

-Lab Week 5-

Problem 1: Sum of Even Numbers

Write a program that calculates the sum of all even numbers from 1 to N, where N is provided by the user. The program should then display the sum.

Here are the steps:

1. Ask the user to enter a positive integer N.
2. Use a for loop to iterate from 1 to N.
3. Within the loop, check if each number is even (i.e., divisible by 2). If it's even, add it to a running sum.
4. After the loop, display the sum of all even numbers from 1 to N.

For example, if the user enters 6, the program should calculate and display the sum of even numbers from 1 to 6, which is $2 + 4 + 6 = 12$.

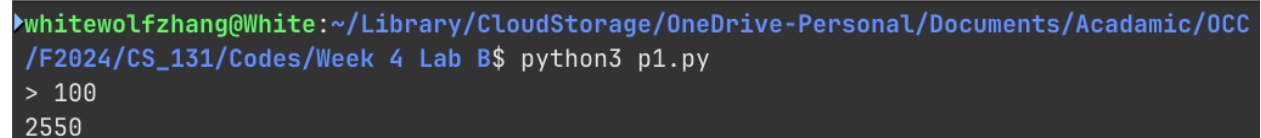
Paste your code below:

```
sum = 0
n = int(input("> "))

for x in range(1, n+1):
    if x % 2 == 0:
        sum += x

print(sum)
```

Paste the screenshot of your output below:



```
whitewolfzhang@White:~/Library/CloudStorage/OneDrive-Personal/Documents/Acadamic/OCC/F2024/CS_131/Codes/Week 4 Lab B$ python3 p1.py
> 100
2550
```

Problem2 : Multiplication Table

Write a program that generates a multiplication table for numbers from 1 to N, where N is provided by the user. The table should be displayed in the following format:

```
Multiplication Table for N:

  1  2  3  ...  N
1 1  2  3  ...  N
2 2  4  6  ...  2N
3 3  6  9  ...  3N
. . . . .
. . . . .
N N  2N 3N ...  N^2
```

Here are the steps:

1. Ask the user to enter a positive integer N.
2. Check if N is a positive integer. If not, display an error message and ask for input again until a valid positive integer is provided.
3. Use nested loops to generate and display the multiplication table up to N as shown above.

For example, if the user enters 4, the program should generate the following table:

```
Multiplication Table for 4:

  1  2  3  4
1 1  2  3  4
2 2  4  6  8
3 3  6  9 12
4 4  8 12 16
```

Paste your code below:

```
from math import log10

n = int(input("> "))
while n <= 0:
    print("Positive integer please")
    n = int(input("> "))

print(f"Multiplication table for {n}:")
print()

width = int(log10(n * n)) + 1
```

```
print(" " * int(log10(n)+1), end=' ')
print(" ".join([f"{i:>{width}}" for i in range(1, n+1)]))
```

```
for i in range(1, n+1):
    print(f"{i:>{int(log10(n)+1)}}", end=' ')
    print(" ".join([f"{j * i:>{width}}" for j in range(1, n+1)]))
```

Paste the screenshot of your output below:

```
whitewolfzhang@White: ~/Library/CloudStorage/OneDrive-Personal/Documents/Academic/JCC/F2024/CS_131/Codes/Week 4 Lab B
whitewolfzhang@White: ~/Library/CloudStorage/OneDrive-Personal/Documents/Academic/JCC/F2024/CS_131/Codes/Week 4 Lab B
Positive integer please
n
Positive integer please
n
Multiplication table for 50:

 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59
 2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100 102 104 106 108 110 112 114 116 118 120
 3  6  9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 66 69 72 75 78 81 84 87 90 93 96 99 102 105 108 111 114 117 120 123 126 129 132 135 138 141 144 147 150 153 156 159 162 165 168 171 174
 4  8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84 88 92 96 100 104 108 112 116 120 124 128 132 136 140 144 148 152 156 160 164 168 172 176 180 184 188 192 196 200 204 208 212 216 220 224 228
 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270
 6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96 102 108 114 120 126 132 138 144 150 156 162 168 174 180 186 192 198 204 210 216 222 228 234 240 246 252 258 264 270 276 282 288 294 300 306 312 318 324 330 336
 7 14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119 126 133 140 147 154 161 168 175 182 189 196 203 210 217 224 231 238 245 252 259 266 273 280 287 294 301 308 315 322 329 336 343 350 357 364 371 378 385 392 399 406 413
 8 16 24 32 40 48 56 64 72 80 88 96 104 112 120 128 136 144 152 160 168 176 184 192 200 208 216 224 232 240 248 256 264 272 280 288 296 304 312 320 328 336 344 352 360 368 376 384 392 400 408 416 424 432 440 448 456 464 472
 9 18 27 36 45 54 63 72 81 90 99 108 117 126 135 144 153 162 171 180 189 198 207 216 225 234 243 252 261 270 279 288 297 306 315 324 333 342 351 360 369 378 387 396 405 414 423 432 441 450 459 468 477 486 495 504 513 522 531
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350 360 370 380 390 400 410 420 430 440 450 460 470 480 490 500 510 520 530 540 550 560 570 580 590
11 22 33 44 55 66 77 88 99 110 121 132 143 154 165 176 187 198 209 220 231 242 253 264 275 286 297 308 319 330 341 352 363 374 385 396 407 418 429 440 451 462 473 484 495 506 517 528 539 550 561 572 583 594 605 616 627 638 649
12 24 36 48 60 72 84 96 108 120 132 144 156 168 180 192 204 216 228 240 252 264 276 288 300 312 324 336 348 360 372 384 396 408 420 432 444 456 468 480 492 504 516 528 540 552 564 576 588 600 612 624 636 648 660 672 684 696 708 720
13 26 39 52 65 78 91 104 117 130 143 156 169 182 195 208 221 234 247 260 273 286 299 312 325 338 351 364 377 390 403 416 429 442 455 468 481 494 507 520 533 546 559 572 585 598 611 624 637 650 663 676 689 702 715 728 741 754 767
14 28 42 56 70 84 98 112 126 140 154 168 182 196 210 224 238 252 266 280 294 308 322 336 350 364 378 392 406 420 434 448 462 476 490 504 518 532 546 560 574 588 602 616 630 644 658 672 686 700 714 728 742 756 770 784 798 812 826
15 30 45 60 75 90 105 120 135 150 165 180 195 210 225 240 255 270 285 300 315 330 345 360 375 390 405 420 435 450 465 480 495 510 525 540 555 570 585 600 615 630 645 660 675 690 705 720 735 750 765 780 795 810 825 840 855 870 885
16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320 336 352 368 384 396 412 428 444 460 476 492 508 524 540 556 572 588 604 620 636 652 668 684 700 716 732 748 764 780 796 812 828 844 860 876 892 908 924 940
17 34 51 68 85 102 119 136 153 170 187 204 221 238 255 272 289 306 323 340 357 374 391 408 425 442 459 476 493 510 527 544 561 578 595 612 629 646 663 680 697 714 731 748 765 782 799 816 833 850 867 884 901 918 935 952 969 986 1003
18 36 54 72 90 108 126 144 162 180 198 216 234 252 270 288 306 324 342 360 378 396 414 432 450 468 486 504 522 540 558 576 594 612 630 648 666 684 702 720 738 756 774 792 810 828 846 864 882 900 918 936 954 972 990 1008 1026 1044 1062
19 38 57 76 95 114 133 152 171 190 209 228 247 266 285 304 323 342 361 380 399 418 437 456 475 494 513 532 551 570 589 608 627 646 665 684 703 722 741 760 779 798 817 836 855 874 893 912 931 950 969 988 1007 1026 1045 1064 1083 1102
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380 400 420 440 460 480 500 520 540 560 580 600 620 640 660 680 700 720 740 760 780 800 820 840 860 880 900 920 940 960 980 1000 1020 1040 1060 1080 1100 1120 1140 1160 1180
21 42 63 84 105 126 147 168 189 210 231 252 273 294 315 336 357 378 399 420 441 462 483 504 525 546 567 588 609 630 651 672 693 714 735 756 777 798 819 840 861 882 903 924 945 966 987 1008 1029 1050 1071 1092 1113 1134 1155 1176 1197 1218 1239
22 44 66 88 110 132 154 176 198 220 242 264 286 308 330 352 374 396 418 440 462 484 506 528 550 572 594 616 638 660 682 704 726 748 770 792 814 836 858 880 902 924 946 968 990 1012 1034 1056 1078 1100 1122 1144 1166 1188 1210 1232 1254 1276
23 46 69 92 115 138 161 184 207 230 253 276 299 322 345 368 391 414 437 460 483 506 529 552 575 598 621 644 667 690 713 736 759 782 805 828 851 874 897 920 943 966 989 1012 1035 1058 1081 1104 1127 1150 1173 1196 1219 1242 1265 1288 1311 1334 1357
24 48 72 96 120 144 168 192 216 240 264 288 312 336 360 384 408 432 456 480 504 528 552 576 600 624 648 672 696 720 744 768 792 816 840 864 888 912 936 960 984 1008 1032 1056 1080 1104 1128 1152 1176 1200 1224 1248 1272 1300 1324 1348 1372
25 50 75 100 125 150 175 200 225 250 275 300 325 350 375 400 425 450 475 500 525 550 575 600 625 650 675 700 725 750 775 800 825 850 875 900 925 950 975 1000 1025 1050 1075 1100 1125 1150 1175 1200 1225 1250 1275 1300 1325 1350 1375
26 52 78 104 130 156 182 208 234 260 286 312 338 364 390 416 442 468 494 520 546 572 598 624 650 676 702 728 754 780 806 832 858 884 910 936 962 988 1014 1040 1066 1092 1118 1144 1170 1196 1222 1248 1274 1300 1326 1352 1378 1404 1430 1456 1482 1508 1534
27 54 81 108 135 162 189 216 243 270 297 324 351 378 405 432 459 486 513 540 567 594 621 648 675 702 729 756 783 810 837 864 891 918 945 972 999 1026 1053 1080 1107 1134 1161 1188 1215 1242 1269 1296 1323 1350 1377 1404 1431 1458 1485 1512 1539 1566 1593
28 56 84 112 140 168 196 224 252 280 308 336 364 392 420 448 476 504 532 560 588 616 644 672 700 728 756 784 812 840 868 896 924 952 980 1008 1036 1064 1092 1120 1148 1176 1204 1232 1260 1288 1316 1344 1372 1400 1428 1456 1484 1512 1540 1568 1596 1624 1652
29 58 87 116 145 174 203 232 261 290 319 348 377 406 435 464 493 522 551 580 609 638 667 696 725 754 783 812 841 870 899 928 957 986 1015 1044 1073 1102 1131 1160 1189 1218 1247 1276 1305 1334 1363 1392 1421 1450 1479 1508 1537 1566 1595 1625 1654 1683
30 60 90 120 150 180 210 240 270 300 330 360 390 420 450 480 510 540 570 600 630 660 690 720 750 780 810 840 870 900 930 960 990 1020 1050 1080 1110 1140 1170 1200 1230 1260 1290 1320 1350 1380 1410 1440 1470 1500 1530 1560 1590 1620 1650 1680 1710
31 62 93 124 155 186 217 248 279 310 341 372 403 434 465 496 527 558 589 620 651 682 713 744 775 806 837 868 899 930 961 992 1023 1054 1085 1116 1147 1178 1209 1240 1271 1302 1333 1364 1395 1426 1457 1488 1519 1550 1581 1612 1643 1674 1705 1736 1767
32 64 96 128 160 192 224 256 288 320 352 384 416 448 480 512 544 576 608 640 672 704 736 768 798 830 862 894 926 958 990 1022 1054 1086 1118 1150 1182 1214 1246 1278 1310 1342 1374 1406 1438 1470 1502 1534 1566 1598 1630 1662 1694 1726 1758 1790 1822 1854
33 66 99 132 164 196 228 260 292 324 356 388 420 452 484 516 548 580 612 644 676 708 740 772 804 836 868 898 930 962 994 1026 1058 1090 1122 1154 1186 1218 1250 1282 1314 1346 1378 1410 1442 1474 1506 1538 1570 1602 1634 1666 1698 1730 1762 1794 1826 1858
34 68 102 136 170 204 238 272 306 340 374 408 442 476 510 544 578 612 646 680 714 748 782 816 850 884 918 952 986 1020 1054 1088 1122 1156 1190 1224 1258 1292 1326 1360 1394 1428 1462 1496 1530 1564 1598 1632 1666 1698 1732 1766 1800 1834 1868 1902 1936
35 70 105 140 175 210 245 280 315 350 385 420 455 490 525 560 595 630 665 700 735 770 805 840 875 910 945 980 1015 1050 1085 1120 1155 1190 1225 1260 1295 1330 1365 1400 1435 1470 1505 1540 1575 1610 1645 1680 1715 1750 1785 1820 1855 1890 1925 1960 1995 2030 2065
36 72 108 144 180 216 252 288 324 360 396 432 468 504 540 576 612 648 684 720 756 792 828 864 900 936 972 1008 1044 1080 1116 1152 1188 1224 1260 1300 1336 1372 1408 1444 1480 1516 1552 1588 1624 1660 1696 1732 1768 1804 1840 1876 1912 1948 1984 2020 2056 2092
37 74 111 144 180 222 259 296 333 370 407 444 481 518 555 592 629 666 703 740 777 814 851 888 925 962 999 1036 1073 1110 1147 1184 1221 1258 1295 1332 1369 1406 1443 1480 1517 1554 1591 1628 1665 1702 1739 1776 1813 1850 1887 1924 1961 2000 2037 2074 2112 2150 2188
38 76 114 147 184 216 252 288 324 360 396 432 468 504 540 576 612 648 684 720 756 792 828 864 900 936 972 1008 1044 1080 1116 1152 1188 1224 1260 1300 1336 1372 1408 1444 1480 1516 1552 1588 1624 1660 1696 1732 1768 1804 1840 1876 1912 1948 1984 2020 2056 2092 2128 2164 2202
39 78 117 150 186 219 255 291 327 363 399 435 471 507 543 579 615 651 687 723 759 795 831 867 903 939 975 1011 1047 1083 1119 1155 1191 1227 1263 1300 1336 1372 1408 1444 1480 1516 1552 1588 1624 1660 1696 1732 1768 1804 1840 1876 1912 1948 1984 2020 2056 2092 2128 2164 2202
40 80 120 154 190 222 258 294 330 366 402 438 474 510 546 582 618 654 690 726 762 798 834 870 906 942 978 1014 1050 1086 1122 1158 1194 1230 1266 1302 1338 1374 1410 1446 1482 1518 1554 1590 1626 1662 1700 1736 1772 1808 1844 1880 1916 1952 1988 2024 2060 2096 2132 2168 2204
41 82 123 156 192 228 264 297 333 369 405 441 477 513 549 585 621 657 693 729 765 801 837 873 909 945 981 1017 1053 1089 1125 1161 1197 1233 1269 1305 1341 1377 1413 1449 1485 1521 1557 1593 1629 1665 1701 1737 1773 1809 1845 1881 1917 1953 1989 2025 2061 2097 2133 2169 2205
42 84 126 158 202 232 266 302 336 372 406 442 478 514 550 586 622 658 694 730 766 802 838 874 910 946 982 1018 1054 1090 1126 1162 1198 1234 1270 1306 1342 1378 1414 1450 1486 1522 1558 1594 1630 1666 1702 1738 1774 1810 1846 1882 1922 1958 1994 2030 2066 2102 2138 2174 2210
43 86 129 161 204 236 268 304 338 374 408 444 478 514 550 586 622 658 694 730 766 802 838 874 910 946 982 1018 1054 1090 1126 1162 1198 1234 1270 1306 1342 1378 1414 1450 1486 1522 1558 1594 1630 1666 1702 1738 1774 1810 1846 1882 1922 1958 1994 2030 2066 2102 2138 2174 2210
44 88 132 164 206 238 270 306 342 378 414 450 486 522 558 594 630 666 702 738 774 810 846 882 918 954 990 1026 1062 1098 1134 1170 1206 1242 1278 1314 1350 1386 1422 1458 1494 1530 1566 1602 1638 1674 1710 1746 1782 1818 1854 1890 1926 1962 2000 2036 2072 2108 2144
45 90 135 167 209 241 273 309 345 381 417 453 489 525 561 597 633 669 705 741 777 813 849 885 921 957 993 1029 1065 1101 1137 1173 1209 1245 1281 1317 1353 1389 1425 1461 1497 1533 1569 1605 1641 1677 1713 1749 1785 1821 1857 1893 1929 1965 2001 2037 2073 2109 2145
46 92 138 170 212 244 276 312 348 384 420 456 492 528 564 600 636 672 708 744 780 816 852 888 924 960 996 1032 1068 1104 1140 1176 1212 1248 1284 1320 1356 1392 1428 1464 1500 1536 1572 1608 1644 1680 1716 1752 1788 1824 1860 1896 1932 1968 2004 2040 2076 2112 2148 2184
47 94 141 173 215 247 279 315 351 387 423 459 495 531 567 603 639 675 711 747 783 819 855 891 927 963 999 1035 1071 1107 1143 1179 1215 1251 1287 1323 1359 1395 1431 1467 1503 1539 1575 1611 1647 1683 1719 1755 1791 1827 1863 1899 1935 1971 2007 2043 2079 2115 2151 2187
48 96 144 176 218 250 282 318 354 390 426 462 498 534 570 606 642 678 714 750 786 822 858 894 930 966 1002 1038 1074 1110 1146 1182 1218 1254 1290 1326 1362 1398 1434 1470 1506 1542 1578 1614 1650 1686 1722 1758 1794 1830 1866 1902 1938 1974 2010 2046 2082 2118 2154 2190
49 98 147 179 221 253 285 321 357 393 429 465 501 537 573 609 645 681 717 753 789 825 861 897 933 969 1005 1041 1077 1113 1149 1185 1221 1257 1293 1329 1365 1401 1437 1473 1509 154
```

Problem3 : Problem: Pattern Printing

Write a program that prints a pattern of stars (*) in the following format:



```
*  
**  
***  
****  
*****
```

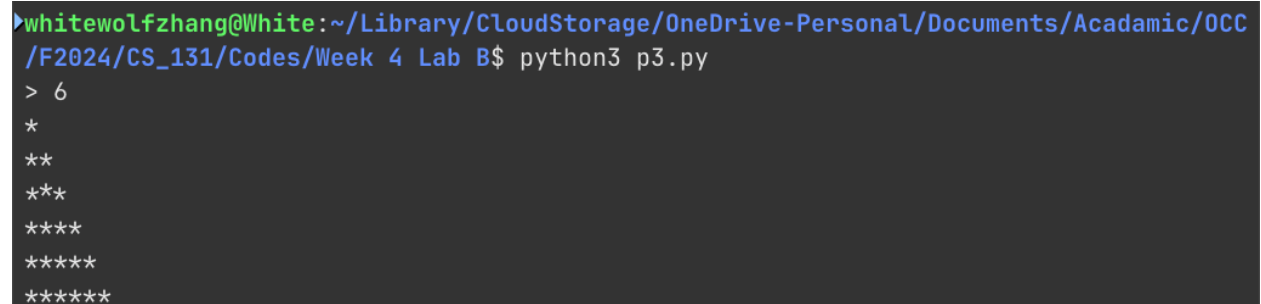
Here are the steps:

1. Ask the user to enter a positive integer N.
2. Use nested loops to print the pattern of stars. The outer loop should iterate from 1 to N, and the inner loop should print the corresponding number of stars on each line.

Paste your code below:

```
n = int(input("> "))  
  
for i in range(1, n+1):  
    for j in range(i):  
        print("*", end='')  
    print()
```

Paste the screenshot of your output below:



```
whitewolfzhang@White:~/Library/CloudStorage/OneDrive-Personal/Documents/Acadamic/OCC  
/F2024/CS_131/Codes/Week 4 Lab B$ python3 p3.py  
> 6  
*  
**  
***  
****  
*****  
*****
```