352525 337845 344516 348756 369015	
	961 258 806 677 664 87 554	331835 352345 338604 356800 359663 336075	350114 345809 343066 340717 348791	
	824 563 976 289 641 130 186	341829 349683 356743 347404 346269 340845	362777 358631 336438 354724 349981	
	331 212 364 981 850 216 49	348800 346186 371330 350240 347447 341296	340522 343719 350429 351921 350866	
	427 942 943 371 569 305 815	346403 357221 367799 360282 334320 336291	355788 351001 339999 348320 344956	
	490 798 218 804 408 376 481	338413 347548 347842 344021 350428 347180	344280 355819 344537 348234 351178	
	73 464 35 249 27 12 539 668	344182 361713 342594 343791 352836 354952	358631 338467 346051 343426 352500	
	494 77 352 706 442 333 908	340728 361952 346535 331703 357544 341728	343748 350759 361913 354012 355999	
	658 734 687 953 677 410 874	359565 355031 354253 349150 339033 362754	357306 354230 354876 346870 342617	
	982 225 717 133 796 521 541	359918 352788 363383 345309 344359 345955	358881 343410 357084 353682 349399	
	172 2 966 636 390 568 15	370109 347309 338475 360206 348887 357125	353960 342869 344273 346162 342307	
	754 459 36 248 536 388 306	349676 349450 347042 347794 354112 343507	342985 352729 353353 363867 331835	
	978 73 214 989 159 253 294	341772 356346 349743 355503 370029 352340	352345 338604 356800 359663 336075	
	188 664 168 523 889 885 656	344869 352244 344676 352504 355000 348126	341829 349683 356743 347404 346269	
	37 758 549 210 761 868 198	349611 350886 341038 357353 348153 341795	340845 348800 346186 371330 350240	
	503 788 566 257 247 602 857	345268 351122 338325 341365 359270 342799	347447 341296 346403 357221 367799	
	783 342 163 114 767 377 455	351474 346044 353110 344511 351395 330285	360282 334320 336291 338413 347548	
	926 630 749 114 646 269 637	359503 352558 345428 352995 331047 329635	347842 344021 350428 347180 344182	
	888 507 645 925 265 547 487	354182 363718 357531 352866 367711 348549	361713 342594 343791 352836 354952	
	378 415 38 881 555 604 490	350512 347868 348734 352055 360940 366812	340728 361952 346535 331703 357544	
	802 558 347 585 900 510 51	351263 341501 344893 361975 344919 355457	341728 359565 355031 354253 349150	
	19 239 506 945 222 255 59	356731 357440 345609 343093 362475 353457	339033 362754 359918 352788 363383	
	868 877 49 756 384 694 34	346996 352337 341636 340175 355004 357342	345309 344359 345955 370109 347309	
	649 241 521 28 8 559 261	341533 344699 350854 355619 364067 345870	338475 360206 348887 357125 349676	
	563 515 752 717 73 99 303	355397 349045 349271 348413 353804 363405	349450 347042 347794 354112 343507	
	325 962 706 344 553 213 641	352390 338694 360886 351385 347269 360172	341772 356346 349743 355503 370029	
	775 820 53 996 697 102 104	353665 361292 349761 352837 343879 350636	352340 344869 352244 344676 352504	
	81 148 138 83 742 12 111	357531 339611 347625 351928 353813 339845	355000 348126 349611 350886 341038	
	750 923 372 666 791 476 383	351713 351843 338359 346354 358914 351860	357353 348153 341795 345268 351122	
	864 928 38 542 890 745 238	351883 335764 355213 353314 341038 346264	338325 341365 359270 342799 351474	
	795 958 880 571 778 933 919	349675 354073	346044 353110 344511 351395 330285	
	828 387 23 909 535 514 992		359503 352558 345428 352995 331047	
	277 878 455 380 153 180 398		329635 354182 363718 357531 352866	
	944 656 781 809 584 820 703		367711 348549 350512 347868 348734	
	826 917 941 622 875 821 545		352055 360940 366812 351263 341501	
	5 106 464 833 493 839 95		344893 361975 344919 355457 356731	
	381 705 439 658 583 895 390		357440 345609 343093 362475 353457	
	89 75 788 33 83 922 194 668		346996 352337 341636 340175 355004	
	94 897 846 11 839 820 238		357342 341533 344699 350854 355619	
	12 365 595 471 181 429 964		364067 345870 355397 349045 349271	
	373 876 697 78 315 356 14		348413 353804 363405 352390 338694	
	210 98 103 637 887 136 721		360886 351385 347269 360172 353665	
	161 331 741 255 580 939 266		361292 349761 352837 343879 350636	
	771 760 504 136 125 451 607		357531 339611 347625 351928 353813	
	659 232 923 32 108 621 462		339845 351713 351843 338359 346354	
	424 329 476 986 427 579 976		358914 351860 351883 335764 355213	
	666 68 697 827 751 438 82		353314 341038 346264 349675 354073	
	331 729 348 455 489			

	Input	Expected	Got	
	204 591 967 656 198 626 888			
	121 10 349 742 472 125 423			
	949 111 851 880 87 517 300			
	136 697 51 926 779 735 656			
	480 190 145 36 133 112 44			
	683 90 285 804 452 634 899			
	925 759 674 226 222 525 458			
	662 395 759 798 92 162 725			
	223 897 733 55 439 878 92			
	572 343 136 255 785 421 412			
	238 407 663 163 166 337 741			
	741 215 199 403 610 958 201			
	54 121 278 629 370 11 685			
	810 242 777 734 585 265 342			
	722 39 106 960 798 121 475			
	965 810 216 706 25 416 109			
	987 374 662 393 847 941 23			
	218 304 708 380 546 837 114			
	483 454 808 206 845 914 166			
	644 35 642 961 198 858 667			
	223 626 128 563 353 142 956			
	200 435 331 770 740 39 150			
	286 228 617 770 35 425 328			
	880 692 494 524 79 488 485			
	277 699 504 501 325 984 416			
	678 127 372 231 562 704 1			
	302 95 504 941 676 121 63			
	711 898 391 591 590 237 116			
	670 78 953 299 129 810 800			
	454 794 568 485 921 941 68			
	836 997 421 490 444 925 431			
	120 398 846 831 649 237 775			
	591 827 243 613 257 196 913			
	386 6 65 192 153 634 29 426			
	927 97 262 276 519 753 720			
	796 536 193 547 383 376 196			
	972 151 787 151 394 401 408			
	943 666 794 301 731 987 454			
	365 16 881 644 466 495 920			
	985 600 993 133 137 186 680			
	872 562 228 844 714 16 996			
	460 769 404 403 787 551 705			
	518 890 511 236 906 744 880			
	724 592 153 61 192 146 195			
	681 684 875 553 246 456 398			
	312 472 746 773 593 150 528			
	380 53 585 250 943 97			
	...snip... 810 868 710 851			
	300 456 567 378 878 948 426			
	826 427 911 17 677 757 939			
	737 461 652 178 310 135 417			

	Input	Expected	Got	
	174 400 893 22 908 438 184 128 148 36 428 956 955 159 834 903 585 660 331 848 29 360 605 968 97 66 620 275 728 107 692 903 507 937 277 767 375 461 895 875 849 676 831 157 835 666 60 772 678 743 620 708 103 226 676 552 644 649 827 725 756 871 980 264 809 257 383 536 70 279 412 272 955 243 429 142 261 841 914 940 937 886 0 392 112 28 945 109 677 772 186 786 996 166 402 805 775 785 341 845 64 105 469 371 701 250 513 962 444 427 254 381 314 606 125 778 635 70 239 664 195 425 450 191 943 852 348 718 990 41 916 54 147 385 426 848 988 291 162 432 71 769 165 737 375 290 867 362 713 107 27 908 884 829 99 828 34 799 898 24 840 814 430 987 552 208 187 892 500 702 324 571 823 841 308 550 131 175 913 196 634 292 104 519 473 555 699 507 354 949 883 195 116 314 534 20 522 74 912 374 776 588 945 951 429 605 501 912 781 414 109 767 58 565 638 532 121 337 39 475 639 923 670 755 589 557 127 111 631 39 486 759 979 783 710 408 389 211 672 522 978 133 289 36 699 928 568 172 617 608 647 256 883 670 363 472 227 490 935 210 881 773 969 860 909 679 620 298 242 293 820 220 778 461 257 829 389 177 1 359 137 1 967 20 671 331 844 250 173 132 812 55 257 781 915 166 812 536 464 406 181 636 627 311 450 236 141 191 413 142 550 551 143 518 923 166 201 768 768 726 252 580 133 509 713 49 676 877 937 140 284 470 129 263 781 579 499 922 770 264 417 673 815 912 543 739 79 96 859 847 822 111			

	Input	Expected	Got	
	780 956 972 493 357 648 371 646 141 7 116 270 270 249 201 769 524 323 385 941 996 201 853 891 292 284 987 503 132 810 966 912 118 938 757 475 587 480 473 80 487 589 350 757 190 903 878 714 226 264 655 575 817 861 466 461 145 806 316 629 616 282 893 86 572 651 913 511 131 386 591 619 975 293 728 517 196 959 232 775 223 239 702 392 100 520 205 598 326 521 227 294 155 121 732 727 124 645 239 255 31 182 226 358 476 307 876 24 266 108 151 841 699 205 585 152 726 790 750 404 663 329 51 818 450 783 545 926 429 136 534 812 671 112 171 147 419 47 523 37 507 675 230 206 880 815 358 958 957 460 715 620 790 766 438 592 901 336 519 682 824 53 495 495 165 18 994 937 417 518 974 924 545 205 482 425 372 193 736 330 653 451 302 443 569 93 36 822 781 555 857 957 960 704 453 125 722 799 62 139 317 389 415 862 946 897 640 318 90 728 0 744 531 303 539 452 748 575 274 529 482 131 486 794 187 291 272 909 91 334 400 760 75 815 975 21 713 615 340 803 695 692 899 226 995 439 678 95 366 304 624 201 788 111 995 975 754 267 885 845 954 285 958 29 101 933 403 166 900 743 321 595 787 221 821 135 12 851 230 378 155 855 931 943 318 279 919 72 898 156 270 852 441 228 234 894 161 637 412 61 732 734 656 519 955 829 654 967 680 885 697 835 92 629 779 410 908 698 834 806 206 104 11 999 332 597 894 493 586 306 906 318 40 562 189 347 391 196 666 423 433 364 259 525 993 390 287 253 440 121 411 646 226 422 645 910 371			

	Input	Expected	Got
	539 404 957 198 310 275 590 225 817 290 968 13 956 392 798 320 651 675 665 393 962 918 833 83 682 479 661 456 124 572 828 16 328 785 214 990 413 156 567 230 446 536 595 403 280 393 75 283 68 741 676 382 11 509 817 693 988 479 150 112 403 978 128 83 763 694 425 176 203 993 758 1 881 353 756 161 98 832 444 518 925 120 900 936 981 718 630 969 549 132 433 304 110 914 739 225 608 164 754 163 157 512 165 38 218 273 199 668 105 995 187 30 467 439 319 448 509 301 417 410 433 851 714 895 765 453 120 725 970 874 241 127 739 758 518 309 31 69 977 489 65 516 871 884 956 190 333 465 491 750 876 276 953 942 171 70 748 292 148 718 518 389 197 609 499 715 918 530 137 896 371 554 412 243 438 720 785 771 186 277 874 414 905 827 708 77 250 456 369 398 526 239 139 724 849 638 791 767 520 928 15 892 482 428 487 921 500 272 692 38 901 566 452 807 746 513 884 996 969 605 746 496 844 885 572 693 875 363 461 395 644 476 287 126 256 774 399 757 47 92 795 300 10 600 107 108 113 991 104 434 948 202 282 793 87 854 486 962 570 299 358 214 128 997 692 736 124 92 493 523 536 641 823 898 593 283 7 58 626 463 492 575 18 775 368 105 981		

	Input	Expected	Got	
✓	1000 1000	509471 507271 502426 517188 507270 498733	509471 507271 502426 517188 507270	✓
	215 923 498 357 772 502 132	494538 502161 496070 492225 486663 486567	498733 494538 502161 496070 492225	
	616 805 955 133 812 658 124	506930 507761 486797 489808 485138 508038	486663 486567 506930 507761 486797	
	184 621 164 233 975 58 864	495673 487723 496147 507046 492780 507007	489808 485138 508038 495673 487723	
	928 355 803 353 829 987 582	502891 500675 501581 493112 511085 498340	496147 507046 492780 507007 502891	
	845 640 88 412 563 938 769	504060 504376 492948 491301 490513 522372	500675 501581 493112 511085 498340	
	687 440 253 655 597 561 788	506985 507761 506950 510362 504703 503771	504060 504376 492948 491301 490513	
	409 219 912 593 192 428 178	494842 508293 502519 499401 490819 512319	522372 506985 507761 506950 510362	
	167 838 394 95 193 197 800	495017 505229 504509 497150 496463 502698	504703 503771 494842 508293 502519	
	375 537 383 220 177 823 633	496789 502062 505787 490327 498896 501864	499401 490819 512319 495017 505229	
	92 761 754 131 553 360 786	495898 498702 497568 498463 498881 497834	504509 497150 496463 502698 496789	
	150 921 926 911 492 190 856	492706 500449 503557 499574 507200 500559	502062 505787 490327 498896 501864	
	684 618 34 852 456 781 299	493056 513905 498082 508817 514170 501864	495898 498702 497568 498463 498881	
	2 978 100 377 867 835 949	480073 512242 512995 502471 515980 502941	497834 492706 500449 503557 499574	
	396 658 582 840 771 689 971	490473 491252 496311 508922 508480 495821	507200 500559 493056 513905 498082	
	676 49 109 826 970 387 89	503053 491583 487580 487559 504915 489393	508817 514170 501864 480073 512242	
	462 577 945 498 195 979 350	521455 510442 506035 500009 498448 498396	512995 502471 515980 502941 490473	
	4 112 2 6 91 454 735 310	500483 499498 505451 501478 504621 489554	491252 496311 508922 508480 495821	
	289 36 59 299 619 899 70	513291 496345 499413 507352 498781 480013	503053 491583 487580 487559 504915	
	660 223 98 709 684 276 31	495009 495709 493500 499439 509193 508916	489393 521455 510442 506035 500009	
	72 365 845 649 310 343 197	490722 501872 504305 508104 494941 503555	498448 498396 500483 499498 505451	
	641 46 201 754 48 559 197	509793 497589 488459 501266 495195 507810	501478 504621 489554 513291 496345	
	502 646 507 143 682 918 442	491877 504460 506375 500214 506120 507319	499413 507352 498781 480013 495009	
	653 818 864 313 393 962 374	500419 474452 495279 499575 491889 473807	495709 493500 499439 509193 508916	
	77 238 405 149 603 250 799	507509 492533 492685 481995 497672 500692	490722 501872 504305 508104 494941	
	913 594 996 554 640 549 660	494019 491224 487642 497223 498876 504628	503555 509793 497589 488459 501266	
	40 108 857 894 106 717 37	497789 488516 504320 504634 505451 504609	495195 507810 491877 504460 506375	
	788 635 831 794 453 695 459	499691 503072 504872 489941 498991 484765	500214 506120 507319 500419 474452	
	846 9 834 276 247 239 425	507716 503100 505684 495159 502609 503774	495279 499575 491889 473807 507509	
	202 842 576 467 436 924 373	507152 497583 506192 501911 509437 495612	492533 492685 481995 497672 500692	
	428 473 34 820 933 891 714	497188 497236 496313 513947 497285 492252	494019 491224 487642 497223 498876	
	39 608 751 180 244 934 326	498051 495881 485537 489038 513651 500411	504628 497789 488516 504320 504634	
	49 629 785 248 638 971 524	495088 494136 486760 501548 491412 496945	505451 504609 499691 503072 504872	
	237 211 301 439 53 878 906	501584 485671 496322 501870 491211 492729	489941 498991 484765 507716 503100	
	841 802 279 269 628 665 89	496673 508007 503625 488857 494190 498976	505684 495159 502609 503774 507152	
	561 557 155 953 517 258 485	490931 506094 505755 506557 509830 491446	497583 506192 501911 509437 495612	
	113 192 811 163 821 948 763	499178 500599 491510 507023 506233 490632	497188 497236 496313 513947 497285	
	811 920 639 400 483 940 191	491915 485159 500527 508806 494459 501719	492252 498051 495881 485537 489038	
	536 818 449 729 973 80 998	497589 488812 521302 489991 497898 487303	513651 500411 495088 494136 486760	
	601 746 439 162 303 594 467	489089 505808 516006 495408 500507 487283	501548 491412 496945 501584 485671	
	172 204 952 286 396 115 449	492104 499990 493936 495679 508948 500108	496322 501870 491211 492729 496673	
	217 416 564 380 688 203 780	508772 502792 497233 499116 507569 507619	508007 503625 488857 494190 498976	
	171 143 971 59 314 420 788	499470 483148 483633 492457 507804 489475	490931 506094 505755 506557 509830	
	287 500 138 888 598 929 402	484348 498434 489420 507920 502339 491206	491446 499178 500599 491510 507023	
	901 875 870 426 79 174 712	510297 509226 498850 500883 498151 501368	506233 490632 491915 485159 500527	
	475 642 513 44 58 77 424	511985 494410 491544 495677 486074 518516	508806 494459 501719 497589 488812	
	746 280 556 269 423 879 328	486931 494045 501173 501926 498808 479895	521302 489991 497898 487303 489089	
	737 651 116 376 503 606 264	496498 495401 506067 487550 484642 502818	505808 516006 495408 500507 487283	
	102 535 19 355 410 241 781	498772 494426 496806 495798 501511 496566	492104 499990 493936 495679 508948	
	489 415 493 964 409 6 360	506147 500192 503739 510864 494006 508009	500108 508772 502792 497233 499116	

	Input	Expected	Got	
	819 435 136 565 715 692 186	512760 509787 490343 515838 514335 501309	507569 507619 499470 483148 483633	
	491 923 514 580 574 982 957	493548 522147 494921 504901 496991 509186	492457 507804 489475 484348 498434	
	77 588 573 531 123 592 887	493478 492455 503759 506689 492521 500628	489420 507920 502339 491206 510297	
	885 833 668 726 601 514 42	515791 490043 494007 503238 499032 498539	509226 498850 500883 498151 501368	
	10 872 402 830 308 538 747	491793 505037 500654 510941 498683 495859	511985 494410 491544 495677 486074	
	23 230 934 866 505 800 447	502329 498709 494493 505710 501997 502250	518516 486931 494045 501173 501926	
	79 783 404 509 723 329 40	485993 482717 490110 488680 495952 501565	498808 479895 496498 495401 506067	
	199 922 279 84 107 300 811	493919 507096 496215 496832 492371 501331	487550 484642 502818 498772 494426	
	708 814 853 71 686 608 901	487583 493568 500262 509859 486862 504162	496806 495798 501511 496566 506147	
	346 146 648 722 729 934 588	490753 493973 494429 509788 497875 503986	500192 503739 510864 494006 508009	
	234 735 35 666 870 791 175	497835 491879 511378 486366 506167 496862	512760 509787 490343 515838 514335	
	593 473 567 792 395 847 229	501900 487315 492834 496603 492446 495628	501309 493548 522147 494921 504901	
	854 147 392 563 313 597 634	496464 489913 501548 501082 500943 492989	496991 509186 493478 492455 503759	
	351 205 887 698 704 887 420	503202 500482 509147 510226 507509 493223	506689 492521 500628 515791 490043	
	433 822 8 667 909 396 685	494153 492056 502848 493087 505804 512811	494007 503238 499032 498539 491793	
	779 539 860 372 12 780 517	498769 479539 489113 498921 493287 506567	505037 500654 510941 498683 495859	
	759 979 98 966 126 842 529	502416 483638 499883 492374 501816 501158	502329 498709 494493 505710 501997	
	439 439 515 142 997 754 840	503547 492325 498849 507604 503669 473968	502250 485993 482717 490110 488680	
	701 641 260 134 463 621 153	502676 492067 496878 488765 491506 505427	495952 501565 493919 507096 496215	
	724 369 839 503 260 51 228	495467 512732 484429 500044 514714 489348	496832 492371 501331 487583 493568	
	625 183 97 384 162 547 350	518179 518754 496433 492127 500333 497547	500262 509859 486862 504162 490753	
	288 389 231 79 180 98 222	483440 507776 515121 516831 491422 500394	493973 494429 509788 497875 503986	
	177 852 414 878 846 27 364	517134 510534 490918 508536 495568 489424	497835 491879 511378 486366 506167	
	661 0 870 386 369 61 889	510162 496603 492482 520517 505367 489035	496862 501900 487315 492834 496603	
	629 464 469 606 648 566 343	519362 493402 508062 498167 508606 514158	492446 495628 496464 489913 501548	
	810 113 693 451 854 277 882	509462 501044 493659 518550 499974 510060	501082 500943 492989 503202 500482	
	35 727 456 212 932 871 443	487695 499812 491314 498744 503820 512993	509147 510226 507509 493223 494153	
	778 898 807 439 898 29 825	488867 499080 499217 489606 494993 496685	492056 502848 493087 505804 512811	
	619 90 67 600 555 536 207	500907 506298 500761 508926 495410 508375	498769 479539 489113 498921 493287	
	555 455 550 717 920 595 168	490596 494906 505952 486506 484182 503785	506567 502416 483638 499883 492374	
	775 224 403 810 952 859 22	500671 485537 485710 496601 517322 497281	501816 501158 503547 492325 498849	
	884 82 817 662 980 625 453	502264 505758 511528 502286 495370 508181	507604 503669 473968 502676 492067	
	230 6 279 201 97 698 802 4	504353 508197 508212 516371 485582 507180	496878 488765 491506 505427 495467	
	586 9 911 41 911 628 962	492608 502874 496599 516558 492968 495425	512732 484429 500044 514714 489348	
	858 797 89 83 552 899 387	485951 496813 510824 504416 508630 507734	518179 518754 496433 492127 500333	
	411 273 271 846 91 285 826	499373 498288 491102 508439 492948 510840	497547 483440 507776 515121 516831	
	68 738 409 74 369 610 523	499097 505440 485522 493259 504385 497346	491422 500394 517134 510534 490918	
	419 764 879 6 125 790 47 36	475633 499663 499858 509254 501415 500259	508536 495568 489424 510162 496603	
	419 361 895 568 450 330 120	486086 501395 499319 510902 501815 493993	492482 520517 505367 489035 519362	
	701 717 883 975 988 729 66	490524 498469 507996 497029 492263 507268	493402 508062 498167 508606 514158	
	625 908 134 715 669 560 437	493892 489481 500713 506326 514075 500105	509462 501044 493659 518550 499974	
	631 84 856 396 315 214 521	503157 490124 501660 496894 514079 505443	510060 487695 499812 491314 498744	
	106 262 558 877 975 805 445	493859 500752 504459 492681 483171 492904	503820 512993 488867 499080 499217	
	778 135 917 479 852 152 454	491034 507546 490560 506814 490920 501260	489606 494993 496685 500907 506298	
	192 234 872 169 494 358 884	507619 503785 494183 498545 516722 492140	500761 508926 495410 508375 490596	
	163 919 673 794 355 530 542	518039 507861 494213 511774 494608 504398	494906 505952 486506 484182 503785	
	670 744 64 128 358 974 5	517009 494724 495422 505475 501163 506542	500671 485537 485710 496601 517322	
	686 779 802 464 266 71 943	494791 501991 511434 526704 487339 515006	497281 502264 505758 511528 502286	
	118 576 750 662 810 622 831	497272 495062 507091 494430 515341 508179	495370 508181 504353 508197 508212	
	304 981 67 819 900 741 965	487917 488553 509798 507803 502692 504522	516371 485582 507180 492608 502874	
	607 271 508 277 367 572 758	513397 513632 492470 509430 510124 493987	496599 516558 492968 495425 485951	

	Input	Expected	Got	
	78 546 763 764 677 918 580	513785 501302 510388 495314 510812 507477	496813 510824 504416 508630 507734	
	943 989 523 413 565 625 427	505213 511514 499600 490385 496942 502506	499373 498288 491102 508439 492948	
	727 248 258 383 229 325 202	502720 500446 513760 488656 509815 492484	510840 499097 505440 485522 493259	
	481 66 520 88 689 380 717	509537 493077 503230 501069 506908 487197	504385 497346 475633 499663 499858	
	409 952 827 839 850 591 603	504086 493639 500432 500451 501355 505429	509254 501415 500259 486086 501395	
	879 861 535 174 202 410 587	491075 505612 500933 489369 500571 507260	499319 510902 501815 493993 490524	
	768 36 14 847 284 624 231	491963 512845 494582 495793 505515 509450	498469 507996 497029 492263 507268	
	865 301 785 346 720 305 786	494906 482233 523162 507032 498267 504517	493892 489481 500713 506326 514075	
	409 685 855 170 989 683 9	499486 501926 487921 497800 501823 484598	500105 503157 490124 501660 496894	
	191 626 964 422 487 499 596	500799 493580 505578 508562 499672 497867	514079 505443 493859 500752 504459	
	689 910 535 809 946 901 657	495522 514200 493560 481084 508259 507330	492681 483171 492904 491034 507546	
	582 525 240 447 179 25 793	486472 491416 496694 483445 500784 479953	490560 506814 490920 501260 507619	
	899 683 931 660 720 786 831	522803 511468 489936 491424 500752 499326	503785 494183 498545 516722 492140	
	62 469 192 253 447 509 676	503898 504518 501140 505650 506205 501350	518039 507861 494213 511774 494608	
	934 8 624 976 270 160 785	480294 503029 483759 505146 505610 502282	504398 517009 494724 495422 505475	
	568 413 794 150 291 34 949	475260 503144 497641 506212 492897 494221	501163 506542 494791 501991 511434	
	470 412 742 369 447 25 29	496547 494508 507253 517249 508089 506472	526704 487339 515006 497272 495062	
	167 812 212 229 633 405 835	494942 514112 501596 503089 482310 505318	507091 494430 515341 508179 487917	
	81 914 863 367 274 487 343	495376 510499 497358 501729 505450 492168	488553 509798 507803 502692 504522	
	897 999 481 465 413 627 968	515169 510197 523406 504431 495516 500515	513397 513632 492470 509430 510124	
	704 14 917 526 426 12 895	508537 489465 499166 499242 493021 523836	493987 513785 501302 510388 495314	
	873 37 276 392 201 489 622	503281 500571 504336 498643 486106 509082	510812 507477 505213 511514 499600	
	187 246 457 268 512 320 987	495310 492788 502540 513699 500140 501685	490385 496942 502506 502720 500446	
	138 159 683 35 511 164 853	504147 507965 516428 482716 486182 496003	513760 488656 509815 492484 509537	
	924 143 173 980 157 90 858	494379 495620 500927 503370 483622 496485	493077 503230 501069 506908 487197	
	935 454 753 160 844 381 553	501797 504024 494091 500497 503122 480760	504086 493639 500432 500451 501355	
	45 222 527 584 820 984 204	483676 503977 474461 500732 498154 503273	505429 491075 505612 500933 489369	
	332 656 192 471 167 875 858	503888 513880 499462 508282 511731 501051	500571 507260 491963 512845 494582	
	678 391 63 954 534 236 286	498599 485819 503497 493011 502422 523149	495793 505515 509450 494906 482233	
	692 679 144 627 133 249 140	481088 496211 500370 484799 491627 512882	523162 507032 498267 504517 499486	
	977 631 693 375 853 220 959	480158 499080 508626 511617 507340 498051	501926 487921 497800 501823 484598	
	674 556 164 358 564 708 181	486038 492370 502141 507202 511352 501186	500799 493580 505578 508562 499672	
	83 583 392 762 974 455 68	498155 492612 500690 514304 488732 495265	497867 495522 514200 493560 481084	
	508 692 355 552 371 499 532	503255 501456 495032 524738 496095 502848	508259 507330 486472 491416 496694	
	504 749 672 834 732 365 561	495512 494939 517961 495601 506698 502061	483445 500784 479953 522803 511468	
	585 937 520 611 845 36 322	495708 512482 502230 509925 512479 492794	489936 491424 500752 499326 503898	
	409 744 503 844 327 247 606	495456 510023 508524 480926 485393 512672	504518 501140 505650 506205 501350	
	653 703 675 514 395 382 66	503813 496775 514244 499980 491554 502226	480294 503029 483759 505146 505610	
	118 881 950 622 982 622 808	499369 508512 480636 499669 483395 494586	502282 475260 503144 497641 506212	
	714 339 369 652 276 890 615	487341 481555 499408 497985 513910 506821	492897 494221 496547 494508 507253	
	473 278 937 234 23 793 79	489934 498884 505662 506983 497127 508291	517249 508089 506472 494942 514112	
	702 40 685 708 743 712 222	493162 493937 506443 483858 507121 521000	501596 503089 482310 505318 495376	
	490 94 640	502941 503116 478175 502207 499237 506840	510499 497358 501729 505450 492168	
	608 328 943 583 310 565 391	494233 502796 487183 500095 488398 497863	515169 510197 523406 504431 495516	
	25 905 761 677 533 3 292	495525 516488 495568 499535 489793 498699	500515 508537 489465 499166 499242	
	359 633 582 593 656 727 672	498880 506031 504085 508354 503006 487365	493021 523836 503281 500571 504336	
	711 767 710 419 863 774 993	487975 487482 501928 495750 487785 494811	498643 486106 509082 495310 492788	
	...snip... 10 72 782 520	487151 501536 488629 489627 493265 505047	502540 513699 500140 501685 504147	
	191 742 786 942 30 806 47	502393 499663 508648 504773 489099 502727	507965 516428 482716 486182 496003	
	428 501 301 763 98 337 67	481000 491267 506533 497820 498037 508965	494379 495620 500927 503370 483622	
	772 386 773 597 941 46 707	500897 498938 492334 501617 496843 502984	496485 501797 504024 494091 500497	

	Input	Expected	Got	
	879 182 926 271	503844 483799 514437 500045 502566 504931	503122 480760 483676 503977 474461	
	248 945 534 320 727 54 863	495348 493102 486355 505046 513152 484103	500732 498154 503273 503888 513880	
	470 841 805 852 999 204 632	500673 498634 493034 490001 511496 486919	499462 508282 511731 501051 498599	
	500 505 395 951 843 815 75	489963 486666 502749 506865 507534 506021	485819 503497 493011 502422 523149	
	581 940 672 522 338 379 754	504898 478510 503127 492975 491196 511127	481088 496211 500370 484799 491627	
	521 657 377 769 602 911 441	500884 495361 492352 488946 498599 494728	512882 480158 499080 508626 511617	
	330 966 304 152 159 462 4	510264 489772 510972 508219 515328 515908	507340 498051 486038 492370 502141	
	158 666 988 11 524 384 314	497796 486708 502915 512450 499046 493096	507202 511352 501186 498155 492612	
	719 551 389 300 843 61 174	505795 513912 472989 502408 492470 496621	500690 514304 488732 495265 503255	
	181 792 280 702 449 658 823	496762 490033 506117 491885 484426 507830	501456 495032 524738 496095 502848	
	51 569 264 733 535 569 885	504684 494573 492431 500549 517016 502199	495512 494939 517961 495601 506698	
	694 383 241 205 401 230 216	509138 496635 495882 507625 517822 499602	502061 495708 512482 502230 509925	
	925 966 882 644 517 271 296	502865 490499 502016 513206 492680 506914	512479 492794 495456 510023 508524	
	360 684 823 893 476 103 596	516159 509042 499846 505081 506619 508709	480926 485393 512672 503813 496775	
	277 761 771 680 331 36 414	491499 499636 496761 498193	514244 499980 491554 502226 499369	
	218 957 299 265 340 893 470		508512 480636 499669 483395 494586	
	741 123 38 19 89 920 663		487341 481555 499408 497985 513910	
	958 543 960 670 227 783 563		506821 489934 498884 505662 506983	
	55 886 511 332 0 283 12 331		497127 508291 493162 493937 506443	
	671 778 901 628 430 166 320		483858 507121 521000 502941 503116	
	323 988 61 446 26 80 887		478175 502207 499237 506840 494233	
	298 96 845 193 56 867 420		502796 487183 500095 488398 497863	
	191 430 827 429 294 511 429		495525 516488 495568 499535 489793	
	929 876 760 600 654 662 228		498699 498880 506031 504085 508354	
	436 180 548 759 169 961 557		503006 487365 487975 487482 501928	
	547 394 444 198 490 641 391		495750 487785 494811 487151 501536	
	546 508 164 89 939 991 518		488629 489627 493265 505047 502393	
	585 503 948 514 731 60 114		499663 508648 504773 489099 502727	
	385 74 342 822 255 242 933		481000 491267 506533 497820 498037	
	776 203 491 323 597 287 521		508965 500897 498938 492334 501617	
	87 929 265 985 437 429 426		496843 502984 503844 483799 514437	
	728 772 945 313 627 893 179		500045 502566 504931 495348 493102	
	358 305 293 96 380 987 270		486355 505046 513152 484103 500673	
	987 229 203 763 785 694 438		498634 493034 490001 511496 486919	
	382 982 960 822 263 225 159		489963 486666 502749 506865 507534	
	700 6 586 781 778 883 94		506021 504898 478510 503127 492975	
	406 128 626 116 433 919 212		491196 511127 500884 495361 492352	
	165 259 482 152 488 38 267		488946 498599 494728 510264 489772	
	625 732 706 8 66 18 182 329		510972 508219 515328 515908 497796	
	243 693 382 601 631 163 379		486708 502915 512450 499046 493096	
	514 609 137 642 235 606 428		505795 513912 472989 502408 492470	
	507 170 593 766 653 98 606		496621 496762 490033 506117 491885	
	691 365 232 775 71 592 194		484426 507830 504684 494573 492431	
	441 126 523 684 819 257 637		500549 517016 502199 509138 496635	
	451 420 17 317 382 506 312		495882 507625 517822 499602 502865	
	617 112 740 476 283 685 242		490499 502016 513206 492680 506914	
	288 783 201 331 501 785 458		516159 509042 499846 505081 506619	
	572 729 652 366 855 528 50		508709 491499 499636 496761 498193	
	26 785 688 829 558 57 147			
	940 915 459 909 380 551 386			
	15 236 980 303 20 533 986			

	Input	Expected	Got	
	521 318 444 445 47 449 811 902 977 214 929 114 254 110 24 663 257 964 578 68 226 958 619 612 973 856 944 628 228 478 966 749 796 763 546 196 212 358 450 541 924 731 7 178 842 32 841 99 348 771 520 574 730 139 538 55 347 483 36 575 313 354 324 109 117 223 657 329 581 108 870 857 839 230 35 681 262 228 781 610 999 653 537 81 792 427 489 140 910 525 715 575 879 392 685 349 615 342 678 548 802 901 405 642 131 792 323 745 20 456 707 19 109 244 453 902 672 942 394 934 819 461 510 50 853 547 399 820 241 430 368 44 331 125 38 814 917 361 559 937 818 266 309 279 863 762 533 887 56 927 173 875 389 683 277 594 582 677 767 176 459 135 572 790 613 610 956 530 971 515 468 141 133 129 773 996 243 306 883 299 586 57 526 327 92 155 273 27 184 40 203 643 528 775 785 141 737 741 671 60 608 491 554 742 972 327 738 215 985 974 866 571 383 744 250 475 900 524 854 436 564 57 80 92 184 217 585 273 959 257 334 567 100 888 309 73 567 400 640 552 726 507 124 109 603 374 936 503 250 791 292 815 200 372 259 737 589 197 10 900 454 344 468 906 584 129 979 151 881 620 704 607 479 180 68 82 906 5 938 157 148 230 324 348 954 935 437 895 132 448 796 938 144 616 845 81 97 176 232 979 148 288 938 627 468 7 62 727 364 0 884 512 582 560 212 536 847 650 431 980 450 579 270 946 195 467 27 645 644 612 624 144 900 914 124 721 921 186 800 637 538 684 149 120 596 714 8 443 716 791 775 166 371 398 112 918 865 492 563 861			

	Input	Expected	Got	
	104 539 6 356 454 130 429 727 668 229 365 206 265 866 678 861 580 686 657 648 477 784 814 200 182 279 119 48 123 682 261 579 574 267 935 28 749 717 107 769 946 824 975 212 691 5 425 623 691 434 272 521 219 438 721 753 69 840 153 192 875 415 771 449 34 59 829 784 776 288 553 74 113 881 286 156 886 64 779 930 498 403 803 69 842 524 823 911 717 976 456 944 743 227 745 778 638 926 562 414 214 467 841 679 700 127 835 939 543 615 869 394 18 24 463 212 900 638 476 617 967 932 913 62 511 10 840 150 936 754 916 503 222 757 182 922 237 18 861 132 985 82 526 355 106 342 568 7 332 44 976 651 328 242 714 839 252 554 341 541 309 610 44 883 719 578 805 308 948 19 441 933 453 319 289 912 661 209 919 994 605 247 997 933 489 711 124 94 266 818 635 927 780 31 810 499 609 967 160 558 986 601 843 440 920 484 352 934 693 623 280 650 222 277 583 64 989 60 158 607 230 145 534 10 528 344 861 489 311 21 399 650 974 243 442 895 727 146 181 773 121 461 423 343 738 359 407 79 771 917 686 1 414 572 11 294 916 872 784 580 246 183 230 220 778 672 467 506 818 648 631 939 461 406 634 200 765 394 631 536 663 670 889 430 242 900 724 159			

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 5

Correct

Mark 10.00 out of 10.00

[DeleteColumn]

Cho ma trận kích thước $m \times n$ chứa các số nguyên, các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n . Cho số nguyên i nhiệm vụ của bạn là viết chương trình xóa cột thứ i của ma trận.

Đầu vào

Đầu vào từ bàn phím gồm $m + 2$ dòng:

- Dòng đầu tiên chứa 2 số nguyên m, n cách nhau bởi một dấu cách ($m, n \leq 1000$).
- m dòng tiếp theo, mỗi dòng chứa n số, cách nhau bởi một dấu cách, biểu diễn ma trận đã cho.
- Dòng cuối cùng chứa số nguyên i .

Đầu ra

In ra màn hình m dòng, mỗi dòng gồm $n - 1$ số, cách nhau bởi một dấu cách, biểu diễn ma trận sau khi đã xóa cột thứ i .

For example:

Input	Result
3 3	3 5
1 3 5	4 7
2 4 7	4 2
3 4 2	
1	

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7
8  int main()
9  {
10     ios_base::sync_with_stdio(0);
11     cin.tie(0);
12     int m,n; cin>>m>>n;
13     int a[m][n];
14     for(int i=0;i<m;i++)
15     {
16         for(int j=0;j<n;j++)
17         {
18             cin>>a[i][j];
19         }
20     }
21     int id; cin>>id;
22     for(int i=0;i<m;i++)
23     {
24         for(int j=0;j<n;j++)
25         {
26             if(j!=id-1) cout<<a[i][j]<<" ";
27         }
28         cout<<el;
29     }
30     return 0;
31 }
32
33
```

	Input	Expected	Got	
✓	3 3 1 3 5 2 4 7 3 4 2 1	3 5 4 7 4 2	3 5 4 7 4 2	✓
✓	10 20 610 132 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 515 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 185 381 120 695 333 985 387 360 615 369 154 966 934 341 352 948 690 516 26 953 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142 2	610 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 185 120 695 333 985 387 360 615 369 154 966 934 341 352 948 690 516 26 953 251 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142	610 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 185 120 695 333 985 387 360 615 369 154 966 934 341 352 948 690 516 26 953 251 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 6

Correct

Mark 10.00 out of 10.00

[DeleteRow]

Cho ma trận kích thước $m \times n$ chứa các số nguyên, các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n . Cho số nguyên i , nhiệm vụ của bạn là xóa hàng thứ i của ma trận.

Đầu vào

Đầu vào từ bàn phím gồm $m + 2$ dòng.

- Dòng đầu tiên chứa 2 số nguyên m, n cách nhau bởi một dấu cách ($m, n \leq 1000$);
- m dòng tiếp theo, mỗi dòng chứa n số, cách nhau bởi một dấu cách, biểu diễn ma trận đã cho;
- Dòng tiếp theo chứa số nguyên i .

Đầu ra

In ra màn hình $m - 1$ dòng, mỗi dòng gồm n số, cách nhau bởi một dấu cách, biểu diễn ma trận sau khi đã xóa hàng thứ i .

For example:

Input	Result
3 3	2 4 7
1 3 5	3 4 2
2 4 7	
3 4 2	
1	

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7
8  int main()
9  {
10     ios_base::sync_with_stdio(0);
11     cin.tie(0);
12     int m,n; cin>>m>>n;
13     int a[m][n];
14     for(int i=0;i<m;i++)
15     {
16         for(int j=0;j<n;j++)
17         {
18             cin>>a[i][j];
19         }
20     }
21     int id; cin>>id;
22     for(int i=0;i<m;i++)
23     {
24         if(i==id-1) continue;
25         for(int j=0;j<n;j++)
26         {
27             cout<<a[i][j]<<" ";
28         }
29         cout<<el;
30     }
31     return 0;
32 }
33

```

	Input	Expected	Got	
✓	3 3 1 3 5 2 4 7 3 4 2 1	2 4 7 3 4 2	2 4 7 3 4 2	✓
✓	10 20 610 132 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 721 582 40 682 98 163 778 711 561 571 586 901 515 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142 6	610 132 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 515 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142	610 132 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 515 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 7

Correct

Mark 10.00 out of 10.00

[BookingTicket]

Rạp chiếu phim áp dụng hệ thống đặt vé online để hỗ trợ khách hàng đặt vé nhanh và tiện lợi hơn.

Các ghế trong rạp chiếu phim được lưu trữ trong một mảng 2 chiều với các hàng tương ứng với các hàng ghế và các cột tương ứng với số ghế trong hàng.

Vị trí ghế đã bán sẽ chứa giá trị 1, ngược lại, nếu chưa được bán sẽ được gán giá trị 0.

Hệ thống chỉ cho phép người dùng đặt 1 vé trong mỗi giao dịch. Người dùng chọn vị trí ghế muốn đặt và hệ thống sẽ kiểm tra.

Nếu ghế chưa được bán, hệ thống thông báo đặt chỗ thành công và người dùng có thể thanh toán. Ngược lại, hệ thống báo ghế đã bán và kết thúc giao dịch.

Viết chương trình mô phỏng hệ thống trên. Chương trình nhận đầu vào là ma trận chứa thông tin phòng chiếu (ghế đã đặt và ghế trống) và ghế người dùng chọn (gồm vị trí hàng và cột).

Nếu tất cả các ghế đều chưa bán, in ra thông báo "Your seat is booked. Please complete the payment." Ngược lại, in ra thông báo ghế đã bán, ví dụ, "Seat 12 is sold."

Các hàng, các cột được đánh số từ 1.

For example:

Input	Result
3 4 1 0 0 1 0 1 0 1 1 0 1 1 1 1	Seat 11 is sold.

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7
8  int main()
9  {
10     ios_base::sync_with_stdio(0);
11     cin.tie(0);
12     int m,n; cin>>m>>n;
13     int a[m][n];
14     for(int i=0;i<m;i++)
15     {
16         for(int j=0;j<n;j++)
17         {
18             cin>>a[i][j];
19         }
20     }
21     int x,y; cin>>x>>y;
22     if(a[x-1][y-1]==1)
23     {
24         cout<<"Seat "<<x<<y<<" is sold.";
25     }
26     else
27     {
28         cout<<"Your seat is booked. Please complete the payment.";
29     }

```

```
30     return 0;  
31 }  
32  
33
```

	Input	Expected	Got	
✓	3 4 1 0 0 1 0 1 0 1 1 0 1 1 1 1	Seat 11 is sold.	Seat 11 is sold.	✓
✓	4 6 0 1 0 1 1 0 0 1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 1 1	Your seat is booked. Please complete the payment.	Your seat is booked. Please complete the payment.	✓
✓	4 6 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 4 5	Seat 45 is sold.	Seat 45 is sold.	✓
✓	4 6 1 1 0 1 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 4 3	Seat 43 is sold.	Seat 43 is sold.	✓

	Input	Expected	Got	
✓	4 6 0 0 1 1 0 1 0 0 0 1 0 1 0 0 1 1 0 1 0 1 1 0 0 0 4 1	Your seat is booked. Please complete the payment.	Your seat is booked. Please complete the payment.	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 8

Correct

Mark 10.00 out of 10.00

[Determinant]

Định thức, trong đại số tuyến tính, là một hàm cho mỗi ma trận vuông A , tương ứng với số vô hướng, ký hiệu là **det**(A). Ý nghĩa hình học của định thức là tỷ lệ xích cho thể tích khi A được coi là một biến đổi tuyến tính. Định thức được sử dụng để giải (và biện luận) các hệ phương trình đại số tuyến tính.

Hãy viết chương trình để tính định thức của ma trận vuông 3×3 .

Đầu vào: 3 dòng, mỗi dòng gồm ba số nguyên cách nhau bởi dấu cách là các giá trị của ma trận vuông 3×3 .

Đầu ra: định thức của ma trận.

For example:

Input	Result
1 2 3 4 5 6 7 8 9	0

Answer: (penalty regime: 0 %)

```

1  #include <iostream>
2  using namespace std;
3  int determinant(int matrix[3][3]) {
4      return matrix[0][0] * (matrix[1][1] * matrix[2][2] - matrix[1][2] * matrix[2][1]) -
5             matrix[0][1] * (matrix[1][0] * matrix[2][2] - matrix[1][2] * matrix[2][0]) +
6             matrix[0][2] * (matrix[1][0] * matrix[2][1] - matrix[1][1] * matrix[2][0]);
7  }
8
9  int main() {
10     int matrix[3][3];
11     for (int i = 0; i < 3; ++i) {
12         for (int j = 0; j < 3; ++j) {
13             cin >> matrix[i][j];
14         }
15     }
16     cout << determinant(matrix) << endl;
17
18     return 0;
19 }
20

```

	Input	Expected	Got	
✓	1 2 3 4 5 6 7 8 9	0	0	✓
✓	90 -82 -93 -26 3 52 -96 46 -27	328782	328782	✓
✓	241 -795 -761 488 963 -699 238 -66 300	506079318	506079318	✓
✓	455 738 973 495 -584 30 223 -394 -774	435684086	435684086	✓
✓	-854 -772 -276 -312 -621 413 494 -442 -607	-611838686	-611838686	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 9

Correct

Mark 10.00 out of 10.00

[Sudoku]

Sudoku (ban đầu có tên gọi là **Number Place**) là một trò chơi câu đố sắp xếp chữ số dựa trên logic theo tổ hợp.

Mục tiêu của trò chơi là điền các chữ số vào một lưới (grid) 9×9 sao cho mỗi cột, mỗi hàng, và mỗi phần trong số chín lưới con 3×3 cấu tạo nên lưới chính (cũng gọi là "hộp" (box), "khối" (block), hoặc "khu vực" (region)) đều chứa tất cả các chữ số từ 1 tới 9.

Cơ cấu điều chỉnh câu đố cung cấp một lưới đã được hoàn thành một phần, mỗi câu đố được thiết lập tốt có một giải pháp hoàn thành duy nhất.

Với một bài toán sudoku, một số được xem như là điền đúng vị trí khi:

- Trong hàng ngang 9 ô của số đó, không có số nào lặp lại số này.
- Trong hàng dọc 9 ô của số đó, không có số nào lặp lại số này.
- Trong ô vuông 3×3 của số đó, không có số nào lặp lại số này.

Viết chương trình nhận đầu vào là một Sudoku kích thước 9×9 với 9 lưới con kích thước 3×3 và một số n cùng với tọa độ vị trí (i, j) người chơi muốn đặt n trên bàn chơi.

Kiểm tra xem vị trí đặt n có hợp lệ hay không. Nếu có, in ra màn hình "Valid place.", ngược lại, in thông báo "Invalid place."

Nếu số n không hợp lệ, in thông báo "Invalid number."

Note: trong bàn chơi Sudoku được nhập vào, nếu vị trí (i, j) trong bàn chơi chưa có bất cứ số nào thì vị trí lưu giá trị 0. Tọa độ (i, j) được đánh số từ 1

For example:

Input	Result
0 2 0 3 1 6 5 7 8 8 6 7 5 0 4 1 9 0 0 1 0 8 9 0 2 4 6 9 3 0 7 0 5 6 2 4 6 0 2 9 4 1 0 3 5 7 4 5 2 6 3 9 8 1 3 5 4 0 7 0 8 1 9 1 0 8 4 0 9 3 0 2 2 9 6 1 3 8 4 5 7 4 1 1	Valid place.

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7  int a[1003][1003];
8  int main()
9  {
10     ios_base::sync_with_stdio(0);
11     cin.tie(0);
12     for(int i=1;i<=9;i++)
13     {
14         for(int j=1;j<=9;j++)
15         {
16             cin>>a[i][j];
17         }
18     }

```

```
18     }
19     int x,y,val; cin>>val>>x>>y;
20     if(val<1||val>9)
21     {
22         cout<<"Invalid number.";
23         return 0;
24     }
25     if(x<1||x>9||y>9||y<1)
26     {
27         cout<<"Invalid place.";
28         return 0;
29     }
30     for(int i=1;i<=9;i++)
31     {
32         if(a[i][y]==val)
33         {
34             cout<<"Invalid place.";
35             return 0;
36         }
37     }
38     for(int j=1;j<=9;j++)
39     {
40         if(a[x][j]==val)
41         {
42             cout<<"Invalid place.";
43             return 0;
44         }
45     }
46     //int tmpx= x/3*3+(x%3!=0);
47     //int tmpy=y/3*3+(y%3!=0);
48     //for(int i=tmpx;i<=tmpx+2;i++)
49     //{
50         //for(int j=tmpy;j<=tmpy+2;j++)
51         //{
52         //    if(a[i][j]==val)
```

	Input	Expected	Got	
✓	0 2 0 3 1 6 5 7 8 8 6 7 5 0 4 1 9 0 0 1 0 8 9 0 2 4 6 9 3 0 7 0 5 6 2 4 6 0 2 9 4 1 0 3 5 7 4 5 2 6 3 9 8 1 3 5 4 0 7 0 8 1 9 1 0 8 4 0 9 3 0 2 2 9 6 1 3 8 4 5 7 10 1 2	Invalid number.	Invalid number.	✓
✓	0 2 0 3 1 6 5 7 8 8 6 7 5 0 4 1 9 0 0 1 0 8 9 0 2 4 6 9 3 0 7 0 5 6 2 4 6 0 2 9 4 1 0 3 5 7 4 5 2 6 3 9 8 1 3 5 4 0 7 0 8 1 9 1 0 8 4 0 9 3 0 2 2 9 6 1 3 8 4 5 7 4 1 1	Valid place.	Valid place.	✓

	Input	Expected	Got	
✓	0 2 0 3 1 6 5 7 8 8 6 7 5 0 4 1 9 0 0 1 0 8 9 0 2 4 6 9 3 0 7 0 5 6 2 4 6 0 2 9 4 1 0 3 5 7 4 5 2 6 3 9 8 1 3 5 4 0 7 0 8 1 9 1 0 8 4 0 9 3 0 2 2 9 6 1 3 8 4 5 7 4 10 12	Invalid place.	Invalid place.	✓
✓	0 2 0 3 1 6 5 7 8 8 6 7 5 0 4 1 9 0 0 1 0 8 9 0 2 4 6 9 3 0 7 0 5 6 2 4 6 0 2 9 4 1 0 3 5 7 4 5 2 6 3 9 8 1 3 5 4 0 7 0 8 1 9 1 0 8 4 0 9 3 0 2 2 9 6 1 3 8 4 5 7 5 4 5	Invalid place.	Invalid place.	✓
✓	0 2 0 3 1 6 5 7 8 8 6 7 5 0 4 1 9 0 0 1 0 8 9 0 2 4 6 9 3 0 7 0 5 6 2 4 6 0 2 9 4 1 0 3 5 7 4 5 2 6 3 9 8 1 3 5 4 0 7 0 8 1 9 1 0 8 4 0 9 3 0 2 2 9 6 1 3 8 4 5 7 8 5 2	Valid place.	Valid place.	✓
✓	0 2 0 3 1 6 5 7 8 8 6 7 5 0 4 1 9 0 0 1 0 8 9 0 2 4 6 9 3 0 7 0 5 6 2 4 6 0 2 9 4 1 0 3 5 7 4 5 2 6 3 9 8 1 3 5 4 0 7 0 8 1 9 1 0 8 4 0 9 3 0 2 2 9 6 1 3 8 4 5 7 0 1 1	Invalid number.	Invalid number.	✓

	Input	Expected	Got	
✓	0 2 0 3 1 6 5 7 8 8 6 7 5 0 4 1 9 0 0 1 0 8 9 0 2 4 6 9 3 0 7 0 5 6 2 4 6 0 2 9 4 1 0 3 5 7 4 5 2 6 3 9 8 1 3 5 4 0 7 0 8 1 9 1 0 8 4 0 9 3 0 2 2 9 6 1 3 8 4 5 7 4 1 -1	Invalid place.	Invalid place.	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 10

Correct

Mark 10.00 out of 10.00

[SumPerRowK]

Viết chương trình nhận đầu vào là mảng $(2 \times n \times m)$, in ra màn hình tổng của các số trên hàng thứ k .

Kết quả làm tròn đến số thập phân thứ hai.

For example:

Input	Result
4 3 68 62 11 77 56 20 26 37 93 1 44 72 3	156.00
2 3 -142.223270493 8.67644117219 42.2916318896 -161.952521645 16.4510205899 -33.3353243042 2	-178.84

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7  double a[1003][1003];
8  int main()
9  {
10     ios_base::sync_with_stdio(0);
11     cin.tie(0);
12     int m,n;cin>>m>>n;
13     for(int i=1;i<=m;i++)
14     {
15         for(int j=1;j<=n;j++)
16         {
17             cin>>a[i][j];
18         }
19     }
20     double ans=0;
21     int k; cin>>k;
22     for(int j=1;j<=n;j++)
23     {
24         ans+=a[k][j];
25     }
26     cout<<fixed<<setprecision(2)<<ans;
27     return 0;
28 }
29
30

```


	Input	Expected	Got	
✓	4 5 7.74554504 19.72632458 -12.39398584 -2.54492907 -40.34440142 -1.47685049 17.84090848 -5.31887803 -21.32172779 12.3452829 27.53414824 8.52408783 14.60320842 -23.13836256 -10.78727531 25.32200324 -37.14053272 21.21635967 -34.45060259 1.08197459 4	-23.97	-23.97	✓
✓	4 5 -0.249939244489 5.56860692483 19.3441100248 11.046189467 5.79537597558 -22.8161066687 -13.6999016082 9.23012650439 -10.5527379305 3.47595357588 -13.0101083027 31.7516025096 21.3182703279 -7.08924324152 -16.2804045465 -18.2916395788 -8.36602698047 17.9153664145 -16.270578329 5.84882643391 3	16.69	16.69	✓
✓	4 3 68 62 11 77 56 20 26 37 93 1 44 72 3	156.00	156.00	✓
✓	2 3 -142.223270493 8.67644117219 42.2916318896 -161.952521645 16.4510205899 -33.3353243042 2	-178.84	-178.84	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Back to Course