

# [Python] \_\_init\_\_(??? ??) & ??? (?????? 25? 1? ??)

17. [] [] [] [] . [] [] [] [] [] [] [] .

```
class Node:
    def __init__(self, value):
        self.value = value
        self.children = []

def tree(li):
    nodes = [Node(i) for i in li]
    for i in range(1, len(li)):
        nodes[(i - 1) // 2].children.append(nodes[i])
    return nodes[0]

def calc(node, level=0):
    if node is None:
        return 0
    return (node.value if level % 2 == 1 else 0) + sum(calc(n, level + 1) for n in
node.children)

li = [3, 5, 8, 12, 15, 18, 21]

root = tree(li)

print(calc(root))
```

- \_\_init\_\_: [] [] [] [] [] . Node [] [] [] []
- children: [] [] [] []
- tree(): [] [] [] [] []
- calc(): [] [] [] [] [] [] []

**13**

이 i 1인 경우 ?

li = [3, 5, 8, 12, 15, 18, 21] 이고 i = 0일 때 nodes[(0 - 1) // 2] = nodes[-1]은 배열의 마지막 원소인 21을 반환한다. nodes[0]은 배열의 첫 번째 원소인 3을 반환한다. (root)는 배열의 첫 번째 원소인 3을 반환한다. , nodes[i]는 배열의 i 번째 원소를 반환한다.

18, 21은 2의 거듭제곱이므로, 이들을 2의 거듭제곱으로 올린 후 2를 빼면 된다.

“이 코드는 배열을 사용하여 이진 트리를 표현한다.”

- 이진 트리를 배열로 표현할 때, 노드의 인덱스와 레벨을 계산하는 방법을 알아야 한다.
- 이진 트리의 루트 노드는 인덱스 0에 위치한다.

노드 인덱스	노드 레벨
nodes(index)	노드 인덱스 (0부터 시작)
nodes(level)	노드 레벨 (0부터 시작)

- index 3 (값 12)의 부모 노드는 index 1 (값 5)이다.
- index 4 (값 15)의 부모 노드는 index 1 (값 5)이다.
- index 5 (값 18)의 부모 노드는 index 2 (값 8)이다.
- index 6 (값 21)의 부모 노드는 index 2 (값 8)이다.

if node is None: 이 조건문을 어떻게 처리할 것인가?

- node가 None이면, "노드가 존재하지 않음"이라는 메시지를 출력하고, 해당 노드의 자식 노드를 생성하지 않는다.
- if node is None: → "노드가 존재하지 않음"이라는 메시지를 출력하고, 해당 노드의 자식 노드를 생성하지 않는다.
- 이진 트리를 배열로 표현할 때, 노드의 인덱스와 레벨을 계산하는 방법을 알아야 한다.

node가 None이면, "노드가 존재하지 않음"이라는 메시지를 출력하고, 해당 노드의 자식 노드를 생성하지 않는다.

```
3
 / \
5   8
 / \
12  15
```

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 99 102 105 108 111 114 117 120 123 126 129 132 135 138 141 144 147 150 153 156 159 162 165 168 171 174 177 180 183 186 189 192 195 198 201 204 207 210 213 216 219 222 225 228 231 234 237 240 243 246 249 252 255 258 261 264 267 270 273 276 279 282 285 288 291 294 297 300 303 306 309 312 315 318 321 324 327 330 333 336 339 342 345 348 351 354 357 360 363 366 369 372 375 378 381 384 387 390 393 396 399 402 405 408 411 414 417 420 423 426 429 432 435 438 441 444 447 450 453 456 459 462 465 468 471 474 477 480 483 486 489 492 495 498 501 504 507 510 513 516 519 522 525 528 531 534 537 540 543 546 549 552 555 558 561 564 567 570 573 576 579 582 585 588 591 594 597 600 603 606 609 612 615 618 621 624 627 630 633 636 639 642 645 648 651 654 657 660 663 666 669 672 675 678 681 684 687 690 693 696 699 702 705 708 711 714 717 720 723 726 729 732 735 738 741 744 747 750 753 756 759 762 765 768 771 774 777 780 783 786 789 792 795 798 801 804 807 810 813 816 819 822 825 828 831 834 837 840 843 846 849 852 855 858 861 864 867 870 873 876 879 882 885 888 891 894 897 900 903 906 909 912 915 918 921 924 927 930 933 936 939 942 945 948 951 954 957 960 963 966 969 972 975 978 981 984 987 990 993 996 999 1002 1005 1008 1011 1014 1017 1020 1023 1026 1029 1032 1035 1038 1041 1044 1047 1050 1053 1056 1059 1062 1065 1068 1071 1074 1077 1080 1083 1086 1089 1092 1095 1098 1101 1104 1107 1110 1113 1116 1119 1122 1125 1128 1131 1134 1137 1140 1143 1146 1149 1152 1155 1158 1161 1164 1167 1170 1173 1176 1179 1182 1185 1188 1191 1194 1197 1200 1203 1206 1209 1212 1215 1218 1221 1224 1227 1230 1233 1236 1239 1242 1245 1248 1251 1254 1257 1260 1263 1266 1269 1272 1275 1278 1281 1284 1287 1290 1293 1296 1299 1302 1305 1308 1311 1314 1317 1320 1323 1326 1329 1332 1335 1338 1341 1344 1347 1350 1353 1356 1359 1362 1365 1368 1371 1374 1377 1380 1383 1386 1389 1392 1395 1398 1401 1404 1407 1410 1413 1416 1419 1422 1425 1428 1431 1434 1437 1440 1443 1446 1449 1452 1455 1458 1461 1464 1467 1470 1473 1476 1479 1482 1485 1488 1491 1494 1497 1500 1503 1506 1509 1512 1515 1518 1521 1524 1527 1530 1533 1536 1539 1542 1545 1548 1551 1554 1557 1560 1563 1566 1569 1572 1575 1578 1581 1584 1587 1590 1593 1596 1599 1602 1605 1608 1611 1614 1617 1620 1623 1626 1629 1632 1635 1638 1641 1644 1647 1650 1653 1656 1659 1662 1665 1668 1671 1674 1677 1680 1683 1686 1689 1692 1695 1698 1701 1704 1707 1710 1713 1716 1719 1722 1725 1728 1731 1734 1737 1740 1743 1746 1749 1752 1755 1758 1761 1764 1767 1770 1773 1776 1779 1782 1785 1788 1791 1794 1797 1800 1803 1806 1809 1812 1815 1818 1821 1824 1827 1830 1833 1836 1839 1842 1845 1848 1851 1854 1857 1860 1863 1866 1869 1872 1875 1878 1881 1884 1887 1890 1893 1896 1899 1902 1905 1908 1911 1914 1917 1920 1923 1926 1929 1932 1935 1938 1941 1944 1947 1950 1953 1956 1959 1962 1965 1968 1971 1974 1977 1980 1983 1986 1989 1992 1995 1998 2001 2004 2007 2010 2013 2016 2019 2022 2025 2028 2031 2034 2037 2040 2043 2046 2049 2052 2055 2058 2061 2064 2067 2070 2073 2076 2079 2082 2085 2088 2091 2094 2097 2100 2103 2106 2109 2112 2115 2118 2121 2124 2127 2130 2133 2136 2139 2142 2145 2148 2151 2154 2157 2160 2163 2166 2169 2172 2175 2178 2181 2184 2187 2190 2193 2196 2199 2202 2205 2208 2211 2214 2217 2220 2223 2226 2229 2232 2235 2238 2241 2244 2247 2250 2253 2256 2259 2262 2265 2268 2271 2274 2277 2280 2283 2286 2289 2292 2295 2298 2301 2304 2307 2310 2313 2316 2319 2322 2325 2328 2331 2334 2337 2340 2343 2346 2349 2352 2355 2358 2361 2364 2367 2370 2373 2376 2379 2382 2385 2388 2391 2394 2397 2400 2403 2406 2409 2412 2415 2418 2421 2424 2427 2430 2433 2436 2439 2442 2445 2448 2451 2454 2457 2460 2463 2466 2469 2472 2475 2478 2481 2484 2487 2490 2493 2496 2499 2502 2505 2508 2511 2514 2517 2520 2523 2526 2529 2532 2535 2538 2541 2544 2547 2550 2553 2556 2559 2562 2565 2568 2571 2574 2577 2580 2583 2586 2589 2592 2595 2598 2601 2604 2607 2610 2613 2616 2619 2622 2625 2628 2631 2634 2637 2640 2643 2646 2649 2652 2655 2658 2661 2664 2667 2670 2673 2676 2679 2682 2685 2688 2691 2694 2697 2700 2703 2706 2709 2712 2715 2718 2721 2724 2727 2730 2733 2736 2739 2742 2745 2748 2751 2754 2757 2760 2763 2766 2769 2772 2775 2778 2781 2784 2787 2790 2793 2796 2799 2802 2805 2808 2811 2814 2817 2820 2823 2826 2829 2832 2835 2838 2841 2844 2847 2850 2853 2856 2859 2862 2865 2868 2871 2874 2877 2880 2883 2886 2889 2892 2895 2898 2901 2904 2907 2910 2913 2916 2919 2922 2925 2928 2931 2934 2937 2940 2943 2946 2949 2952 2955 2958 2961 2964 2967 2970 2973 2976 2979 2982 2985 2988 2991 2994 2997 3000 3003 3006 3009 3012 3015 3018 3021 3024 3027 3030 3033 3036 3039 3042 3045 3048 3051 3054 3057 3060 3063 3066 3069 3072 3075 3078 3081 3084 3087 3090 3093 3096 3099 3102 3105 3108 3111 3114 3117 3120 3123 3126 3129 3132 3135 3138 3141 3144 3147 3150 3153 3156 3159 3162 3165 3168 3171 3174 3177 3180 3183 3186 3189 3192 3195 3198 3201 3204 3207 3210 3213 3216 3219 3222 3225 3228 3231 3234 3237 3240 3243 3246 3249 3252 3255 3258 3261 3264 3267 3270 3273 3276 3279 3282 3285 3288 3291 3294 3297 3300 3303 3306 3309 3312 3315 3318 3321 3324 3327 3330 3333 3336 3339 3342 3345 3348 3351 3354 3357 3360 3363 3366 3369 3372 3375 3378 3381 3384 3387 3390 3393 3396 3399 3402 3405 3408 3411 3414 3417 3420 3423 3426 3429 3432 3435 3438 3441 3444 3447 3450 3453 3456 3459 3462 3465 3468 3471 3474 3477 3480 3483 3486 3489 3492 3495 3498 3501 3504 3507 3510 3513 3516 3519 3522 3525 3528 3531 3534 3537 3540 3543 3546 3549 3552 3555 3558 3561 3564 3567 3570 3573 3576 3579 3582 3585 3588 3591 3594 3597 3600 3603 3606 3609 3612 3615 3618 3621 3624 3627 3630 3633 3636 3639 3642 3645 3648 3651 3654 3657 3660 3663 3666 3669 3672 3675 3678 3681 3684 3687 3690 3693 3696 3699 3702 3705 3708 3711 3714 3717 3720 3723 3726 3729 3732 3735 3738 3741 3744 3747 3750 3753 3756 3759 3762 3765 3768 3771 3774 3777 3780 3783 3786 3789 3792 3795 3798 3801 3804 3807 3810 3813 3816 3819 3822 3825 3828 3831 3834 3837 3840 3843 3846 3849 3852 3855 3858 3861 3864 3867 3870 3873 3876 3879 3882 3885 3888 3891 3894 3897 3900 3903 3906 3909 3912 3915 3918 3921 3924 3927 3930 3933 3936 3939 3942 3945 3948 3951 3954 3957 3960 3963 3966 3969 3972 3975 3978 3981 3984 3987 3990 3993 3996 3999 4002 4005 4008 4011 4014 4017 4020 4023 4026 4029 4032 4035 4038 4041 4044 4047 4050 4053 4056 4059 4062 4065 4068 4071 4074 4077 4080 4083 4086 4089 4092 4095 4098 4101 4104 4107 4110 4113 4116 4119 4122 4125 4128 4131 4134 4137 4140 4143 4146 4149 4152 4155 4158 4161 4164 4167 4170 4173 4176 4179 4182 4185 4188 4191 4194 4197 4200 4203 4206 4209 4212 4215 4218 4221 4224 4227 4230 4233 4236 4239 4242 4245 4248 4251 4254 4257 4260 4263 4266 4269 4272 4275 4278 4281 4284 4287 4290 4293 4296 4299 4302 4305 4308 4311 4314 4317 4320 4323 4326 4329 4332 4335 4338 4341 4344 4347 4350 4353 4356 4359 4362 4365 4368 4371 4374 4377 4380 4383 4386 4389 4392 4395 4398 4401 4404 4407 4410 4413 4416 4419 4422 4425 4428 4431 4434 4437 4440 4443 4446 4449 4452 4455 4458 4461 4464 4467 4470 4473 4476 4479 4482 4485 4488 4491 4494 4497 4500 4503 4506 4509 4512 4515 4518 4521 4524 4527 4530 4533 4536 4539 4542 4545 4548 4551 4554 4557 4560 4563 4566 4569 4572 4575 4578 4581 4584 4587 4590 4593 4596 4599 4602 4605 4608 4611 4614 4617 4620 4623 4626 4629 4632 4635 4638 4641 4644 4647 4650 4653 4656 4659 4662 4665 4668 4671 4674 4677 4680 4683 4686 4689 4692 4695 4698 4701 4704 4707 4710 4713 4716 4719 4722 4725 4728 4731 4734 4737 4740 4743 4746 4749 4752 4755 4758 4761 4764 4767 4770 4773 4776 4779 4782 4785 4788 4791 4794 4797 4800 4803 4806 4809 4812 4815 4818 4821 4824 4827 4830 4833 4836 4839 4842 4845 4848 4851 4854 4857 4860 4863 4866 4869 4872 4875 4878 4881 4884 4887 4890 4893 4896 4899 4902 4905 4908 4911 4914 4917 4920 4923 4926 4929 4932 4935 4938 4941 4944 4947 4950 4953 4956 4959 4962 4965 4968 4971 4974 4977 4980 4983 4986 4989 4992 4995 4998 5001 5004 5007 5010 5013 5016 5019 5022 5025 5028 5031 5034 5037 5040 5043 5046 5049 5052 5055 5058 5061 5064 5067 5070 5073 5076 5079 5082 5085 5088 5091 5094 5097 5100 5103 5106 5109 5112 5115 5118 5121 5124 5127 5130 5133 5136 5139 5142 5145 5148 5151 5154 5157 5160 5163 5166 5169 5172 5175 5178 5181 5184 5187 5190 5193 5196 5199 5202 5205 5208 5211 5214 5217 5220 5223 5226 5229 5232 5235 5238 5241 5244 5247 5250 5253 5256 5259 5262 5265 5268 5271 5274 5277 5280 5283 5286 5289 5292 5295 5298 5301 5304 5307 5310 5313 5316 5319 5322 5325 5328 5331 5334 5337 5340 5343 5346 5349 5352 5355 5358 5361 5364 5367 5370 5373 5376 5379 5382 5385 5388 5391 5394 5397 5400 5403 5406 5409 5412 5415 5418 5421 5424 5427 5430 5433 5436 5439 5442 5445 5448 5451 5454 5457 5460 5463 5466 5469 5472 5475 5478 5481 5484 5487 5490 5493 5496 5499 5502 5505 5508 5511 5514 5517 5520 5523 5526 5529 5532 5535 5538 5541 5544 5547 5550 5553 5556 5559 5562 5565 5568 5571 5574 5577 5580 5583 5586 5589 5592 5595 5598 5601 5604 5607 5610 5613 5616 5619 5622 5625 5628 5631 5634 5637 5640 5643 5646 5649 5652 5655 5658 5661 5664 5667 5670 5673 5676 5679 5682 5685 5688 5691 5694 5697 5700 5703 5706 5709 5712 5715 5718 5721 5724 5727 5730 5733 5736 5739 5742 5745 5748 5751 5754 5757 5760 5763 5766 5769 5772 5775 5778 5781 5784 5787 5790 5793 5796 5799 5802 5805 5808 5811 5814 5817 5820 5823 5826 5829 5832 5835 5838 5841 5844 5847 5850 5853 5856 5859 5862 5865 5868 5871 5874 5877 5880 5883 5886 5889 5892 5895 5898 5901 5904 5907 5910 5913 5916 5919 5922 5925 5928 5931 5934 5937 5940 5943 5946 5949 5952 5955 5958 5961 5964 5967 5970 5973 5976 5979 5982 5985 5988 5991 5994 5997 6000 6003 6006 6009 6012 6015 6018 6021 6024 6027 6030 6033 6036 6039 6042 6045 6048 6051 6054 6057 6060 6063 6066 6069 6072 6075 6078 6081 6084 6087 6090 6093 6096 6099 6102 6105 6108 6111 6114 6117 6120 6123 6126 6129 6132 6135 6138 6141 6144 6147 6150 6153 6156 6159 6162 6165 6168 6171 6174 6177 6180 6183 6186 6189 6192 6195 6198 6201 6204 6207 6210 6213 6216 6219 6222 6225 6228 6231 6234 6237 6240 6243 6246 6249 6252 6255 6258 6261 6264 6267 6270 6273 6276 6279 6282 6285 6288 6291 6294 6297 6300 6303 6306 6309 6312 6315 6318 6321 6324 6327 6330 6333 6336 6339 6342 6345 6348 6351 6354 6357 6360 6363 6366 6369 6372 6375 6378 6381 6384 6387 6390 6393 6396 6399 6402 6405 6408 6411 6414 6417 6420 6423 6426 6429 6432 6435 6438 6441 6444 6447 6450 6453 6456 6459 6462 6465 6468 6471 6474 6477 6480 6483 6486 6489 6492 6495 6498 6501 6504 6507 6510 6513 6516 6519 6522 6525 6528 6531 6534 6537 6540 6543 6546 6549 6552 6555 6558 6561 6564 6567 6570 6573 6576 6579 6582 6585 6588 6591 6594 6597 6600 6603 6606 6609 6612 6615 6618 6621 6624 6627 6630 6633 6636 6639 6642 6645 6648 6651 6654 6657 6660 6663 6666 6669 6672 6675 6678 6681 6684 6687 6690 6693 6696 6699 6702 6705 6708 6711 6714 6717 6720 6723 6726 6729 6732 6735 6738 6741 6744 6747 6750 6753 6756 6759 6762 6765 6768 6771 6774 6777 6780 6783 6786 6789 6792 6795 6798 6801 6804 6807 6810 6813 6816 6819 6822 6825 6828 6831 6834 6837 6840 6843 6846 6849 6852 6855 6858 6861 6864 6867 6870 6873 6876 6879 6882 6885 6888 6891 6894 6897 6900 6903 6906 6909 6912 6915 6918 6921 6924 6927 6930 6933 6936 6939 6942 6945 6948 6951 6954 6957 6960 6963 6966 6969 6972 6975 6978 6981 6984 6987 6990 6993 6996 6999 7002 7005 7008 7011 7014 7017 7020 7023 7026 7029 7032 7035 7038 7041 7044 7047 7050 7053 7056 7059 7062 7065 7068 7071 7074 7077 7080 7083 7086 7089 7092 7095 7098 7101 7104 7107 7110 7113 7116 7119 7122 7125 7128 7131 7134 7137 7140 7143 7146 7149 7152 7155 7158 7161 7164 7167 7170 7173 7176 7179 7182 7185 7188 7191 7194 7197 7200 7203 7206 7209 7212 7215 7218 7221 7224 7227 7230 7233 7236 7239 7242 7245 7248 7251 7254 7257 7260 7263 7266 7269 7272 7275 7278 7281 7284 7287 7290 7293 7296 7299 7302 7305 7308 7311 7314 7317 7320 7323 7326 7329 7332 7335 7338 7341 7344 7347 7350 7353 7356 7359 7362 7365 7368 7371 7374 7377 7380 7383 7386 7389 7392 7395 7398 7401 7404 7407 7410 7413 7416 7419 7422 7425 7428 7431 7434 7437 7440 7443 7446 7449 7452 7455 7458 7461 7464 7467 7470 7473 7476 7479 7482 7485 7488 7491 7494 7497 7500 7503 7506 7509 7512 7515 7518 7521 7524 7527 7530 7533 7536 7539 7542 7545 7548 7551 7554 7557 7560 7563 7566 7569 7572 7575 7578 7581 7584 7587 7590 7593 7596 7599 7602 7605 7608 7611 7614 7617 7620 7623 7626 7629 7632 7635 7638 7641

index	:	value	
0	:	3	→ root
1	:	5	→ 3의 왼쪽 자식
2	:	8	→ 3의 오른쪽 자식
3	:	12	→ 5의 왼쪽 자식
4	:	15	→ 5의 오른쪽 자식
5	:	18	→ 8의 왼쪽 자식
6	:	21	→ 8의 오른쪽 자식

### 3. ?? ?? calc()

```
def calc(node, level=0):
    if node is None:
        return 0
    return (node.value if level % 2 == 1 else 0) + sum(calc(n, level + 1) for n in node.children)
```

- `node` : 현재 노드 (None일 수 있음)
- **level** : 현재 노드의 레벨 (1, 3, ...)일 때는 `node.value`를 더함

-       1 value    $5 + 8 = 13$

