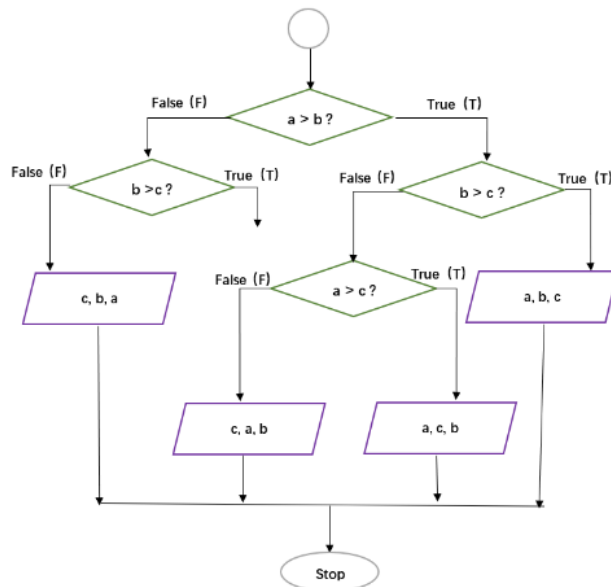


## 1. Flowchart

[10 points] Write a function `Print_values` with arguments `a`, `b`, and `c` to reflect the following flowchart. Here the purple parallelogram operator is to print values in the given order. Report your output with some random `a`, `b`, and `c` values.



### Description

When  $a > b > c$ , print (a, b, c);

When  $a > c > b$ , print (a, c, b);

When  $c > b > a$ , print (c, b, a);

When  $c > a > b$ , print (c, a, b);

When  $b > c > a$ , print (c, a, b);

When  $b > a > c$ , print (a, c, b).

### Result of Example

When  $a=6, b=4, c=3$ , immediately  $a > b > c$ , the result is 6, 4, 3, immediately (a, b, c);

When  $a=6, b=3, c=4$ , immediately  $a > c > b$ , the result is 6, 4, 3, immediately (a, c, b);

When  $a=3, b=4, c=6$ , immediately  $c > b > a$ , the result is 6, 4, 3, immediately (c, b, a);

When  $a=4, b=3, c=6$ , immediately  $c > a > b$ , the result is 6, 4, 3, immediately (c, a, b);

When  $a=3, b=6, c=4$ , immediately  $c > a > b$ , the result is 4, 3, 6, immediately (c, a, b);

When  $a=4, b=6, c=3$ , immediately  $c > a > b$ , the result is 4, 3, 6, immediately (a, c, b).

```
Console 1/A x
...:     if b>c:
...:         print(a,b,c)
...:     else:
...:         if a>c:
...:             print(a,c,b)
...:         else:
...:             print(c,a,b)
...:     else:
...:         if b>c:
...:             if a>c:
...:                 print(a,c,b)
...:             else:
...:                 print(c,a,b)
...:         else:
...:             print(c,b,a)
...:
...: # Example
...: # a=6, b=4, c=3
...: Print_values(6,4,3)
...: # a=6, b=3, c=4
...: Print_values(6,3,4)
...: # a=3, b=4, c=6
...: Print_values(3,4,6)
...: # a=4, b=3, c=6
...: Print_values(4,3,6)
...: # a=3, b=6, c=4
...: Print_values(3,6,4)
...: # a=4, b=6, c=3
...: Print_values(4,6,3)
6 4 3
6 4 3
6 4 3
6 4 3
4 3 6
4 3 6
In [82]:
```

---

## 2. Matrix multiplication

**2.1 [5 points]** Make two matrices `M1` ( 5 rows and 10 columns ) and `M2` ( 10 rows and 5 columns ); both are filled with random integers from 0 and 50 .

**2.2 [10 points]** Write a function `Matrix_multip` to do matrix multiplication, i.e., `M1 * M2` . Here you are **ONLY** allowed to use `for` loop, `*` operator, and `+` operator.

---

### Result of Example

M1=

[9, 47, 10, 23, 27, 29, 46, 19, 46, 12]

[19, 34, 45, 11, 46, 33, 39, 22, 0, 42]

[19, 31, 21, 31, 22, 23, 49, 45, 32, 11]

[50, 9, 49, 38, 33, 35, 6, 41, 47, 38]

[47, 16, 15, 28, 18, 44, 30, 50, 4, 46]

M2=

[24, 19, 41, 22, 19]

[20, 28, 8, 2, 12]

[2, 36, 34, 50, 4]

[20, 9, 41, 37, 27]

[46, 43, 9, 29, 11]

[31, 6, 45, 25, 10]

[18, 3, 19, 4, 0]

[42, 32, 21, 4, 37]

[24, 1, 28, 35, 30]

[28, 15, 22, 33, 24]

M1\*M2=

[6843, 4361, 6401, 5417, 4354]

[7387, 6659, 7058, 6932, 3904]

[7311, 5132, 7259, 5749, 5015]

[8863, 6884, 10345, 9893, 6832]

[8254, 5555, 8599, 6452, 5613]

```

Console 1/A x
In [82]: import random
...:
...: #2.1
...: #定义一个可以生成指定行数和列数的随机矩阵的函数
...: def generate_random_matrix(rows,columns):
...:     matrix=[]
...:     for i in range(rows):
...:         row=[]
...:         for j in range(columns):
...:             random_numbers = random.randint(0,50) #生产一个0~50之间随机数
...:             row.append(random_numbers)
...:         matrix.append(row)
...:     return matrix
...:
...: #生成一个5行10列的矩阵M1
...: M1=generate_random_matrix(5,10)
...: #生成一个10行5列的矩阵M2
...: M2=generate_random_matrix(10,5)
...:
...: #打印矩阵M1和M2
...: print("M1=")
...: for row in M1:
...:     print(row)
...: print("M2=")
...: for row in M2:
...:     print(row)
...:
...: #2.2
...: #定义矩阵乘法函数
...: def Matrix_multip(M1,M2):
...:     #先判断矩阵是否可以相乘
...:     if len(M1) != len(M2[0]):
...:         print("矩阵维度不匹配, 无法相乘!")
...:     else:
...:         multip_matrix = [[0 for _ in range(len(M1))] for _ in range(len(M1))]
...:         for i in range(len(M1)):
...:             for j in range(len(M1)):
...:                 for k in range(len(M2)):
...:                     multip_matrix[i][j] = multip_matrix[i][j]+M1[i][k] * M2[k][j]
...:         return multip_matrix
...:
...: #计算并打印M1*M2的结果
...: result=Matrix_multip(M1,M2)
...: print("M1*M2=")
...: for row in result:
...:     print(row)
M1=
[11, 45, 11, 29, 43, 18, 11, 28, 13, 28]
[15, 44, 50, 33, 17, 4, 8, 32, 4, 1]
[3, 45, 40, 31, 38, 26, 41, 4, 9, 41]
[38, 2, 43, 23, 25, 9, 2, 10, 50, 34]
[18, 5, 14, 11, 4, 39, 4, 4, 46, 37]
M2=
[36, 31, 50, 40, 27]
[17, 47, 41, 13, 4]
[3, 5, 14, 39, 8]
[37, 43, 20, 23, 34]
[34, 6, 43, 14, 0]
[47, 11, 12, 3, 3]
[2, 10, 45, 4, 39]
[47, 1, 34, 28, 18]
[5, 42, 43, 1, 40]
[0, 46, 15, 33, 2]
M1*M2=
[5978, 6186, 7620, 4542, 3114]
[4965, 4674, 6328, 5096, 3165]
[4969, 6933, 8104, 5226, 3826]
[4379, 6419, 7317, 5589, 4513]
[3577, 5467, 5010, 3152, 3251]

```

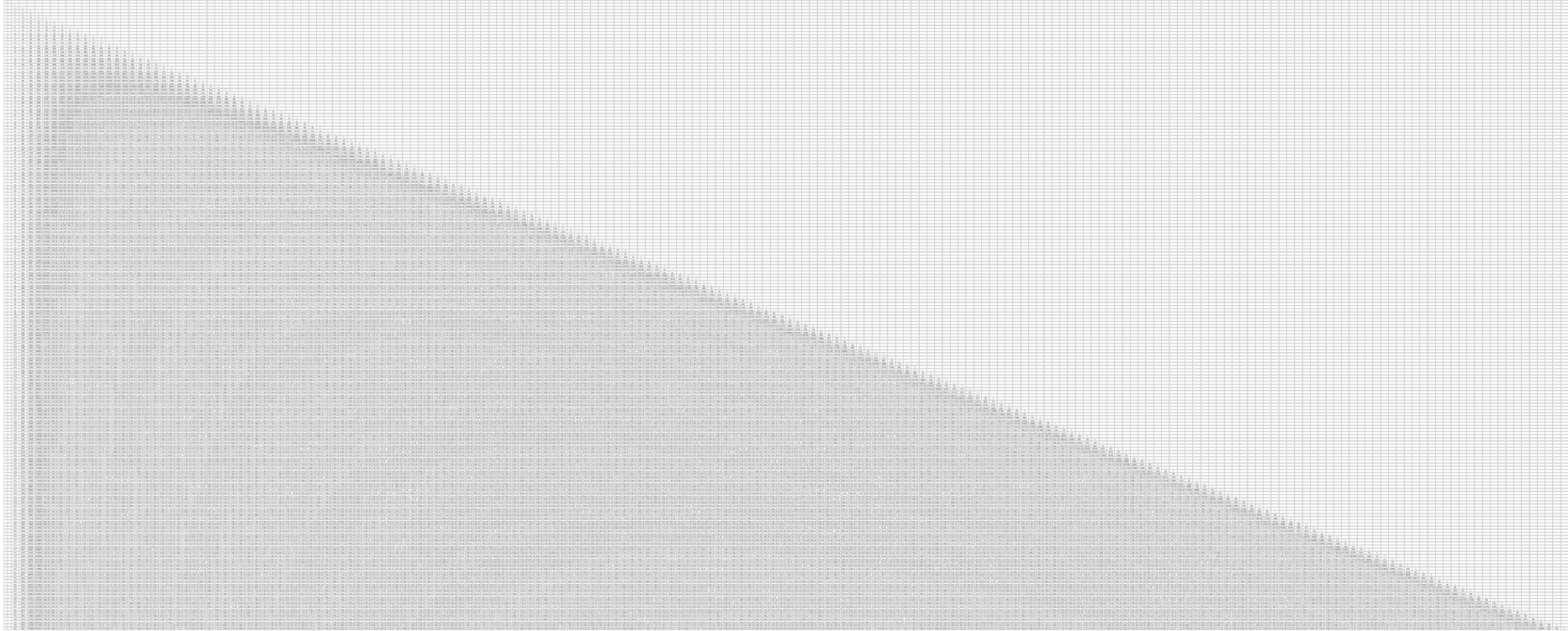
In [83]:

### 3. Pascal triangle

[20 points] One of the most interesting number patterns is [Pascal's triangle](#) (named after Blaise Pascal). Write a function `Pascal_triangle` with an argument `k` to print the  $k^{\text{th}}$  line of the Pascal triangle. Report `Pascal_triangle(100)` and `Pascal_triangle(200)`.

`Pascal_triangle(100)`

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 105 84 36 9 1
1 10 45 120 165 120 45 10 1
1 11 55 165 220 165 55 11 1
1 12 66 220 330 220 66 12 1
1 13 78 286 429 286 78 13 1
1 14 91 364 637 364 91 14 1
1 15 105 462 900 462 105 15 1
1 16 120 560 1287 560 120 16 1
1 17 136 680 1716 680 136 17 1
1 18 153 816 2431 816 153 18 1
1 19 171 969 3060 969 171 19 1
1 20 190 1140 3770 1140 190 20 1
1 21 210 1330 4641 1330 210 21 1
1 22 231 1540 5610 1540 231 22 1
1 23 253 1771 6720 1771 253 23 1
1 24 276 2024 8008 2024 276 24 1
1 25 300 2300 9690 2300 300 25 1
1 26 325 2600 11700 2600 325 26 1
1 27 351 2926 14186 2926 351 27 1
1 28 378 3280 17123 3280 378 28 1
1 29 406 3675 20625 3675 406 29 1
1 30 435 4105 24660 4105 435 30 1
1 31 465 4572 29371 4572 465 31 1
1 32 496 5076 34738 5076 496 32 1
1 33 528 5618 40847 5618 528 33 1
1 34 561 6200 47765 6200 561 34 1
1 35 595 6823 55565 6823 595 35 1
1 36 630 7488 64364 7488 630 36 1
1 37 666 8196 74297 8196 666 37 1
1 38 703 8948 85418 8948 703 38 1
1 39 741 9746 97742 9746 741 39 1
1 40 780 10590 111275 10590 780 40 1
1 41 820 11480 126104 11480 820 41 1
1 42 861 12416 142343 12416 861 42 1
1 43 903 13400 160021 13400 903 43 1
1 44 946 14433 179278 14433 946 44 1
1 45 990 15516 200145 15516 990 45 1
1 46 1035 16650 222765 16650 1035 46 1
1 47 1081 17835 247185 17835 1081 47 1
1 48 1128 19072 273464 19072 1128 48 1
1 49 1176 20363 301661 20363 1176 49 1
1 50 1225 21710 331735 21710 1225 50 1
1 51 1275 23115 363745 23115 1275 51 1
1 52 1326 24580 397751 24580 1326 52 1
1 53 1378 26107 433815 26107 1378 53 1
1 54 1431 27698 471998 27698 1431 54 1
1 55 1485 29355 512351 29355 1485 55 1
1 56 1540 31080 554936 31080 1540 56 1
1 57 1596 32875 600005 32875 1596 57 1
1 58 1653 34742 647700 34742 1653 58 1
1 59 1711 36683 698083 36683 1711 59 1
1 60 1770 38699 751206 38699 1770 60 1
1 61 1830 40792 807111 40792 1830 61 1
1 62 1891 42964 865850 42964 1891 62 1
1 63 1953 45217 927475 45217 1953 63 1
1 64 2016 47553 992038 47553 2016 64 1
1 65 2080 49975 1059681 49975 2080 65 1
1 66 2145 52485 1130456 52485 2145 66 1
1 67 2211 55085 1204405 55085 2211 67 1
1 68 2278 57778 1281581 57778 2278 68 1
1 69 2346 60567 1362036 60567 2346 69 1
1 70 2415 63455 1445923 63455 2415 70 1
1 71 2485 66445 1533305 66445 2485 71 1
1 72 2556 69540 1624334 69540 2556 72 1
1 73 2628 72743 1718163 72743 2628 73 1
1 74 2701 76058 1815945 76058 2701 74 1
1 75 2775 79488 1917833 79488 2775 75 1
1 76 2850 83036 2023980 83036 2850 76 1
1 77 2926 86705 2134540 86705 2926 77 1
1 78 3003 90498 2249665 90498 3003 78 1
1 79 3081 94418 2369500 94418 3081 79 1
1 80 3160 98468 2494299 98468 3160 80 1
1 81 3240 102651 2624105 102651 3240 81 1
1 82 3321 106970 2759061 106970 3321 82 1
1 83 3403 111428 2899310 111428 3403 83 1
1 84 3486 116028 3044905 116028 3486 84 1
1 85 3570 120774 3196000 120774 3570 85 1
1 86 3655 125669 3352749 125669 3655 86 1
1 87 3741 130717 3515305 130717 3741 87 1
1 88 3828 135921 3683821 135921 3828 88 1
1 89 3916 141285 3858450 141285 3916 89 1
1 90 4005 146812 4039345 146812 4005 90 1
1 91 4095 152506 4225760 152506 4095 91 1
1 92 4186 158370 4417848 158370 4186 92 1
1 93 4278 164408 4614753 164408 4278 93 1
1 94 4371 170624 4816630 170624 4371 94 1
1 95 4465 177021 5023723 177021 4465 95 1
1 96 4560 183603 5236185 183603 4560 96 1
1 97 4656 190374 5454160 190374 4656 97 1
1 98 4753 197338 5677803 197338 4753 98 1
1 99 4851 204498 5907360 204498 4851 99 1
1 100 4950 211858 6143085 211858 4950 100 1
1 101 5050 219421 6385233 219421 5050 101 1
1 102 5151 227191 6634060 227191 5151 102 1
1 103 5253 235171 6889821 235171 5253 103 1
1 104 5356 243365 7152761 243365 5356 104 1
1 105 5460 251777 7423125 251777 5460 105 1
1 106 5565 260411 7701168 260411 5565 106 1
1 107 5671 269271 7987145 269271 5671 107 1
1 108 5778 278361 8281310 278361 5778 108 1
1 109 5886 287685 8582917 287685 5886 109 1
1 110 5995 297247 8892310 297247 5995 110 1
1 111 6105 307051 9209843 307051 6105 111 1
1 112 6216 317101 9534860 317101 6216 112 1
1 113 6328 327401 9867605 327401 6328 113 1
1 114 6441 337955 10208423 337955 6441 114 1
1 115 6555 348767 10557650 348767 6555 115 1
1 116 6670 359841 10915621 359841 6670 116 1
1 117 6786 371181 11282671 371181 6786 117 1
1 118 6903 382791 11658135 382791 6903 118 1
1 119 7021 394675 12042358 394675 7021 119 1
1 120 7140 406837 12434685 406837 7140 120 1
1 121 7260 419281 12835461 419281 7260 121 1
1 122 7381 432011 13245031 432011 7381 122 1
1 123 7503 445031 13663740 445031 7503 123 1
1 124 7626 458345 14091943 458345 7626 124 1
1 125 7750 471957 14529995 471957 7750 125 1
1 126 7875 485871 14978250 485871 7875 126 1
1 127 8001 499991 15436963 499991 8001 127 1
1 128 8128 514321 15906488 514321 8128 128 1
1 129 8256 528865 16388179 528865 8256 129 1
1 130 8385 543627 16881290 543627 8385 130 1
1 131 8515 558611 17385175 558611 8515 131 1
1 132 8646 573831 17899990 573831 8646 132 1
1 133 8778 589291 18427089 589291 8778 133 1
1 134 8911 604995 18966716 604995 8911 134 1
1 135 9045 620947 19519115 620947 9045 135 1
1 136 9180 637151 20083530 637151 9180 136 1
1 137 9316 653611 20660205 653611 9316 137 1
1 138 9453 670331 21249485 670331 9453 138 1
1 139 9591 687315 21851615 687315 9591 139 1
1 140 9730 704567 22465940 704567 9730 140 1
1 141 9870 722091 23092815 722091 9870 141 1
1 142 10011 739891 23732585 739891 10011 142 1
1 143 10153 757971 24385595 757971 10153 143 1
1 144 10296 776335 25051190 776335 10296 144 1
1 145 10440 794987 25729725 794987 10440 145 1
1 146 10585 813931 26421545 813931 10585 146 1
1 147 10731 833171 27126005 833171 10731 147 1
1 148 10878 852711 27843460 852711 10878 148 1
1 149 11026 872557 28574165 872557 11026 149 1
1 150 11175 892713 29318475 892713 11175 150 1
1 151 11325 913183 30076745 913183 11325 151 1
1 152 11476 933971 30849240 933971 11476 152 1
1 153 11628 955091 31636325 955091 11628 153 1
1 154 11781 976547 32438355 976547 11781 154 1
1 155 11935 998343 33255685 998343 11935 155 1
1 156 12090 1020483 34088570 1020483 12090 156 1
1 157 12246 1042981 34937365 1042981 12246 157 1
1 158 12403 1065841 35802425 1065841 12403 158 1
1 159 12561 1089067 36684005 1089067 12561 159 1
1 160 12720 1112573 37582460 1112573 12720 160 1
1 161 12880 1136463 38498145 1136463 12880 161 1
1 162 13041 1160741 39431425 1160741 13041 162 1
1 163 13203 1185411 40382665 1185411 13203 163 1
1 164 13366 1210477 41352230 1210477 13366 164 1
1 165 13530 1235943 42340585 1235943 13530 165 1
1 166 13695 1261813 43348095 1261813 13695 166 1
1 167 13861 1288091 44376125 1288091 13861 167 1
1 168 14028 1314681 45425040 1314681 14028 168 1
1 169 14196 1341687 46495205 1341687 14196 169 1
1 170 14365 1369113 47586985 1369113 14365 170 1
1 171 14535 1396963 48699745 1396963 14535 171 1
1 172 14706 1425241 49833950 1425241 14706 172 1
1 173 14878 1453959 50989965 1453959 14878 173 1
1 174 15051 1483119 52168155 1483119 15051 174 1
1 175 15225 1512733 53368985 1512733 15225 175 1
1 176 15400 1542815 54592820 1542815 15400 176 1
1 177 15576 1573279 55839925 1573279 15576 177 1
1 178 15753 1604139 57110665 1604139 15753 178 1
1 179 15931 1635407 58405405 1635407 15931 179 1
1 180 16110 1667097 59724510 1667097 16110 180 1
1 181 16290 1699223 61068245 1699223 16290 181 1
1 182 16471 1731799 62436985 1731799 16471 182 1
1 183 16653 1764839 63831995 1764839 16653 183 1
1 184 16836 1798357 65253740 1798357 16836 184 1
1 185 17020 1832367 66702585 1832367 17020 185 1
1 186 17205 1866883 68179895 1866883 17205 186 1
1 187 17391 1901919 69685035 1901919 17391 187 1
1 188 17578 1937489 71218380 1937489 17578 188 1
1 189 17766 1973607 72779305 1973607 17766 189 1
1 190 17955 2010287 74368185 2010287 17955 190 1
1 191 18145 2047543 75985395 2047543 18145 191 1
1 192 18336 2085389 77631300 2085389 18336 192 1
1 193 18528 2123839 79306375 2123839 18528 193 1
1 194 18721 2162907 81011085 2162907 18721 194 1
1 195 18915 2202607 82745895 2202607 18915 195 1
1 196 19110 2242953 84511270 2242953 19110 196 1
1 197 19306 2283859 86307685 2283859 19306 197 1
1 198 19503 2325339 88135605 2325339 19503 198 1
1 199 19701 2367407 90005500 2367407 19701 199 1
1 200 19900 2410077 91907845 2410077 19900 200 1
1 201 20100 2453363 93843110 2453363 20100 201 1
1 202 20301 2497289 95811760 2497289 20301 202 1
1 203 20503 2541869 97814260 2541869 20503 203 1
1 204 20706 2587117 99850175 2587117 20706 204 1
1 205 20910 2633047 101919070 2633047 20910 205 1
1 206 21115 2679673 104022510 2679673 21115 206 1
1 207 21321 2727009 106161150 2727009 21321 207 1
1 208 21528 2775069 108335550 2775069 21528 208 1
1 209 21736 2823867 110546265 2823867 21736 209 1
1 210 21945 2873417 112792850 2873417 21945 210 1
1 211 22155 2923733 115075960 2923733 22155 211 1
1 212 22366 2974829 117396150 2974829 22366 212 1
1 213 22578 3026719 119744085 3026719 22578 213 1
1 214 22791 3079417 122129430 3079417 22791 214 1
1 215 23005 3132937 124552850 3132937 23005 215 1
1 216 23220 3187293 127004910 3187293 23220 216 1
1 217 23436 3242509 129495265 3242509 23436 217 1
1 218 23653 3298599 132024580 3298599 23653 218 1
1 219 23871 3355577 134593410 3355577 23871 219 1
1 220 24090 3413457 137192410 3413457 24090 220 1
1 221 24310 3472253 139822240 3472253 24310 221 1
1 222 24531 3532079 142483560 3532079 24531 222 1
1 223 24753 3592949 145176930 3592949 24753 223 1
1 224 24976 3654887 147902910 3654887 24976 224 1
1 225 25200 3717907 150661160 3717907 25200 225 1
1 226 25425 3782023 153453240 3782023 25425 226 1
1 227 25651 3847259 156279710 3847259 25651 227 1
1 228 25878 3913629 159141130 3913629 25878 228 1
1 229 26106 3981157 162038060 3981157 26106 229 1
1 230 26335 4049857 164971060 4049857 26335 230 1
1 231 26565 4119743 167940690 4119743 26565 231 1
1 232 26796 4190839 170937510 4190839 26796 232 1
1 233 27028 4263169 173962180 4263169 27028 233 1
1 234 27261 4336747 177015260 4336747 27261 234 1
1 235 27495 4411587 180097310 4411587 27495 235 1
1 236 27730 4487703 183208990 4487703 27730 236 1
1 237 27966 4565109 186350860 4565109 27966 237 1
1 238 28203 4643829 189523570 4643829 28203 238 1
1 239 28441 4723877 192727680 4723877 28441 239 1
1 240 28680 4805267 195963850 4805267 28680 240 1
1 241 28920 4888013 199232740 4888013 28920 241 1
1 242 29161 4972139 202535910 4972139 29161 242 1
1 243 29403 5057659 205873910 5057659 29403 243 1
1 244 29646 5144597 209247290 5144597 29646 244 1
1 245 29890 5232967 212647610 5232967 29890 245 1
1 246 30135 5322783 216095420 5322783 30135 246 1
1 247 30381 5414059 219581280 5414059 30381 247 1
1 248 30628 5506809 223105750 5506809 30628 248 1
1 249 30876 5601047 226659390 5601047 30876 249 1
1 250 31125 5696787 230252660 5696787 31125 250 1
1 251 31375 5794043 233886120 5794043 31375 251 1
1 252 31626 5892829 237559330 5892829 31626 252 1
1 253 31878 5993159 241272940 5993159 31878 253 1
1 254 32131 6095047 245027510 6095047 32131 254 1
1 255 32385 6198507 248823590 6198507 32385 255 1
1 256 32640 6303553 252670740 6303553 32640 256 1
1 257 32896 6410209 256559510 6410209 32896 257 1
1 258 33153 6518489 260499460 6518489 33153 258 1
1 259 33411 6628307 264491140 6628307 33411 259 1
1 260 33670 6739687 268535110 6739687 33670 260 1
1 261 33930 6852643 272631990 6852643 33930 261 1
1 262 34191 6967189 276782340 6967189 34191 262 1
1 263 34453 7083339 280986710 7083339 34453 263 1
1 264 34716 7201107 285245660 7201107 34716 264 1
1 265 34980 7320507 289559650 7320507 34980 265 1
1 266 35245 7441553 293929240 7441553 35245 266 1
1 267 35511 7564259 298354990 7564259 35511 267 1
1 268 35778 7688639 302827360 7688639 35778 268 1
1 269 36046 7814707 307346910 7814707 36046 269 1
1 270 36315 7942387 311914190 7942387 36315 270 1
1 271 36585 8071693 316538760 8071693 36585 271 1
1 272 36856 8202639 321221170 8202639 36856 272 1
1 273 37128 8334339 325961980 8334339 37128 273 1
1 274 37401 8467707 330761750 8467707 37401 274 1
1 275 37675 8602757 335621040 8602757 37675 275 1
1 276 37950 8739493 340541410 8739493 37950 276 1
1 277 38226 8877929 345523410 8877929 38226 277 1
1 278 38503 9018069 350567590 9018069 38503 278 1
1 279 38781 9159
```



Name	Type	Size	Value
triangle_10	list	10	[[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1], [1, 5, 10, 10, ...
triangle_100	list	100	[[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1], [1, 5, 10, 10, ...
triangle_200	list	200	[[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1], [1, 5, 10, 10, ...

Help

Variable Explorer

Plots

Files

Console 1/A ×

```
...: for row in triangle:
...:     formatted_row = ' '.join(map(str, row))
...:     print(formatted_row.center(max_width))
...:
...: #生成10阶的Pascal_triangle数组
...: triangle_10=Pascal_triangle(10)
...: #生成100阶的Pascal_triangle数组
...: triangle_100=Pascal_triangle(100)
...: #生成200阶的Pascal_triangle数组
...: triangle_200=Pascal_triangle(200)
...:
...: #打印10阶的Pascal_triangle数组
...: Print_pascals_triangle(triangle_10)
...:
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

## 4. Add or double

**[20 points]** If you start with 1 RMB and, with each move, you can either double your money or add another 1 RMB, what is the smallest number of moves you have to make to get to exactly  $x$  RMB? Here  $x$  is an integer randomly selected from 1 to 100. Write a function `Least_moves` to print your results. For example, `Least_moves(2)` should print 1, and `Least_moves(5)` should print 3.

### Description

To calculate the shortest number of steps from 1 to  $x$ , that is, calculate the shortest number of steps from  $x$  to 1. Therefore, define a function. When  $x$  is an odd number,  $x=x-1$ . When  $x$  is an even number,  $x=x/2$ . The two run alternately until  $x=1$ . The shortest number of steps is the number of steps for  $x$  to be calculated.

### Result of Example

`Least_moves(3)=1`

`Least_moves(5)=3`

`Least_moves(33)=6`

`Least_moves(56)=7`

`Least_moves(84)=8`

```
In [87]:
...: def Least_moves(x):
...:     moves = 0
...:     if x == 1:                                #如果x=1,moves=0
...:         return moves
...:     else:
...:         while x > 1:
...:             if x % 2 == 0:                    #如果x为偶数
...:                 x = x // 2                    # 除以2
...:             else:
...:                 x = x - 1                    #如果x为奇数，减1
...:                 moves += 1                    #每进行1次运算，步长加1
...:         return moves
...:
...: import random
...: x=random.randint(1,100)    #x为1~100的随机数
...: moves=Least_moves(x)       #计算到x值的最短步数
...: print(f"获得 {x} RMB的最少步数: {moves}")  #输出获得x元的最少步数为moves
获得 56 RMB的最少步数: 7
```



## 5. Dynamic programming

Insert `+` or `-` operation anywhere between the digits `123456789` in a way that the expression evaluates to an integer number. You may join digits together to form a bigger number. However, the digits must stay in the original order.

**5.1 [30 points]** Write a function `Find_expression`, which should be able to print every possible solution that makes the expression evaluate to a random integer from `1` to `100`. For example, `Find_expression(50)` should print lines include:

$$1 - 2 + 34 + 5 + 6 + 7 + 8 - 9 = 50$$

and

$$1 + 2 + 34 - 56 + 78 - 9 = 50$$

**5.2 [5 points]** Count the total number of suitable solutions for any integer  $i$  from `1` to `100`, assign the count to a list called `Total_solutions`. Plot the list `Total_solutions`, so which number(s) yields the maximum and minimum of `Total_solutions`?

### 5.1 Result of Example (Find\_expression(50))

打印出所有 `target=50` 的等式

$$1+2+3+4-56+7+89=50$$

$$1+2+3-4+56-7+8-9=50$$

$$1+2-3+4+56+7-8-9=50$$

$$1+2+34-5-6+7+8+9=50$$

$$1+2+34-56+78-9=50$$

$$1+2-34+5-6-7+89=50$$

$$1-2+3-45+6+78+9=50$$

$$1-2-3+4+56-7-8+9=50$$

$$1-2-3-4-5-6+78-9=50$$

$$1-2+34+5+6+7+8-9=50$$

$$1-2+34-5-67+89=50$$

$$1-2-34-5-6+7+89=50$$

$$1-23+4+5-6+78-9=50$$

$$1-23-4-5-6+78+9=50$$

$$12+3+4-56+78+9=50$$

$$12-3-4-5+67-8-9=50$$

$$12-3+45+6+7-8-9=50$$

```

打印出所有target=50的等式
1+2+3+4-56+7+89=50
1+2+3-4+56-7+8-9=50
1+2-3+4+56+7-8-9=50
1+2+34-5-6+7+8+9=50
1+2+34-56+78-9=50
1+2-34+5-6-7+89=50
1-2+3-45+6+78+9=50
1-2-3+4+56-7-8+9=50
1-2-3-4-5-6+78-9=50
1-2+34+5+6+7+8-9=50
1-2+34-5-6+7+89=50
1-2-34-5-6+7+89=50
1-23+4+5-6+78-9=50
1-23-4-5-6+78+9=50
12+3+4-56+78+9=50
12-3-4-5+67-8-9=50
12-3+45+6+7-8-9=50
Max Solutions: 26 for Target: 1
Min Solutions: 6 for Target: 88

```

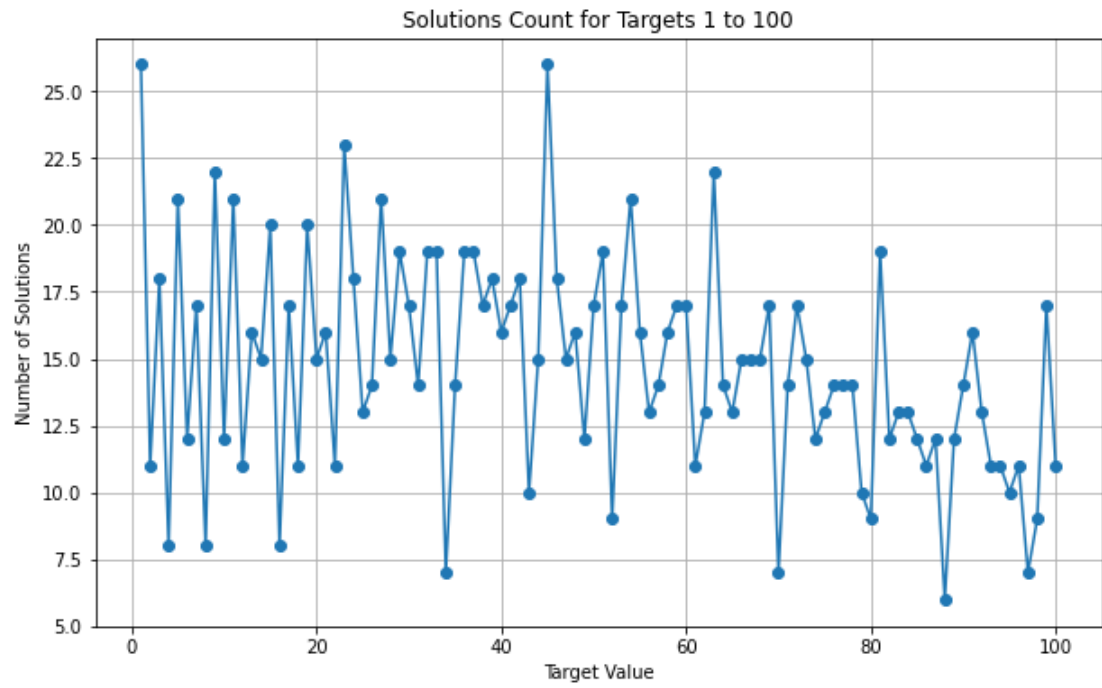
## 5.2

The maximum of Total\_solutions is 26 for target=1.

The minimum of Total\_solutions is 6 for target=88.

tar get	number of solutions	tar get	number of solutions	tar get	number of solutions	tar get	number of solutions	tar get	number of solutions
1	26	21	16	41	17	61	11	81	19
2	11	22	11	42	18	62	13	82	12
3	18	23	23	43	10	63	22	83	13
4	8	24	18	44	15	64	14	84	13
5	21	25	13	45	26	65	13	85	12
6	12	26	14	46	18	66	15	86	11
7	17	27	21	47	15	67	15	87	12
8	8	28	15	48	16	68	15	88	6
9	22	29	19	49	12	69	17	89	12
10	12	30	17	50	17	70	7	90	14
11	21	31	14	51	19	71	14	91	16
12	11	32	19	52	9	72	17	92	13
13	16	33	19	53	17	73	15	93	11
14	15	34	7	54	21	74	12	94	11
15	20	35	14	55	16	75	13	95	10
16	8	36	19	56	13	76	14	96	11

17	17	37	19	57	14	77	14	97	7
18	11	38	17	58	16	78	14	98	9
19	20	39	18	59	17	79	10	99	17
20	15	40	16	60	17	80	9	10	11
							0		



Name ▲	Type	Size	Value
expressions	list	17	[('1+2+3+4-56+7+89', 1), ('1+2+3-4+56-7+8-9', 1), ('1+2-3+4+56+7-8-9', ...
max_solutions	int	1	26
max_target	int	1	1
min_solutions	int	1	6
min_target	int	1	88
target	int	1	100
Total_solutions	list	100	[26, 11, 18, 8, 21, 12, 17, 8, 22, 12, ...]

Help Variable Explorer Plots Files



Console 1/A x



```

...: print('Min Solutions: ', min_solutions, ' for Target: ', min_target)
...:
...: plt.show()
打印出所有target=50的等式
1+2+3+4-56+7+89=50
1+2+3-4+56-7+8-9=50
1+2-3+4+56+7-8-9=50
1+2+3+4-5-6+7+8+9=50
1+2+3+4-56+78-9=50
1+2-3+4+5-6-7+89=50
1-2+3-45+6+78+9=50
1-2-3+4+56-7-8+9=50
1-2-3-4-5-6+78-9=50
1-2+3+4+5+6+7+8-9=50
1-2+3+4-5-6+7+89=50
1-2-3+4-5-6+7+89=50
1-23+4+5-6+78-9=50
1-23-4-5-6+78+9=50
12+3+4-56+78+9=50
12-3-4-5+67-8-9=50
12-3+4+5+6+7-8-9=50
Max Solutions: 26 for Target: 1
Min Solutions: 6 for Target: 88

```