

Programming using C

week 06 practice session and coding

Name:**S.Jananejaipratha**

Department:**AIML-'B'**

Roll No.:**241501079**

Complexity: **T**
Correctness
Maximal cost of
1.00
If flag
is present

Given an array A of sorted integers and another non-negative integer k, find if there exists 2 indices i and j such that $A[i] - A[j] = k, i < j$.

Input format

1. First line is number of test cases T. Following T lines contain:
2. k, followed by n integers of the array
3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn't

Example

Input

3
3 1 3 5
4

Output

1

Input

3
3 1 3 5
99

Output

0

Answer: `pairwise_difference.cpp`

```
1 pairwise_difference.cpp
2 int main()
3 {
4     int t;
5     scanf("%d",&t);
6     while(t-->0)
7     {
8         int n;
9         scanf("%d",&n);
10        int a[n];
11        for(int i=0;i<n;i++)
12        {
13            scanf("%d",&a[i]);
14        }
15        int k;
16        scanf("%d",&k);
17        int flag=0;
18        for(int i=0;i<n-1;i++){
19            for(int j=i+1;j<n;j++){
20                if(a[j]-a[i]==k || a[i]-a[j]==k){flag=1;break;}
21            }
22        }
23        printf("%d\n",flag);
24    }
25    return 0;
26 }
```

	Input	Expected	Got
✓	3 3 1 3 5 4		✓
✓	3 3 1 3 5 99		✓

Passed all tests! ✓

Activate Windows
Go to Settings to activate Windows.

2

100%

Accepted

100%

100%

100%

Sam loves chocolate and needs help for the 1st day of the year. Starting of the year, x is considered from 1 to V . On days where x is odd, Sam will buy x chocolates, on days where x is even, Sam will not purchase any chocolates.

Complete the code in the editor so that for each day N , where $1 \leq N \leq V$ is given as an input, the number of chocolates Sam purchased during days 1 through N is printed as an output. This is a function only challenge, no input is handled for you by the boilerplate code in the editor.

Input Format

The program takes an array of integers as a parameter.

The boilerplate code in the editor handles reading the following input from stdin, according to the following signature (all integers are int) and calling calculate(x).

The first line of input contains an integer T (the number of test cases), each line i of the T subsequent lines describes the i th test case as an integer N (the number of days).

Constraints

$1 \leq T \leq 100$
 $1 \leq N_i \leq 1 \times 10^6$
 $1 \leq x \leq V \leq 10^6$

Output Format

For each test case, T is an array value method should print the total number of chocolates Sam purchased by day N on a new line.

Sample Input 0

3
1
2
3

Sample Output 0

1
3
4

Explanation

Test Case 0: $N = 1$
Sam buys 1 chocolate on day 1, giving us a total of 1 chocolate. Thus, we print 1 as our answer.

Test Case 1: $N = 2$
Sam buys 1 chocolate on day 1 and 0 on day 2. This gives us a total of 1 chocolate. Thus, we print 1 as a new line.

Test Case 2: $N = 3$
Sam buys 1 chocolate on day 1, 0 on day 2, and 3 on day 3. This gives us a total of 4 chocolates. Thus, we print 4 as a new line.

Answer: (correct output is 4)

```
def calculate(x):  
    # Complete this function  
    # Return the total number of chocolates  
    # purchased by day x  
    # Example:  
    # Input: 1  
    # Output: 1  
    # Input: 2  
    # Output: 1  
    # Input: 3  
    # Output: 4  
    # Input: 4  
    # Output: 5  
    # Input: 5  
    # Output: 8  
    # Input: 6  
    # Output: 9  
    # Input: 7  
    # Output: 12  
    # Input: 8  
    # Output: 13  
    # Input: 9  
    # Output: 17  
    # Input: 10  
    # Output: 18  
    # Input: 11  
    # Output: 22  
    # Input: 12  
    # Output: 23  
    # Input: 13  
    # Output: 27  
    # Input: 14  
    # Output: 28  
    # Input: 15  
    # Output: 33  
    # Input: 16  
    # Output: 34  
    # Input: 17  
    # Output: 39  
    # Input: 18  
    # Output: 40  
    # Input: 19  
    # Output: 45  
    # Input: 20  
    # Output: 46  
    # Input: 21  
    # Output: 51  
    # Input: 22  
    # Output: 52  
    # Input: 23  
    # Output: 57  
    # Input: 24  
    # Output: 58  
    # Input: 25  
    # Output: 63  
    # Input: 26  
    # Output: 64  
    # Input: 27  
    # Output: 69  
    # Input: 28  
    # Output: 70  
    # Input: 29  
    # Output: 75  
    # Input: 30  
    # Output: 76  
    # Input: 31  
    # Output: 81  
    # Input: 32  
    # Output: 82  
    # Input: 33  
    # Output: 87  
    # Input: 34  
    # Output: 88  
    # Input: 35  
    # Output: 93  
    # Input: 36  
    # Output: 94  
    # Input: 37  
    # Output: 99  
    # Input: 38  
    # Output: 100  
    # Input: 39  
    # Output: 105  
    # Input: 40  
    # Output: 106  
    # Input: 41  
    # Output: 111  
    # Input: 42  
    # Output: 112  
    # Input: 43  
    # Output: 117  
    # Input: 44  
    # Output: 118  
    # Input: 45  
    # Output: 123  
    # Input: 46  
    # Output: 124  
    # Input: 47  
    # Output: 129  
    # Input: 48  
    # Output: 130  
    # Input: 49  
    # Output: 135  
    # Input: 50  
    # Output: 136  
    # Input: 51  
    # Output: 141  
    # Input: 52  
    # Output: 142  
    # Input: 53  
    # Output: 147  
    # Input: 54  
    # Output: 148  
    # Input: 55  
    # Output: 153  
    # Input: 56  
    # Output: 154  
    # Input: 57  
    # Output: 159  
    # Input: 58  
    # Output: 160  
    # Input: 59  
    # Output: 165  
    # Input: 60  
    # Output: 166  
    # Input: 61  
    # Output: 171  
    # Input: 62  
    # Output: 172  
    # Input: 63  
    # Output: 177  
    # Input: 64  
    # Output: 178  
    # Input: 65  
    # Output: 183  
    # Input: 66  
    # Output: 184  
    # Input: 67  
    # Output: 189  
    # Input: 68  
    # Output: 190  
    # Input: 69  
    # Output: 195  
    # Input: 70  
    # Output: 196  
    # Input: 71  
    # Output: 201  
    # Input: 72  
    # Output: 202  
    # Input: 73  
    # Output: 207  
    # Input: 74  
    # Output: 208  
    # Input: 75  
    # Output: 213  
    # Input: 76  
    # Output: 214  
    # Input: 77  
    # Output: 219  
    # Input: 78  
    # Output: 220  
    # Input: 79  
    # Output: 225  
    # Input: 80  
    # Output: 226  
    # Input: 81  
    # Output: 231  
    # Input: 82  
    # Output: 232  
    # Input: 83  
    # Output: 237  
    # Input: 84  
    # Output: 238  
    # Input: 85  
    # Output: 243  
    # Input: 86  
    # Output: 244  
    # Input: 87  
    # Output: 249  
    # Input: 88  
    # Output: 250  
    # Input: 89  
    # Output: 255  
    # Input: 90  
    # Output: 256  
    # Input: 91  
    # Output: 261  
    # Input: 92  
    # Output: 262  
    # Input: 93  
    # Output: 267  
    # Input: 94  
    # Output: 268  
    # Input: 95  
    # Output: 273  
    # Input: 96  
    # Output: 274  
    # Input: 97  
    # Output: 279  
    # Input: 98  
    # Output: 280  
    # Input: 99  
    # Output: 285  
    # Input: 100  
    # Output: 286  
    # Input: 101  
    # Output: 291  
    # Input: 102  
    # Output: 292  
    # Input: 103  
    # Output: 297  
    # Input: 104  
    # Output: 298  
    # Input: 105  
    # Output: 303  
    # Input: 106  
    # Output: 304  
    # Input: 107  
    # Output: 309  
    # Input: 108  
    # Output: 310  
    # Input: 109  
    # Output: 315  
    # Input: 110  
    # Output: 316  
    # Input: 111  
    # Output: 321  
    # Input: 112  
    # Output: 322  
    # Input: 113  
    # Output: 327  
    # Input: 114  
    # Output: 328  
    # Input: 115  
    # Output: 333  
    # Input: 116  
    # Output: 334  
    # Input: 117  
    # Output: 339  
    # Input: 118  
    # Output: 340  
    # Input: 119  
    # Output: 345  
    # Input: 120  
    # Output: 346  
    # Input: 121  
    # Output: 351  
    # Input: 122  
    # Output: 352  
    # Input: 123  
    # Output: 357  
    # Input: 124  
    # Output: 358  
    # Input: 125  
    # Output: 363  
    # Input: 126  
    # Output: 364  
    # Input: 127  
    # Output: 369  
    # Input: 128  
    # Output: 370  
    # Input: 129  
    # Output: 375  
    # Input: 130  
    # Output: 376  
    # Input: 131  
    # Output: 381  
    # Input: 132  
    # Output: 382  
    # Input: 133  
    # Output: 387  
    # Input: 134  
    # Output: 388  
    # Input: 135  
    # Output: 393  
    # Input: 136  
    # Output: 394  
    # Input: 137  
    # Output: 399  
    # Input: 138  
    # Output: 400  
    # Input: 139  
    # Output: 405  
    # Input: 140  
    # Output: 406  
    # Input: 141  
    # Output: 411  
    # Input: 142  
    # Output: 412  
    # Input: 143  
    # Output: 417  
    # Input: 144  
    # Output: 418  
    # Input: 145  
    # Output: 423  
    # Input: 146  
    # Output: 424  
    # Input: 147  
    # Output: 429  
    # Input: 148  
    # Output: 430  
    # Input: 149  
    # Output: 435  
    # Input: 150  
    # Output: 436  
    # Input: 151  
    # Output: 441  
    # Input: 152  
    # Output: 442  
    # Input: 153  
    # Output: 447  
    # Input: 154  
    # Output: 448  
    # Input: 155  
    # Output: 453  
    # Input: 156  
    # Output: 454  
    # Input: 157  
    # Output: 459  
    # Input: 158  
    # Output: 460  
    # Input: 159  
    # Output: 465  
    # Input: 160  
    # Output: 466  
    # Input: 161  
    # Output: 471  
    # Input: 162  
    # Output: 472  
    # Input: 163  
    # Output: 477  
    # Input: 164  
    # Output: 478  
    # Input: 165  
    # Output: 483  
    # Input: 166  
    # Output: 484  
    # Input: 167  
    # Output: 489  
    # Input: 168  
    # Output: 490  
    # Input: 169  
    # Output: 495  
    # Input: 170  
    # Output: 496  
    # Input: 171  
    # Output: 501  
    # Input: 172  
    # Output: 502  
    # Input: 173  
    # Output: 507  
    # Input: 174  
    # Output: 508  
    # Input: 175  
    # Output: 513  
    # Input: 176  
    # Output: 514  
    # Input: 177  
    # Output: 519  
    # Input: 178  
    # Output: 520  
    # Input: 179  
    # Output: 525  
    # Input: 180  
    # Output: 526  
    # Input: 181  
    # Output: 531  
    # Input: 182  
    # Output: 532  
    # Input: 183  
    # Output: 537  
    # Input: 184  
    # Output: 538  
    # Input: 185  
    # Output: 543  
    # Input: 186  
    # Output: 544  
    # Input: 187  
    # Output: 549  
    # Input: 188  
    # Output: 550  
    # Input: 189  
    # Output: 555  
    # Input: 190  
    # Output: 556  
    # Input: 191  
    # Output: 561  
    # Input: 192  
    # Output: 562  
    # Input: 193  
    # Output: 567  
    # Input: 194  
    # Output: 568  
    # Input: 195  
    # Output: 573  
    # Input: 196  
    # Output: 574  
    # Input: 197  
    # Output: 579  
    # Input: 198  
    # Output: 580  
    # Input: 199  
    # Output: 585  
    # Input: 200  
    # Output: 586  
    # Input: 201  
    # Output: 591  
    # Input: 202  
    # Output: 592  
    # Input: 203  
    # Output: 597  
    # Input: 204  
    # Output: 598  
    # Input: 205  
    # Output: 603  
    # Input: 206  
    # Output: 604  
    # Input: 207  
    # Output: 609  
    # Input: 208  
    # Output: 610  
    # Input: 209  
    # Output: 615  
    # Input: 210  
    # Output: 616  
    # Input: 211  
    # Output: 621  
    # Input: 212  
    # Output: 622  
    # Input: 213  
    # Output: 627  
    # Input: 214  
    # Output: 628  
    # Input: 215  
    # Output: 633  
    # Input: 216  
    # Output: 634  
    # Input: 217  
    # Output: 639  
    # Input: 218  
    # Output: 640  
    # Input: 219  
    # Output: 645  
    # Input: 220  
    # Output: 646  
    # Input: 221  
    # Output: 651  
    # Input: 222  
    # Output: 652  
    # Input: 223  
    # Output: 657  
    # Input: 224  
    # Output: 658  
    # Input: 225  
    # Output: 663  
    # Input: 226  
    # Output: 664  
    # Input: 227  
    # Output: 669  
    # Input: 228  
    # Output: 670  
    # Input: 229  
    # Output: 675  
    # Input: 230  
    # Output: 676  
    # Input: 231  
    # Output: 681  
    # Input: 232  
    # Output: 682  
    # Input: 233  
    # Output: 687  
    # Input: 234  
    # Output: 688  
    # Input: 235  
    # Output: 693  
    # Input: 236  
    # Output: 694  
    # Input: 237  
    # Output: 699  
    # Input: 238  
    # Output: 700  
    # Input: 239  
    # Output: 705  
    # Input: 240  
    # Output: 706  
    # Input: 241  
    # Output: 711  
    # Input: 242  
    # Output: 712  
    # Input: 243  
    # Output: 717  
    # Input: 244  
    # Output: 718  
    # Input: 245  
    # Output: 723  
    # Input: 246  
    # Output: 724  
    # Input: 247  
    # Output: 729  
    # Input: 248  
    # Output: 730  
    # Input: 249  
    # Output: 735  
    # Input: 250  
    # Output: 736  
    # Input: 251  
    # Output: 741  
    # Input: 252  
    # Output: 742  
    # Input: 253  
    # Output: 747  
    # Input: 254  
    # Output: 748  
    # Input: 255  
    # Output: 753  
    # Input: 256  
    # Output: 754  
    # Input: 257  
    # Output: 759  
    # Input: 258  
    # Output: 760  
    # Input: 259  
    # Output: 765  
    # Input: 260  
    # Output: 766  
    # Input: 261  
    # Output: 771  
    # Input: 262  
    # Output: 772  
    # Input: 263  
    # Output: 777  
    # Input: 264  
    # Output: 778  
    # Input: 265  
    # Output: 783  
    # Input: 266  
    # Output: 784  
    # Input: 267  
    # Output: 789  
    # Input: 268  
    # Output: 790  
    # Input: 269  
    # Output: 795  
    # Input: 270  
    # Output: 796  
    # Input: 271  
    # Output: 801  
    # Input: 272  
    # Output: 802  
    # Input: 273  
    # Output: 807  
    # Input: 274  
    # Output: 808  
    # Input: 275  
    # Output: 813  
    # Input: 276  
    # Output: 814  
    # Input: 277  
    # Output: 819  
    # Input: 278  
    # Output: 820  
    # Input: 279  
    # Output: 825  
    # Input: 280  
    # Output: 826  
    # Input: 281  
    # Output: 831  
    # Input: 282  
    # Output: 832  
    # Input: 283  
    # Output: 837  
    # Input: 284  
    # Output: 838  
    # Input: 285  
    # Output: 843  
    # Input: 286  
    # Output: 844  
    # Input: 287  
    # Output: 849  
    # Input: 288  
    # Output: 850  
    # Input: 289  
    # Output: 855  
    # Input: 290  
    # Output: 856  
    # Input: 291  
    # Output: 861  
    # Input: 292  
    # Output: 862  
    # Input: 293  
    # Output: 867  
    # Input: 294  
    # Output: 868  
    # Input: 295  
    # Output: 873  
    # Input: 296  
    # Output: 874  
    # Input: 297  
    # Output: 879  
    # Input: 298  
    # Output: 880  
    # Input: 299  
    # Output: 885  
    # Input: 300  
    # Output: 886  
    # Input: 301  
    # Output: 891  
    # Input: 302  
    # Output: 892  
    # Input: 303  
    # Output: 897  
    # Input: 304  
    # Output: 898  
    # Input: 305  
    # Output: 903  
    # Input: 306  
    # Output: 904  
    # Input: 307  
    # Output: 909  
    # Input: 308  
    # Output: 910  
    # Input: 309  
    # Output: 915  
    # Input: 310  
    # Output: 916  
    # Input: 311  
    # Output: 921  
    # Input: 312  
    # Output: 922  
    # Input: 313  
    # Output: 927  
    # Input: 314  
    # Output: 928  
    # Input: 315  
    # Output: 933  
    # Input: 316  
    # Output: 934  
    # Input: 317  
    # Output: 939  
    # Input: 318  
    # Output: 940  
    # Input: 319  
    # Output: 945  
    # Input: 320  
    # Output: 946  
    # Input: 321  
    # Output: 951  
    # Input: 322  
    # Output: 952  
    # Input: 323  
    # Output: 957  
    # Input: 324  
    # Output: 958  
    # Input: 325  
    # Output: 963  
    # Input: 326  
    # Output: 964  
    # Input: 327  
    # Output: 969  
    # Input: 328  
    # Output: 970  
    # Input: 329  
    # Output: 975  
    # Input: 330  
    # Output: 976  
    # Input: 331  
    # Output: 981  
    # Input: 332  
    # Output: 982  
    # Input: 333  
    # Output: 987  
    # Input: 334  
    # Output: 988  
    # Input: 335  
    # Output: 993  
    # Input: 336  
    # Output: 994  
    # Input: 337  
    # Output: 999  
    # Input: 338  
    # Output: 1000  
    # Input: 339  
    # Output: 1005  
    # Input: 340  
    # Output: 1006  
    # Input: 341  
    # Output: 1011  
    # Input: 342  
    # Output: 1012  
    # Input: 343  
    # Output: 1017  
    # Input: 344  
    # Output: 1018  
    # Input: 345  
    # Output: 1023  
    # Input: 346  
    # Output: 1024  
    # Input: 347  
    # Output: 1029  
    # Input: 348  
    # Output: 1030  
    # Input: 349  
    # Output: 1035  
    # Input: 350  
    # Output: 1036  
    # Input: 351  
    # Output: 1041  
    # Input: 352  
    # Output: 1042  
    # Input: 353  
    # Output: 1047  
    # Input: 354  
    # Output: 1048  
    # Input: 355  
    # Output: 1053  
    # Input: 356  
    # Output: 1054  
    # Input: 357  
    # Output: 1059  
    # Input: 358  
    # Output: 1060  
    # Input: 359  
    # Output: 1065  
    # Input: 360  
    # Output: 1066  
    # Input: 361  
    # Output: 1071  
    # Input: 362  
    # Output: 1072  
    # Input: 363  
    # Output: 1077  
    # Input: 364  
    # Output: 1078  
    # Input: 365  
    # Output: 1083  
    # Input: 366  
    # Output: 1084  
    # Input: 367  
    # Output: 1089  
    # Input: 368  
    # Output: 1090  
    # Input: 369  
    # Output: 1095  
    # Input: 370  
    # Output: 1096  
    # Input: 371  
    # Output: 1101  
    # Input: 372  
    # Output: 1102  
    # Input: 373  
    # Output: 1107  
    # Input: 374  
    # Output: 1108  
    # Input: 375  
    # Output: 1113  
    # Input: 376  
    # Output: 1114  
    # Input: 377  
    # Output: 1119  
    # Input: 378  
    # Output: 1120  
    # Input: 379  
    # Output: 1125  
    # Input: 380  
    # Output: 1126  
    # Input: 381  
    # Output: 1131  
    # Input: 382  
    # Output: 1132  
    # Input: 383  
    # Output: 1137  
    # Input: 384  
    # Output: 1138  
    # Input: 385  
    # Output: 1143  
    # Input: 386  
    # Output: 1144  
    # Input: 387  
    # Output: 1149  
    # Input: 388  
    # Output: 1150  
    # Input: 389  
    # Output: 1155  
    # Input: 390  
    # Output: 1156  
    # Input: 391  
    # Output: 1161  
    # Input: 392  
    # Output: 1162  
    # Input: 393  
    # Output: 1167  
    # Input: 394  
    # Output: 1168  
    # Input: 395  
    # Output: 1173  
    # Input: 396  
    # Output: 1174  
    # Input: 397  
    # Output: 1179  
    # Input: 398  
    # Output: 1180  
    # Input: 399  
    # Output: 1185  
    # Input: 400  
    # Output: 1186  
    # Input: 401  
    # Output: 1191  
    # Input: 402  
    # Output: 1192  
    # Input: 403  
    # Output: 1197  
    # Input: 404  
    # Output: 1198  
    # Input: 405  
    # Output: 1203  
    # Input: 406  
    # Output: 1204  
    # Input: 407  
    # Output: 1209  
    # Input: 408  
    # Output: 1210  
    # Input: 409  
    # Output: 1215  
    # Input: 410  
    # Output: 1216  
    # Input: 411  
    # Output: 1221  
    # Input: 412  
    # Output: 1222  
    # Input: 413  
    # Output: 1227  
    # Input: 414  
    # Output: 1228  
    # Input: 415  
    # Output: 1233  
    # Input: 416  
    # Output: 1234  
    # Input: 417  
    # Output: 1239  
    # Input: 418  
    # Output: 1240  
    # Input: 419  
    # Output: 1245  
    # Input: 420  
    # Output: 1246  
    # Input: 421  
    # Output: 1251  
    # Input: 422  
    # Output: 1252  
    # Input: 423  
    # Output: 1257  
    # Input: 424  
    # Output: 1258  
    # Input: 425  
    # Output: 1263  
    # Input: 426  
    # Output: 1264  
    # Input: 427  
    # Output: 1269  
    # Input: 428  
    # Output: 1270  
    # Input: 429  
    # Output: 1275  
    # Input: 430  
    # Output: 1276  
    # Input: 431  
    # Output: 1281  
    # Input: 432  
    # Output: 1282  
    # Input: 433  
    # Output: 1287  
    # Input: 434  
    # Output: 1288  
    # Input: 435  
    # Output: 1293  
    # Input: 436  
    # Output: 1294  
    # Input: 437  
    # Output: 1299  
    # Input: 438  
    # Output: 1300  
    # Input: 439  
    # Output: 1305  
    # Input: 440  
    # Output: 1306  
    # Input: 441  
    # Output: 1311  
    # Input: 442  
    # Output: 1312  
    # Input: 443  
    # Output: 1317  
    # Input: 444  
    # Output: 1318  
    # Input: 445  
    # Output: 1323  
    # Input: 446  
    # Output: 1324  
    # Input: 447  
    # Output: 1329  
    # Input: 448  
    # Output: 1330  
    # Input: 449  
    # Output: 1335  
    # Input: 450  
    # Output: 1336  
    # Input: 451  
    # Output: 1341  
    # Input: 452  
    # Output: 1342  
    # Input: 453  
    # Output: 1347  
    # Input: 454  
    # Output: 1348  
    # Input: 455  
    # Output: 1353  
    # Input: 456  
    # Output: 1354  
    # Input: 457  
    # Output: 1359  
    # Input: 458  
    # Output: 1360  
    # Input: 459  
    # Output: 1365  
    # Input: 460  
    # Output: 1366  
    # Input: 461  
    # Output: 1371  
    # Input: 462  
    # Output: 1372  
    # Input: 463  
    # Output: 1377  
    # Input: 464  
    # Output: 1378  
    # Input: 465  
    # Output: 1383  
    # Input: 466  
    # Output: 1384  
    # Input: 467  
    # Output: 1389  
    # Input: 468  
    # Output: 1390  
    # Input: 469  
    # Output: 1395  
    # Input: 470  
    # Output: 1396  
    # Input: 471  
    # Output: 1401  
    # Input: 472  
    # Output: 1402  
    # Input: 473  
    # Output: 1407  
    # Input: 474  
    # Output: 1408  
    # Input: 475  
    # Output: 1413  
    # Input: 476  
    # Output: 1414  
    # Input: 477  
    # Output: 1419  
    # Input: 478  
    # Output: 1420  
    # Input: 479  
    # Output: 1425  
    # Input: 480  
    # Output: 1426  
    # Input: 481  
    # Output: 1431  
    # Input: 482  
    # Output: 1432  
    # Input: 483  
    # Output: 1437  
    # Input: 484  
    # Output: 1438  
    # Input: 485  
    # Output: 1443  
    # Input: 486  
    # Output: 1444  
    # Input: 487  
    # Output: 1449  
    # Input: 488  
    # Output: 1450  
    # Input: 489  
    # Output: 1455  
    # Input: 490  
    # Output: 1456  
    # Input: 491  
    # Output: 1461  
    # Input: 492  
    # Output: 1462  
    # Input: 493  
    # Output: 1467  
    # Input: 494  
    # Output: 1468  
    # Input: 495  
    # Output: 1473  
    # Input: 496  
    # Output: 1474  
    # Input: 497  
    # Output: 1479  
    # Input: 498  
    # Output: 1480  
    # Input: 499  
    # Output: 1485  
    # Input: 500  
    # Output: 1486  
    # Input: 501  
    # Output: 1491  
    # Input: 502  
    # Output: 1492  
    # Input: 503  
    # Output: 1497  
    # Input: 504  
    # Output: 1498  
    # Input: 505  
    # Output: 1503  
    # Input: 506  
    # Output: 1504  
    # Input: 507  
    # Output: 1509  
    # Input: 508  
    # Output: 1510  
    # Input: 509  
    # Output: 1515  
    # Input: 510  
    # Output: 1516  
    # Input: 511  
    # Output: 1521  
    # Input: 512  
    # Output: 1522  
    # Input: 513  
    # Output: 1527  
    # Input: 514  
    # Output: 1528  
    # Input: 515  
    # Output: 1533  
    # Input: 516  
    # Output: 1534  
    # Input: 517  
    # Output: 1539  
    # Input: 518  
    # Output: 1540  
    # Input: 519  
    # Output: 1545  
    # Input: 520  
    # Output: 1546  
    # Input: 521  
    # Output: 1551  
    # Input: 522  
    # Output: 1552  
    # Input: 523  
    # Output: 1557  
    # Input: 524  
    # Output: 1558  
    # Input: 525  
    # Output: 1563  
    # Input: 526  
    # Output: 1564  
    # Input: 527  
    # Output: 1569  
    # Input: 528  
    # Output: 1570  
    # Input: 529  
    # Output: 1575  
    # Input: 530  
    # Output: 1576  
    # Input: 531  
    # Output: 1581  
    # Input: 532  
    # Output: 1582  
    # Input: 533  
    # Output: 1587  
    # Input: 534  
    # Output: 1588  
    # Input: 535  
    # Output: 1593  
    # Input: 536  
    # Output: 1594  
    # Input: 537  
    # Output: 1599  
    # Input: 538  
    # Output: 1600  
    # Input: 539  
    # Output: 1605  
    # Input: 540  
    # Output: 1606  
    # Input: 541  
    # Output: 1611  
    # Input: 542  
    # Output: 1612  
    # Input: 543  
    # Output: 1617  
    # Input: 544  
    # Output: 1618  
    # Input: 545  
    # Output: 1623  
    # Input: 546  
    # Output: 1624  
    # Input: 547  
    # Output: 1629  
    # Input: 548  
    # Output: 1630  
    # Input: 549  
    # Output: 1635  
    # Input: 550  
    # Output: 1636  
    # Input: 551  
    # Output: 1641  
    # Input: 552  
    # Output:
```

10 minutes 3

10 minutes

Marked out of 100

Flag question

The number of goals achieved by two football teams in matches in a league is given in the form of two lists. Consider:

- Football team A has played three matches, and has scored { 1, 2, 3 } goals in each match respectively.
- Football team B has played two matches, and has scored { 2, 4 } goals in each match respectively.
- Your task is to compute, for each match of team B, the total number of matches of team A, where team A has scored less than or equal to the number of goals scored by team B in that match.
- In the above case:
- For 2 goals scored by team B in its first match, team A has 2 matches with scores 1 and 2.
- For 4 goals scored by team B in its second match, team A has 3 matches with scores 1, 2 and 3.

Hence, the answer: {2, 3}.

Complete the code in the editor below. The program must return an array of `m` positive integers, one for each `matches[i]` representing the total number of elements `nums[j]` satisfying `nums[j] <= matches[i]` where `0 <= j < n` and `0 <= i < m`, in the given order.

It has the following:

`nums[nums[0]..nums[n-1]]`: first array of positive integers
`matches[matches[0]..matches[m-1]]`: second array of positive integers

Constraints:

- $2 \leq n, m \leq 10^5$
- $1 \leq \text{nums}[i] \leq 10^5$ where $0 \leq i < n$
- $1 \leq \text{matches}[i] \leq 10^6$ where $0 \leq i < m$

Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function

The first line contains an integer `n`, the number of elements in `nums`.

The next `n` lines each contain an integer describing `nums[i]` where $0 \leq i < n$.

The next line contains an integer `m`, the number of elements in `matches`.

The next `m` lines each contain an integer describing `matches[i]` where $0 \leq i < m$.

Sample Case 0

Sample Input 0

```
4
1
1
4
2
4
2
3
5
```

Sample Output 0

```
2
4
```

Explanation 0

We are given `n = 4`, `nums = [1, 4, 2, 4]`, `m = 2`, and `matches = [3, 5]`.

1. For `matches[0] = 3`, we have 2 elements in `nums` (`nums[0] = 1` and `nums[2] = 2`) that are $\leq \text{matches}[0]$.

2. For `matches[1] = 5`, we have 4 elements in `nums` (`nums[0] = 1`, `nums[1] = 4`, `nums[2] = 2`, and `nums[3] = 4`) that are $\leq \text{matches}[1]$.

Thus, the function returns the array `[2, 4]` as the answer.

Activate Windows
Go to Settings to activate Windows.

Sample Output :

1
0
1
4

Explanation :

We are given, $n = 5$, $nums = [2, 10, 1, 4, 6]$, $m = 4$, and $nums = [2, 1, 7, 8]$.

- For $nums[0] = 2$, we have 1 element in $nums$ ($nums[0] = 2$) that is $< nums[0]$.
- For $nums[1] = 10$, there are 0 elements in $nums$ that are $< nums[1]$.
- For $nums[2] = 7$, we have 3 elements in $nums$ ($nums[0] = 2$, $nums[2] = 5$, and $nums[3] = 6$) that are $< nums[2]$.
- For $nums[3] = 8$, we have 4 elements in $nums$ ($nums[0] = 2$, $nums[2] = 5$, $nums[3] = 4$, and $nums[4] = 6$) that are $< nums[3]$.

Thus, the function returns the array $[1, 0, 3, 4]$ as the answer.

Answer: [1,0,3,4] (3 % 3)

```

1  def countLessNums(nums):
2      n = len(nums)
3      res = [0] * n
4      for i in range(n):
5          for j in range(i+1, n):
6              if nums[i] < nums[j]:
7                  res[i] += 1
8          for j in range(i-1, -1, -1):
9              if nums[j] < nums[i]:
10                 res[i] += 1
11     return res
12
13 # Example usage:
14 nums = [2, 10, 1, 4, 6]
15 result = countLessNums(nums)
16 print(result)
17
18 # Output: [1, 0, 3, 4]
19
20

```

Input	Expected	Got
1	1	1
2	0	0
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8

Passed all tests: ✓

Activate Windows
Go to Settings to activate Windows.