

```
public class Cast{  
    public static void main(String args[]){  
  
        int i=10;  
        long l=1;  
  
        double d=3.4;  
  
        byte b;  
        b= (byte)i;  
        System.out.println(b); // h=10  
  
        i= (int) l;  
        System.out.println(i); // i=10  
  
        l=(long) d;  
        System.out.println(l); // d=3  
    }  
}
```

CODING P ...

AT

AB

C0

N

jela Tripa...

Anandita Bh...

CODING P



```
class Demo {
```

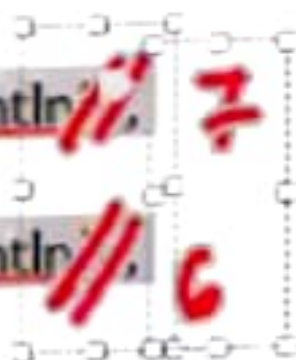
```
    public static void main(String args[]){
```

```
        int i=6;
```

```
        int j=i++;
```

```
        System.out.println(i);
```

```
        System.out.println(j);
```



```
class Demo {
```

```
public static void main(String args[]) {
```

```
    byte i;
```

```
    int j=9;
```

```
    i= j;
```

```
    System.out.println(i );
```

```
}}
```

```
class Demo {  
  
    public static void main(String args()) {  
  
        Boolean x= false;  
  
        System.out.println(- - - ); // !x  
    }  
}
```



You're joined as an attendee. [Learn more](#)



```
class Demo {
```

```
public static void main(String args[]) {
```

```
    int i = -12;
```

```
    int j = 13;
```

```
    i = i & j; //4
```

```
    i = i | j; // -3
```

```
    i = (byte) i;
```

```
    System.out.println(i);
```

```
}
```

Q :

-12 binary

$$\begin{array}{l}
 0 \dots 0 / 0 \dots 0 / 0 \dots 0 \quad 0000 \underline{1100} \quad (12) \\
 1 \dots 1 / 1 \dots 1 / 1 \dots 1 \quad 1111 \underline{0100} \quad (-12) \\
 0 \dots 0 / 0 \dots 0 / 0 \dots 0 \quad 0000 \underline{1101} \quad (13)
 \end{array}
 \left. \vphantom{\begin{array}{l} 0 \dots 0 / 0 \dots 0 / 0 \dots 0 \\ 1 \dots 1 / 1 \dots 1 / 1 \dots 1 \\ 0 \dots 0 / 0 \dots 0 / 0 \dots 0 \end{array}} \right\} \text{OR}$$

$$\begin{array}{l}
 1 \dots 1 / 1 \dots 1 / 1 \dots 1 / 1111 \underline{1101} \quad (\\
 = \\
 \text{2's complement} \quad \begin{array}{r} 10 \\ +1 \\ \hline 11 \end{array} \quad (-3)
 \end{array}$$

```
class Demo {  
  
    public static void main(String args[]) {  
  
        byte i;  
        int j=9;  
        byte z=10;  
  
        i= j; // error : casting  
        i= (byte) j;  
        System.out.println(i );  
    }  
}
```



You're joined as an attendee. [Learn more](#)



```
class Demo {  
  
    public static void main(String args[]) {  
        int i=9;  
        int k = 0;  
  
        if(++i==1 || ++k==1)  
            System.out.println(i); //10  
        System.out.println(k); // 1  
    }  
}
```

1

```
class Demo {  
  
    public static void main(String args[]) {  
        int i=64;  
        int k = 0;  
        k = (i>k)? 10 : ---;  
        k = (i>k)? 10 : ---;  
  
        System.out.println(k); //10  
    }  
}
```




```
class Demo {  
  
    public static void main(String args[]) {  
  
        int i = 5;  
        int j=i>>2;  
  
  
        System.out.println(j); //1  
    }  
}
```

public class Main {

public static void main(String args[]) {

int i=0, j=0;

do {

i++; j = i % 5;

if (i == 5) break;

while (j < 5);

System.out.println("i" + i);

Output:

i j
1 1
2 2
3 3
4 4
5 0

1
2
3
4
5

1
2
3
4
5

(1 <= i <= 5)

1 <= i <= 5

1 <= i <= 5

1 <= i <= 5

Answer
True

$$(3 < 7)$$

□ 2023, published online 2023, 1794, 1-5, available from 2023, 1-5

$$(1 < 3) \text{ T}$$

$$(3 < 4)$$

$$\boxed{4} (x < 7)$$

$$(4 < 7) \text{ T}$$

$$(4 < 3) \text{ F}$$

$$\text{if } (x == 5)$$

$$(4 == 5) \text{ F}$$

$$\boxed{5} (x < 7)$$

$$(5 < 7) \text{ T}$$

$$\text{if } (x == 3) \\ s == 3 \text{ F}$$

$$\text{if } (x == 5) \\ (s == 5) \text{ T}$$

Answer: True

add's new loop

return string

add's state and everything again



add's state and everything again

add's state and everything again

add's state and everything again

1

Output



String Concatenation

0
2

2

(x < 7)

(2 < 7) T

2 (x = x + 1)

2 = 3 F

2 (x = 5)

3

(x < 7)

(3 < 7) T

3 (x = 3) T

Q. 5. Write a method `isPrime()` whether it is a prime number or not.

```
class Solution {
public:
    bool isPrime(int n) {
        if (n < 2) return false;
        for (int i = 2; i <= sqrt(n); i++)
            if (n % i == 0) return false;
        return true;
    }
};

// Example 1: Input: n = 2, Output: true (2 is a prime number)
// Example 2: Input: n = 4, Output: false (4 is not a prime number)
```

$$10101010 \div 10^{\text{len}}$$

Scan ①: ...

$$\frac{10101010}{10^{\text{len}}} = \text{isPrime()};$$

$$\boxed{} / 10;$$

Q23. Write a function and using some using given functions as presented in it.

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 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324

Q.1] Given a number, check whether it is a harmonic number or not.

Input: num = 7

public class Harmonic {

public static boolean isHarmonic(int num) { boolean isHarmonic =

int i = 1, sum = 0;

if (num < 1)

return false;

i = num/2;

while (i <= num/2)

sum += 1/i;

i = num/2;

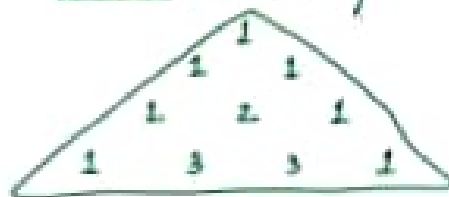
if (sum == num)

System.out.println("Harmonic Number is " + num);

else

System.out.println("Not an Harmonic Number" + num);

Output

sketch : Pascal p_y

12. Write a recursive method `isPrime` that returns `true` if `n` is prime and `false` otherwise.

$n \geq 1; n;$

$n \leq 1$ return `false`

if $n \leq 2$ return `true`

if $n \% 2 == 0$ return `false`

if $n \% 3 == 0$

return `false`

return `isPrime(n/2)`

return `true`

return `isPrime(n/3)`

return `true`

$n \geq 20$
complexity

```

1
public class Long1
{
    // static
    public static void main(String args[])
    {
        int i = 1;
        System.out.println(i);
        int i = 2; // local variable ←
        System.out.println(i);
        System.out.println(Long1.i);
    }
}

```

```

int i;
for (i = 1; i <= 3; i++)
    s.o.p(i);

```

var = ? duplicate variable

1
2
2
3
3

$$= \frac{(i-0)!(t-0+1)}{(i-0)!0!} \\ t=1 \quad t$$

$$\frac{t!}{(t-j)!}$$

term j

term j

$$\frac{nC_n}{nC_{n-1}} \cdot \frac{\frac{n!}{n!(n-n)}}{\frac{n!}{(n-0)!(n-n+1)!}}$$

$$= \frac{(n-1)!}{n!} \cdot \frac{(n-n+1)!}{(n-n)!}$$

$$= \frac{(n-n+1)}{n}$$

$$\frac{nC_n}{nC_{n-1}} = \frac{n-n+1}{n}$$

$$\Rightarrow nC_n = nC_{n-1} \cdot \frac{n-n+1}{n}$$

CODING P ...

AB

SG

C0

N

andita Bh...

Sourav Gos...

CODING P



Q.9

Q.9 Draw the following structure



n

```
for (i: 0; i < n; i++)
```

```
    for (j: 0; j < n-1; j++)
```

```
        cout << "16";
```

```
    }
```

```
for (a: 0; a < i; a++) {
```

```
    }
```

```
}
```

Q.10

Answer : Pascal Pyramid/Triangle


0/4 1
2
1
2
1
2

```
public class Nest2 {  
  
    public static void main(String args[]) {  
        int j=0;  
        do  
            for (int i=0; i++ < 2; )  
                System.out.println(i);  
        while (j++ < 2)  
    }  
}
```

Output:

1000 1000 1000 1000

1024

[illegible]
$$11 \dots 1 \mid 1 \dots 1 \mid 1 \dots 1 \mid 1000 \dots 0000 = 128.$$
$$12 \dots 3 \mid 1 \dots 1 \mid 11 \dots 00 \mid 0 \dots 0 = 124 \ll 3$$

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 105–112

1143

$$-v_2 \quad 0 \quad \dots \quad 0 \quad | \quad 1 \quad -0 \quad | \quad 000000011 \quad | \quad 1111 \quad 1111$$

1 + 1 532

32

$$00 - 0 \left| \begin{array}{ccc} 1 & 1 & 2 \\ 0 & 0 & 0 \end{array} \right| \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \left| \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right| \dots \dots \dots 0 \quad \text{etc}$$

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

4008 J. Neurosci., June 23, 2010 • 30(25):4003–4012

DOI: 10.1002/for

They said we were stupid, that we were illiterate.



```
public class Merge {  
    public static void main(String args[]) {  
        int i, j;  
        int arr1[] = {1, 2, 3, 4, 5};  
        int arr2[] = {6, 7, 8, 9, 10};  
        int arr3[] = new int[10];  
        if (arr1.length + arr2.length > arr3.length) {  
            System.out.println("Error: Array size mismatch");  
        }  
    }  
}
```

$$\begin{array}{cc|cc|cc} 2 & 1 & 1 & 1 & 1 & 1 \\ & & 2 & 1 & 2 & 2 \end{array}$$

How to solve this problem :

```

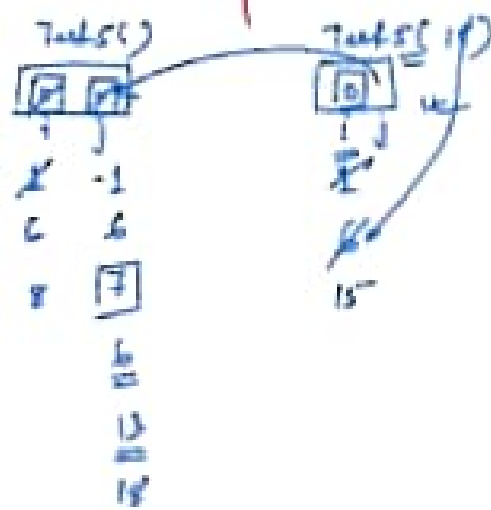
// C++
// static int i = 1; // counter
// public Test5() {
//     i++;
//     System.out.println(i + " ");
// }

```

```

// C++
// static int i = 1; // counter
// public Test5(int i) {
//     this.i = i;
//     System.out.println(i + " ");
// }
// public static void main(String[] args) {
//     Test5 t1 = new Test5(1);
//     Test5 t2 = new Test5(2);
//     Test5 t3 = new Test5(3);
//     Test5 t4 = new Test5(4);
//     Test5 t5 = new Test5(5);
// }

```



Op

1	-1	✓
8	7	✓
1	6	✓
15	15	✓


```

//
public class Test {
    int i;
    Test() {
        System.out.println("Java");
    }
    Test(int i) {
        this.i=i;
        this();
        System.out.println(i);
    }
    public static void main(String args[]) {
        new Test(7);
    }
}

```

```
class Demo {  
  
    public static void main(String args()) {  
  
        int x = 12;  
        int y = 5;  
        x = %y;  
        System.out.println(x );  
    }  
}
```

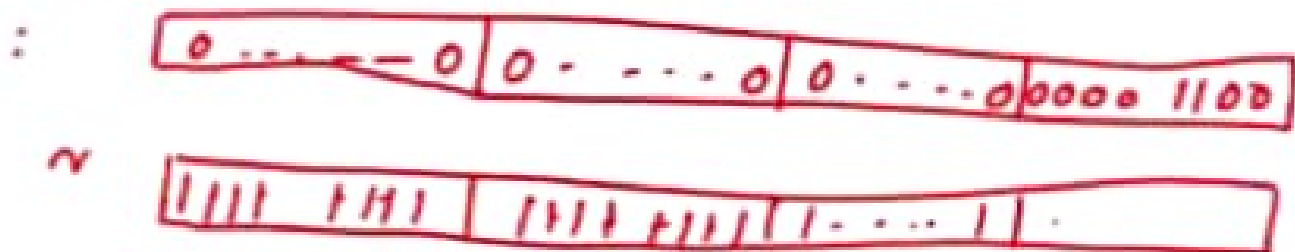
% :


```
class Demo {  
  
    public static void main(String args()) {  
  
        int x = 12;  
        int y = 5;  
        x%y;  
        System.out.println(x );  
    }  
}
```

% :

I

in 4 byte




```
class Demo {
```

```
public static void main(String args[]) {
```

```
    int i = -12;
```

```
    int j = 13;
```

```
    i = i & j;
```

```
    i = i | j;
```

```
    i = (byte) j;
```

```
    System.out.println(i);
```

```
}
```

% :

0...0 / 0...0 / 0...0 / 0000 1100 (12)
1...1 / 1...1 / 1...1 / 1111 0100 (-12) }
0...0 / 0...0 / 0...0 / 0000 1101 (13) }
0...0 / 0...0 / 0...0 / 0100 (4)

```
class Demo {
```

```
public static void main(String args[]) {
```

```
    int i = -12;
```

```
    int j = 13;
```

```
    i = i & j;
```

```
    i = (byte) j;
```

```
    System.out.println(i);
```

```
}
```

12 :

0...0 / 0...0 / 0...0 0000 1100 (12)
1...1 / 1...1 / 1...1 1111 0100 (-12)
0.

```
class Demo {
```

```
public static void main(String args[]) {
```

```
    int i = 12;
```

```
    int j = 13;
```

```
    i = i & j;
```

```
    i = (byte) j;
```

```
    System.out.println(i);
```

```
}
```

0/1 = 12

✓

0000 1100
0000 1101
8

```

class Demo {
    public static void main(String args[]) {
        int i = -12;
        int j = 13;
        i = i ^ j;

        System.out.println(i); // -7
    }
}

```

$0 \dots 0 / 0 \dots 0 / 0 \dots 0 / 0000 \ 1100 \ (12)$
 $1 \dots 1 / 1 \dots 1 / 1 \dots 1 / 1111 \ 0100 \ (-12)$
 $0 \dots 0 / 0 \dots 0 / 0 \dots 0 / 0000 \ 1101 \ (13)$

} XOR →

$1 \dots 1 / 1 \dots 1 / 1 \dots 1 / 1111 \ 1001 \ (-7)$

=
-ve

2's complement

(-ve)

110
 $+ 1$
 $111 \ (7)$

```

class Demo {
    public static void main(String args[]) {
        int i = -12;
        int j = 13;
        i = i ^ j;

        System.out.println(i); // -7
    }
}

```

$0 \dots 0 / 0 \dots 0 / 0 \dots 0 / 00001100 \quad (12)$
 $1 \dots 1 / 1 \dots 1 / 1 \dots 1 / 11110100 \quad (-12)$
 $0 \dots 0 / 0 \dots 0 / 0 \dots 0 / 00001101 \quad (13)$

} XOR →

$1 \dots 1 / 1 \dots 1 / 1 \dots 1 / 11111001 \quad (-7)$

$=$
 $-ve$

2's c .

```
class Demo {  
  
    public static void main(String args[]) {  
  
        int i = 5;  
        int j=i>>2;  
  
        System.out.println(i &  
    )}
```

class Demo {

public static void main(String args[]) {

int i=9;

int k = 0;

if(++i==1 || ++k==1)

System.out.println(i); //10

System.out.println(k); //

}}

```
class Demo {  
  
    public static void main(String args[]) {  
        int i=9;  
        int k = 0;  
        k= (i>k)? 10 : 20;  
        k'= (i<k)? 10 : 20;  
  
        System.out.println(k); //10  
        System.out.println(k'); //20  
    }  
}
```



```
class Demo {  
  
    public static void main(String args[]) {  
  
        int i = 5;  
        int j=i>>2;  
  
  
        System.out.println(j ); //1  
    }  
}
```

```
class Demo {  
  
    public static void main(String args[]) {  
  
        int i = - 5;  
        int j=i>>2;  
  
  
        System.out.println(j ); //-2  
    }  
}
```

Exmp V.3

24 July 2023

18:03

```
public class Nest1 {  
  
    public static void main(String args()) {  
  
        int i=0, j=9;  
  
        do {  
  
            i++;  
  
            if (j -- < i++) {  
                break;  
            } while (i < 5);  
  
            System.out.println(i + "\t" + j);  
        }  
    }  
}
```

Output:

4. Replaced the test of the for loop with a condition.

```

class Main {
    public static void main (String[] args) {
        int sum = 0;
        for (int i = 0; i <= 10; i++) {
            sum = sum + i;
            // System.out.println("Sum: " + sum);
        }
        System.out.println("Sum: " + sum);
    }
}

```

$\text{for } (i = 0; i < 4; i++)$
 $\text{for } (j = 0; j < 4; j++)$
 $\text{if } (arr[i] < arr[j])$
 $\text{swap}(arr[i], arr[j])$

Output

$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix}$

$(0 < 2) \text{ T}$
 $1 \leftarrow$
 $(1 < 2) \text{ T}$
 $(0 < 1) \text{ T}$
 $2 \leftarrow$
 $(2 < 2) \text{ F}$

$(j++)$
 $(0 < 2) \text{ T}$
 $1 \leftarrow$
 $(1 < 2) \text{ T}$
 $2 \leftarrow$
 $(2 < 2) \text{ F}$

$(i++)$
 $(0 < 2) \text{ T}$
 $1 \leftarrow$
 $(1 < 2) \text{ T}$
 $2 \leftarrow$
 $(2 < 2) \text{ F}$

```

public class SumOfSquares {
    public static void main(String[] args) {
        int n = 12;
        int sum = 0;
        for (int i = 1; i <= n; i++) {
            sum += i * i;
        }
        System.out.println("Sum of squares of 1 to " + n + " is: " + sum);
    }
}

```

Q1 i = 6 j = 12

Exmp V.5

24 July 2023

18.10

```
public class Loop {  
  
    static String o="";  
  
    public static void main(String args[]) {  
        int z;  
        for (int x=2; x<7; x++) {  
            if (x== 3) continue;  
  
            if (x==5) break z;  
            o=o+x;  
        }  
        System.out.println(o);  
    }  
}
```

Output:

2

4

10
10 10
10 10 10
10 10 10 10
10 10 10 10 10
10 10 10 10 10 10
10 10 10 10 10 10 10
10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10

$$L(x, y) = x^2 + y^2$$

Q.1. Write a number and add each digit of that number \rightarrow

$$x = y = \boxed{} \% 10;$$

$$sum = y;$$

$$y = \boxed{} / 10;$$

whether a number is divisible by 6 ?



LL1: Implement a method `isDigit()` which returns true if the character is a digit or not.

Buffer Reader (L) = Text Binary (System.in) // Keyboard.

Ex: 1. Buffer Reader class System.in
 → Scanner class
 • Code

Ex: 2 println ("0+4"); // Monitor

Q.8 Draw the following structure

```

*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *

```

2 variables : i, j

for ($i = 0$; $i < 7$; $i++$) {

for ($j = 0$; $j \leq i$; $j++$) {

System.out.print(" ");

end

System.out.println();

}

$$= \frac{(i-0)(i-0+1)}{(i-0)!+1}$$

production, which is based on the following assumptions:

1991

Downloaded from <http://ajph.org/> on May 11, 2015

$\log(\text{COP}) = \text{COP} \cdot \ln(10) = 2.303 \log(\text{COP})$
 $\text{NaOH} \cdot \text{mol} \cdot \text{g}^{-1} \cdot \text{mol}^{-1}$

Figure 10.10: A plot of the function $f(x) = \sin(x)$ for $x \in [0, 2\pi]$. The function is periodic and oscillates between -1 and 1.

100



100

100

100

Figure 1. The effect of the concentration of the solution on the adsorption of the dye.

10

Example 99 :



```

public class DemoPasc(a)
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter value of n: ");
        int n = sc.nextInt(); // 5

        int i = 1;

        for (jgg i = 0; i < n; i++)
        {
            for (jgg k = 0; k < n - i; k++)
                System.out.print(" ");
            for (jgg j = 0; j <= i; j++)
            {
                if (j == 0 || j == i)
                    s = 1;
                else
                    s = s * (i - j + 1) / j;

                System.out.print(s + " ");
            }
            System.out.println();
        }
    }
}

```

public static void main(String[] args) {

int i = 1; while (i <= 3) {

System.out.println(i);

i++; }

System.out.println(i); }

i = 1; i < 3

1 < 3 T Print ①

i = 1 + 1 = 2 2 < 3 T Print ②

i = 2 + 1 = 3 3 < 3 F X

i = 1

i + 1 < 3

① + 1 < 3 T; Print ②

i = 2

② + 1 < 3

T; Print ③

i = 3

3 + 1 < 3

F X

System.out.println(Loop 1.i); // ③

```
1 public class Loop2
2 {
3     public static void main(String args[])
4     {
5         int i = 0;
6         do
7         {
8             System.out.println(i);
9             i++;
10        } while(i < 10);
11    }
12 }
```

9/2

public static void main(String[] args) {
 int i=0, j=0, k=0;
 while(i<3 && j<3 && k<3){
 System.out.print(i+" "+j+" "+k+" ");
 i++;
 j++;
 k++;
 if(i==3 || j==3 || k==3){
 System.out.println();
 i=0; j=0; k=0;
 }
 }
}

o/p

0 0 0 | 0 1 2
 0 1 2 | 1 2 0
 1 2 0 | 2 0 1



while (i++ < 3)
 (0++ < 3) T s.o.print (.k++) = (0++) = 0;
 (1++ < 3) T s.o.p (k++) = (1++) = 1;
 (2++ < 3) T s.o.p (k++) (2++) = 2;
 (3++ < 3) F

do while (j++ < 3)
 (0++ < 3) T
 while (i++ < 3)
 4 < 3 F
 do while (i++ < 3)

```

public class Loop4
{
    public static void main(String args[])
    {
        int i=0, j=0, k=0;
        while(i<3)
        {
            while(j<3)
            {
                while(k<3)
                {
                    System.out.print(i+j+k+" ");
                    k++;
                }
                j++;
            }
            i++;
        }
    }
}

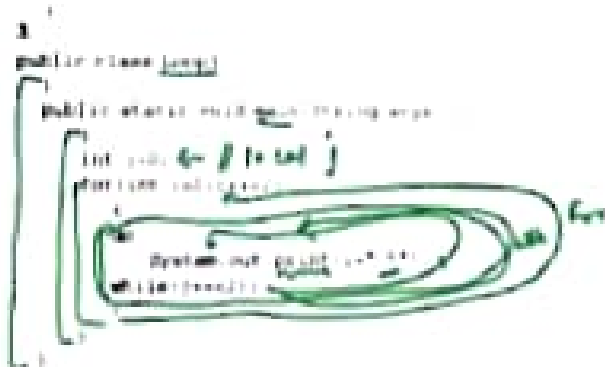
```



while (i++ < 3)
 (0 < 3) T S.o. print (i+j+k) = (0+0+0) = 0;
 (1 < 3) T S.o. print (i+j+k) = (1+0+0) = 1;
 (2 < 3) T S.o. print (i+j+k) = (2+0+0) = 2;

0/4

0 0 0 | 0 1 2 |
 1 1 1 | 1 2 3 |
 2 2 2 | 2 3 4 |



1: 1 1 1 2 | 1
1
1
2



$i + 1 < 2$
 $0 + 1 < 2$
1

Print: 1
while (j < 2)
 $0 + 1 < 2$ T
1
Print: 1
while (j < 2)
 $1 + 1 < 2$ T
2
Print(i): 1
while 2 + 1 < 2 F

$i + 1 < 2$

[Downloaded from ascelibrary.org by University of California, San Diego on 06/07/18. Copyright ASCE. For personal use; all rights reserved.](#)

Keywords: child sexual abuse; disclosure; social support

Figure 1

1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 26



14 Jan

1

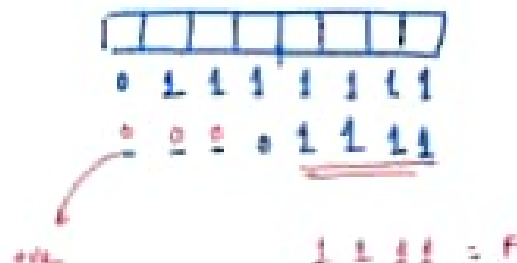
$$(1 \times 2) \text{ T 24 } (j = 0)$$
$$(2, 4, 12)^T = \sum_{i=1}^3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

(9.40)

```

1. class Demo
{
    public static void main(String [] arg)
    {
        byte a=127;
        a++;
        System.out.println(a);
    }
}

```



ok
15

Exercise IV.1

10 Aug 2024 10:00

Exercise IV.1

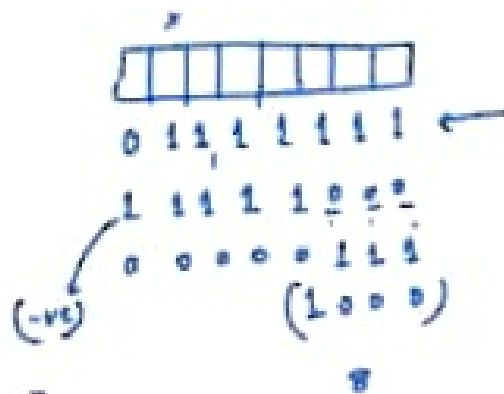
```

1 public static void main(String[] args)
2 {
3     System.out.println("Hello, World!");
4 }

```

0/0

10



V3

```

public class Longest {
    public static void main(String args[]) {
        int arr1[] = {1, 2, 3, 4, 5};
        int arr2[] = {2, 3, 4, 5, 6};
        if (arr1.length == arr2.length) {
            System.out.println("Arrays are of same length");
        } else {
            System.out.println("Arrays are of different length");
        }
    }
}

```

Handwritten notes: arr1 and arr2 are circled. A red arrow points to the `if` condition. The text "Case 1" is written next to the `if` block.

```

public class Longest {
    public static void main(String args[]) {
        int arr1[] = {1, 2, 3, 4, 5};
        int arr2[] = {2, 3, 4, 5, 6};
        if (arr1.length == arr2.length) {
            System.out.println("Arrays are of same length");
        } else {
            System.out.println("Arrays are of different length");
        }
    }
}

```

Handwritten notes: arr1 and arr2 are circled. A red arrow points to the `if` condition. The text "Case 2" is written next to the `else` block.

$$\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

$$\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

Thank you
Rishabh

Case 1
arr1

Case 2;	
0	1
0	2
1	0
1	2
2	0
2	1

```

public class loop8
{
    public static void main(String args[])
    {
        int i=0;
        for(int i=0;i<3;i++)
        {
            int j=0;
            for(int j=0;j<3;j++)
            {
                if(i==j)
                    break;
                System.out.print(i+"*"+j+" ");
            }
        }
    }
}

```

```

public class loop7
{
    public static void main(String args[])
    {
        int i=0;
        for(int i=0;i<3;i++)
        {
            int j=0;
            for(int j=0;j<3;j++)
            {
                if(i==j)
                    break;
                System.out.print(i+"*"+j+" ");
            }
        }
    }
}

```

2023-09-08
 100%
 100%

92



Figure 1

1997年1月1日 星期日 晴 1997年1月1日 星期日 晴 1997年1月1日 星期日 晴

1. **Chemical Engineering**



10. **Answer: D** The correct answer is D. The correct answer is D.

```
public static void main(String[] args) {
```

2. trans (E)-11 // replace original

21

$n = -127$

```
8. class Demo
{
    public static void main(String[] args)
    {
        byte n = 127;
        n--;
        System.out.println(n);
    }
}
```

8. class Demo

```
public static void main(String[] arg)
```

```
{
    int x=127;
```

```
    x=-x+3;
```

```
    System.out.println(x);
}
```

10111111 - 1024

✓

✓

✓

✓

0...0 | 0...0 | 0...0 | 01111111

1...1 | 1...1 | 1...1 | 10000000

1...1 | 11...1 | 11111111 | 00000000

0...0 | 0...0 | 00000011 | 1...1

0...0 | 0...0 | 00000100 | 0...0

1024

04

```

// Program to check if a string is a palindrome
// Input: "malayalam"
// Output: true

import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        String rev = "";
        for (int i = str.length() - 1; i >= 0; i--) {
            rev = str.charAt(i) + rev;
        }
        if (str.equals(rev)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }
}

```

$\frac{0}{1}$ factor



```

class Demo {
    public static void main (String args[]) {
        Demo x = new Demo();
    }
}

```

% No Error.

1

10

Armenia | galt's law

1014

Abstract

1143

```

1: 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
2: 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
3: 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
4: 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
5: 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
6: 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
7: 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
8: 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
9: 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
10: 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000

```


Downloaded from <http://ajphaphysiol.physiology.org/> at University of California, San Diego on May 12, 2015

129

1. [Introduction](#)
 2. [Getting started](#)
 3. [Getting started](#)
 4. [Getting started](#)

```

1 // class Demo
2 {
3     public static void main(String [] args)
4     {
5         int x=100;
6         System.out.println(x);
7     }
8 }

```

```

9 // class Demo
10 {
11     public static void main(String [] args)
12     {
13         int x=100;
14         System.out.println(x);
15     }
16 }

```

```

17 // class Demo
18 {
19     public static void main(String [] args)
20     {
21         int x=100;
22         System.out.println(x);
23     }
24 }

```

```

25 // class Demo
26 {
27     public static void main(String [] args)
28     {
29         int x=100;
30         System.out.println(x);
31     }
32 }

```

```
12. class Demo
```

```
    public static void main(String[] args)
```

```
    {
```

```
        int x=127;
```

```
        System.out.println(x);
```

```
    }
```

0/1
16 // Answer

$$x : (1+x) >>> 3$$

$$= (1+127) >>> 3$$

$$= 128 >>> 3$$

$$= 2^7 >>> 3$$

$$= 2^4$$

$$= 16$$

10000000
11111111
16

Operator	Operand	Type	Associativity
++	int	Increment	Left to Right
--	int	Decrement	Left to Right
*	int	Multiplication	Left to Right
/	int	Division	Left to Right
%	int	Modulus	Left to Right
+	int	Addition	Left to Right
-	int	Subtraction	Left to Right
<<	int	Left Shift	Left to Right
>>	int	Right Shift	Left to Right
<<<	int	Left Shift	Left to Right
>>>	int	Right Shift	Left to Right
<	int	Less Than	Left to Right
>	int	Greater Than	Left to Right
<=	int	Less Than or Equal	Left to Right
>=	int	Greater Than or Equal	Left to Right
==	int	Equal	Left to Right
!=	int	Not Equal	Left to Right
!	int	Not	Left to Right
&	int	Bitwise AND	Left to Right
	int	Bitwise OR	Left to Right
^	int	Bitwise XOR	Left to Right
~	int	Bitwise NOT	Left to Right
&&	int	Logical AND	Left to Right
	int	Logical OR	Left to Right
?:	int	Conditional	Right to Left
=	int	Assignment	Right to Left
+=	int	Compound Assignment	Right to Left
-=	int	Compound Assignment	Right to Left
*=	int	Compound Assignment	Right to Left
/=	int	Compound Assignment	Right to Left
%=	int	Compound Assignment	Right to Left


```

1 class Item
2 {
3     public static void main(String[] args)
4     {

```

Step 1: n=123

Step 2: n=123, p=1000000

Step 3: n=123, p=1000000

$x \ll n \gg$

Step 4
-8

123 : 0 1 1 1 1 1 1
 1 1 1 1 1 0 0 0

 0 0 0 0 1 1 1
 0 0 0 0 1 0 0 0

 8

1. (1000, 1000)

2. (1000, 1000) (1000, 1000) (1000, 1000)

3. (1000, 1000)

4. (1000, 1000)

5. (1000, 1000)

1000
1000

1000 1000

1000 1000

$$x = 122 \quad y = 128$$

$$x = (x + y) / 2 \quad y = y$$

$$x = (\underline{x} + 128) / 2 \quad y = y$$

$$x = (122 + 128) / 2 \quad y = 128$$

$$x = 125 \quad y = 128$$

$$x = 128 \quad y = 128$$

1. (1000, 1000)

1. (1000, 1000) (1000, 1000) (1000, 1000) (1000, 1000) (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)

1. (1000, 1000)


```
17 class Term
```

```
    public static void main(String[] args)
    {
        int x=123, y=123;
        System.out.println(x+y);
        System.out.println(x-y);
    }
}
```

```
18 class Term
```

```
    public static void main(String[] args)
    {
        int x=123, y=123;
        System.out.println(x+y);
        System.out.println(x-y);
    }
}
```

```
19 class Term
```

```
    public static void main(String[] args)
    {
        int x=123, y=123;
        System.out.println(x+y);
        System.out.println(x-y);
    }
}
```

```
20 class Term
```

```
    public static void main(String[] args)
    {
        int x=123, y=123;
        System.out.println(x+y);
        System.out.println(x-y);
    }
}
```

$\therefore \underline{120 \leq (132) \leq 5}$ 120
 $\therefore \underline{120 \leq (144 \cdot 32) \leq 5}$ 144
 $\therefore \underline{(120 \leq 3) \leq 5}$
 $\therefore \underline{120} \leq 5$ $2^{10} \rightarrow 2^5$
 $\therefore \underline{32}$ Ans. THANKS

```

1 //
2 public class Test4
3 {
4     public static void main(String[] args)
5     {
6         System.out.println("Hello World");
7     }
8     public static void main(String[] args)
9     {
10        System.out.println("Hello World");
11    }
12    public static void main(String[] args)
13    {
14        new Test4().main(args);
15    }
16 }

```



Figure 1. The effect of the concentration of the *Agrobacterium* suspension on the transformation efficiency of *Agrobacterium* strains.



$\frac{1}{2} \log \frac{1}{2} = -\frac{1}{2} \log 2 = -\frac{1}{2} \times 0.3010 = -0.1505$
 $\frac{1}{4} \log \frac{1}{4} = -\frac{1}{4} \log 4 = -\frac{1}{4} \times 0.6020 = -0.1505$
 $\frac{1}{8} \log \frac{1}{8} = -\frac{1}{8} \log 8 = -\frac{1}{8} \times 0.9030 = -0.1129$
 $\frac{1}{16} \log \frac{1}{16} = -\frac{1}{16} \log 16 = -\frac{1}{16} \times 1.2041 = -0.0753$
 $\frac{1}{32} \log \frac{1}{32} = -\frac{1}{32} \log 32 = -\frac{1}{32} \times 1.5051 = -0.0469$
 $\frac{1}{64} \log \frac{1}{64} = -\frac{1}{64} \log 64 = -\frac{1}{64} \times 1.8061 = -0.0282$
 $\frac{1}{128} \log \frac{1}{128} = -\frac{1}{128} \log 128 = -\frac{1}{128} \times 2.1071 = -0.0164$
 $\frac{1}{256} \log \frac{1}{256} = -\frac{1}{256} \log 256 = -\frac{1}{256} \times 2.4082 = -0.0094$
 $\frac{1}{512} \log \frac{1}{512} = -\frac{1}{512} \log 512 = -\frac{1}{512} \times 2.7092 = -0.0053$
 $\frac{1}{1024} \log \frac{1}{1024} = -\frac{1}{1024} \log 1024 = -\frac{1}{1024} \times 3.0103 = -0.0029$
 $\frac{1}{2048} \log \frac{1}{2048} = -\frac{1}{2048} \log 2048 = -\frac{1}{2048} \times 3.3113 = -0.0016$
 $\frac{1}{4096} \log \frac{1}{4096} = -\frac{1}{4096} \log 4096 = -\frac{1}{4096} \times 3.6123 = -0.0009$
 $\frac{1}{8192} \log \frac{1}{8192} = -\frac{1}{8192} \log 8192 = -\frac{1}{8192} \times 3.9133 = -0.0005$
 $\frac{1}{16384} \log \frac{1}{16384} = -\frac{1}{16384} \log 16384 = -\frac{1}{16384} \times 4.2143 = -0.0003$
 $\frac{1}{32768} \log \frac{1}{32768} = -\frac{1}{32768} \log 32768 = -\frac{1}{32768} \times 4.5153 = -0.0002$
 $\frac{1}{65536} \log \frac{1}{65536} = -\frac{1}{65536} \log 65536 = -\frac{1}{65536} \times 4.8163 = -0.0001$
 $\frac{1}{131072} \log \frac{1}{131072} = -\frac{1}{131072} \log 131072 = -\frac{1}{131072} \times 5.1173 = -0.0001$
 $\frac{1}{262144} \log \frac{1}{262144} = -\frac{1}{262144} \log 262144 = -\frac{1}{262144} \times 5.4183 = -0.0001$
 $\frac{1}{524288} \log \frac{1}{524288} = -\frac{1}{524288} \log 524288 = -\frac{1}{524288} \times 5.7193 = -0.0001$
 $\frac{1}{1048576} \log \frac{1}{1048576} = -\frac{1}{1048576} \log 1048576 = -\frac{1}{1048576} \times 6.0203 = -0.0001$
 $\frac{1}{2097152} \log \frac{1}{2097152} = -\frac{1}{2097152} \log 2097152 = -\frac{1}{2097152} \times 6.3213 = -0.0001$
 $\frac{1}{4194304} \log \frac{1}{4194304} = -\frac{1}{4194304} \log 4194304 = -\frac{1}{4194304} \times 6.6223 = -0.0001$
 $\frac{1}{8388608} \log \frac{1}{8388608} = -\frac{1}{8388608} \log 8388608 = -\frac{1}{8388608} \times 6.9233 = -0.0001$
 $\frac{1}{16777216} \log \frac{1}{16777216} = -\frac{1}{16777216} \log 16777216 = -\frac{1}{16777216} \times 7.2243 = -0.0001$
 $\frac{1}{33554432} \log \frac{1}{33554432} = -\frac{1}{33554432} \log 33554432 = -\frac{1}{33554432} \times 7.5253 = -0.0001$
 $\frac{1}{67108864} \log \frac{1}{67108864} = -\frac{1}{67108864} \log 67108864 = -\frac{1}{67108864} \times 7.8263 = -0.0001$
 $\frac{1}{134217728} \log \frac{1}{134217728} = -\frac{1}{134217728} \log 134217728 = -\frac{1}{134217728} \times 8.1273 = -0.0001$
 $\frac{1}{268435456} \log \frac{1}{268435456} = -\frac{1}{268435456} \log 268435456 = -\frac{1}{268435456} \times 8.4283 = -0.0001$
 $\frac{1}{536870912} \log \frac{1}{536870912} = -\frac{1}{536870912} \log 536870912 = -\frac{1}{536870912} \times 8.7293 = -0.0001$
 $\frac{1}{1073741824} \log \frac{1}{1073741824} = -\frac{1}{1073741824} \log 1073741824 = -\frac{1}{1073741824} \times 9.0303 = -0.0001$
 $\frac{1}{2147483648} \log \frac{1}{2147483648} = -\frac{1}{2147483648} \log 2147483648 = -\frac{1}{2147483648} \times 9.3313 = -0.0001$
 $\frac{1}{4294967296} \log \frac{1}{4294967296} = -\frac{1}{4294967296} \log 4294967296 = -\frac{1}{4294967296} \times 9.6323 = -0.0001$
 $\frac{1}{8589934592} \log \frac{1}{8589934592} = -\frac{1}{8589934592} \log 8589934592 = -\frac{1}{8589934592} \times 9.9333 = -0.0001$
 $\frac{1}{17179869184} \log \frac{1}{17179869184} = -\frac{1}{17179869184} \log 17179869184 = -\frac{1}{17179869184} \times 10.2343 = -0.0001$
 $\frac{1}{34359738368} \log \frac{1}{34359738368} = -\frac{1}{34359738368} \log 34359738368 = -\frac{1}{34359738368} \times 10.5353 = -0.0001$
 $\frac{1}{68719476736} \log \frac{1}{68719476736} = -\frac{1}{68719476736} \log 68719476736 = -\frac{1}{68719476736} \times 10.8363 = -0.0001$
 $\frac{1}{137438953472} \log \frac{1}{137438953472} = -\frac{1}{137438953472} \log 137438953472 = -\frac{1}{137438953472} \times 11.1373 = -0.0001$
 $\frac{1}{274877906944} \log \frac{1}{274877906944} = -\frac{1}{274877906944} \log 274877906944 = -\frac{1}{274877906944} \times 11.4383 = -0.0001$
 $\frac{1}{549755813888} \log \frac{1}{549755813888} = -\frac{1}{549755813888} \log 549755813888 = -\frac{1}{549755813888} \times 11.7393 = -0.0001$
 $\frac{1}{1099511627776} \log \frac{1}{1099511627776} = -\frac{1}{1099511627776} \log 1099511627776 = -\frac{1}{1099511627776} \times 12.0403 = -0.0001$
 $\frac{1}{2199023255552} \log \frac{1}{2199023255552} = -\frac{1}{2199023255552} \log 2199023255552 = -\frac{1}{2199023255552} \times 12.3413 = -0.0001$
 $\frac{1}{4398046511104} \log \frac{1}{4398046511104} = -\frac{1}{4398046511104} \log 4398046511104 = -\frac{1}{4398046511104} \times 12.6423 = -0.0001$
 $\frac{1}{8796093022208} \log \frac{1}{8796093022208} = -\frac{1}{8796093022208} \log 8796093022208 = -\frac{1}{8796093022208} \times 12.9433 = -0.0001$
 $\frac{1}{17592186044416} \log \frac{1}{17592186044416} = -\frac{1}{175$

2
1000

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	60
61	62
63	64
65	66
67	68
69	70
71	72
73	74
75	76
77	78
79	80
81	82
83	84
85	86
87	88
89	90
91	92
93	94
95	96
97	98
99	100

$$= (f^2, \frac{1}{2}) \rightarrow 1 + f^2 = 1 + 1 = 2$$

1000

600

$$\frac{1}{(1 - \frac{1}{x})} = \sum_{n=0}^{\infty} x^{-n}$$

(1.4.2) 7

$$\uparrow \quad \text{mit } \left(\frac{1}{2} (1 + \sqrt{5}) \right) < 2$$

(2 4 2) 6 — 6.7.8

$$\text{set } L_2 = \{ -1, 0, 1 \} \cup \{ 2 \} \quad \text{--- } \text{cop } j = 2$$
$$= 1 \cdot 9^2 \cdot \sqrt{1} = 9$$
$$\text{rank} \begin{pmatrix} 2 & 4 & 1 \end{pmatrix} = 1$$

100

1-1

public class Test1 {

public static void main(String[] args) {

System.out.println("Hello");

}
}

System.out.println("String");

public static void main(String[] args) {

new Test1().main();

main

String

TRY?

- A. **Design** a class such that only one object of that class can get created by any user.
- B. Create an object of a class, which is having only one private constructor and which no other constructor can be defined.

class Demo {

~~static~~ boolean isCreated; //0

Demo() {

cancel();

}

static Demo cancel() {

if (!isCreated)

return this;

else return null;

}

public static void main() {

Demo obj = new Demo();

Demo obj2 = new Demo();

}

Q3. Write a java program to Calculate the number of objects created by a particular class.

Q. Design a class such that only one object of that class can get created by any user.

Q. Create an object of a class, which is having only one private constructor and which has other constructor can be defined.

class Demo {
static int count;
 Demo() {

Class Demo {

int i;

Static int count = 0; // 0

Demo() {

Count = count + 1;

}

Demo(int a) {

}

}

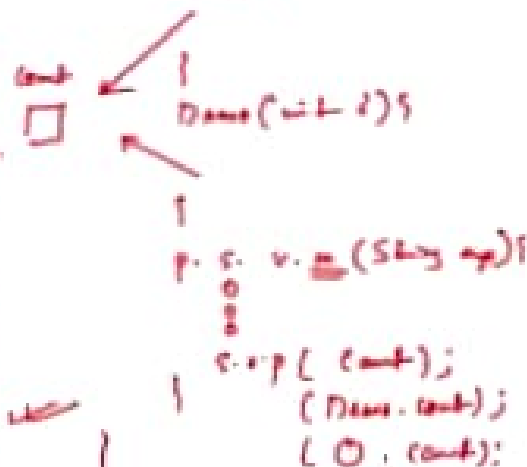
Public static void main(String args[]) {

Demo o1 = new Demo();

Demo o2 = new Demo();

Demo o3 = new Demo();

Demo o4 = new Demo(10);



- ```

class Demo {
 static int count;
}

// Demo (with id) {
// p = s.v.m("String ap");
// o.o.p(count);
// }
// (Demo.count);
// (o.count);

```

```

1 //
2 public class Test1 {
3 • int i;
4 • Test1() {
5 System.out.println("Java"); //
6 }
7 • Test1(int i) {
8 this.i = i;
9 this.i = i;
10 System.out.println(i); //
11 }
12 • public static void main(String args[]) {
13 Test1 t1 = new Test1(1); //
14 }
15 }

```



This is a recursive call to the constructor.  
 This is an empty argument based call.

If Java is not working, then the output will be an error.

28.04.2021 12:28

4. \* Design a class such that only one object of that class can get created by any user.

5. \* Create an object of a class, which is having only one private constructor and which's no other constructor can be defined.

TRY?

```
class Demo {
```

```
 <static> boolean isCreated; //0
```

```
 Demo() {
```

```
 cancel();
```

```
 }
```

```
 static Demo cancel() {
```

```
 if (isCreated)
 return this;
```

```
 else return null;
```

```
 }
```

```
 public static void main() {
```

```
 Demo d1 = new Demo();
```

```
 Demo d2 = new Demo();
```

```
 }
```

1. A. Singleton is a class such that only one object of that class can get created by any user.

2. Singleton is an object of a class, which is having only one private constructor and which the other constructor can be defined.

TRY?

```
class Demo {
```

```
 <static> boolean isCreated; //0
```

```
 Demo() {
```

```
 cancel();
```

```
 }
```

```
 static Demo cancel() {
```

```
 if (isCreated)
```

```
 return this;
```

```
 else return null;
```

```
 }
```

```
 public static void main () {
```

```
 Demo o1 = new Demo();
```

```
 Demo o2 = new Demo();
```

```
 }
```

```
public class Cast{
 public static void main(String args[]) {

 int i=10;

 long l=1;

 double d=3.4;

 byte b;
 b= (byte)i;
 System.out.println(b);

 i= (int) l;
 System.out.println(i);

 l=(long) d;
 System.out.println(l);
 }
}
```

File: src.java

```

class src {
 public static void main (String args []) {
 System.out.println("Hello");
 }
}

```

File: Hello.java

```

public class Hello {
 public static void main (String args []) {
 System.out.println("Hello");
 }
}

```

A ✓

✓ ✓

or Raia or

Rahat {
   
 Capita : Java XYX.java
   
 Ron : Java Hello.

{
   
 Capita = Java Hello.java
   
 Ron = Java Hello.



```
public class Cast{
 public static void main(String args[])

 int i=10;

 long l=1;

 double d=3.4;

 byte b;
 b= (byte)i;
 System.out.println(b);

 j= (int) l;
 System.out.println(i);

 l=(long) d;
```



```
class Demo {

 public static void main(String args()) {

 int x=12;

 System.out.println(~x);

 }
}
```

*b/p*     -12   |   -13

```
public class Cast{
 public static void main(String args[]){

 int i=10;

 long l=1;

 double d=3.4;

 byte b;
 b= (byte)i;
 System.out.println(b); # -> b: 10

 i= (int) l;
 System.out.println(i); // i = 1

 l=(long) d;
 System.out.println(l); // d = 3
 }
}
```

```
class Demo {

 public static void main(String args[]) {

 int i = -12;
 int j=13;

 j=i^j;

 System.out.println(i);
 }
}
```

```
class Demo {
```

```
public static void main(String args[]) {
```

```
 int i = -12;
```

```
 int j = 13;
```

```
 i = i & j;
```

```
 i = (byte) j;
```

```
 System.out.println(i);
```

```
}
```

% :

0000 1100 (12)  
110. (-12)

```
class Demo {

 public static void main(String args()) {

 byte i=10;
 byte j=9;
 int i=j+i;
 i= (byte) (j+i);
 i= (byte) j+ (byte)i;

 System.out.println(i);
 }
}
```

ETTH

```
class Demo {

 public static void main(String args[]) {
 int i=64;
 int k = 0;
 int k'=0;
 k= (i>k)? 10 : ---;
 k= (i>k)? --- : ---;

 System.out.println(k); //10
 System.out.println(k'); //10
 }
}
```

```
class Demo {
```

```
public static void main(String args[]) {
```

```
 int i = 5;
```

```
 int j=i>>2;
```



```
 System.out.println(j); //1
```

```
 }
```

```
class Demo {
```

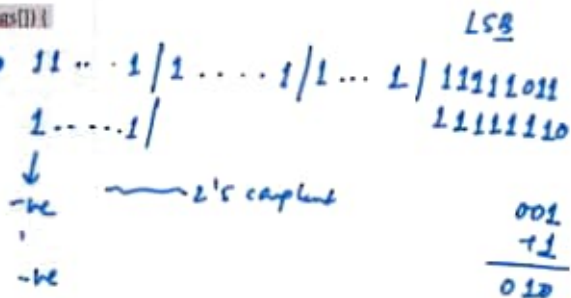
```
public static void main(String args[]) {
```

```
 int i = -5;
```

```
 int j=i>>2;
```

```
 System.out.println(j); //-2
```

```
 }
```





```
class Test2 {
 public static void main(String args[]) {
 int s=0;
 for (int n=1;n<=10; n++) {
 s=s+n;
 }
 System.out.println("Summation of the first ten natural no. : "+s);
 }
}
```

```
class Demo {
```

```
 public static void main(String args[]) {
```

```
 int i=9;
```

```
 int k = 0;
```

```
 if(++i==1 || ++k==1)
```

```
 System.out.println(i); //10
```

```
 }
```

double

1

$\frac{1}{2}$  (

)

2

$\frac{1}{2}$  (

) F

(j + 1 < 2)

(① + 1 < ②) T

double

i

$\frac{1}{2}$  (

)

1

$\frac{1}{2}$  (

)

2

$\frac{1}{2}$  (

) F

publ. in: *Journal of the American Statistical Association*, 1993, 88, 1, 1-11



...and the ...

1. *Journal of the American Medical Association*, 1997; 278: 1019-1024.

1000

100

1/23/2014

100

$$\left[ \begin{array}{c|c|c} 0 & 0 & 1 \\ 1 & 1 & 2 \\ 2 & 1 & 1 \end{array} \right]$$

$(i + t < 2)$   
 $(0 + 1 < 2) \text{ T}$   
 $1 \leftarrow$   
 $(i + t < 2)$   
 $(1 + 1 < 2) \text{ T}$   
 $2 \leftarrow$   
 $(i + t < 2)$   
 $(2 + 1 < 2) \text{ F}$

$$\begin{array}{c|ccc} & \text{0} & \text{1} & \text{2} \\ \hline \text{0} & 0 & 1 & 1 \\ \text{1} & 1 & 0 & 1 \\ \text{2} & 1 & 1 & 0 \end{array}$$
 $(j+1)$

Q. Calculate the sum of the first 10 integers.

```
int i = 1; // initialization
do {
```

```
 sum = sum + i;
```

```
 i++; // incrementation
```

```
} while (i <= 10);
```

```
 S.o.p(sum);
```

□

```
int sum = 0;
```

```
for (int i = 0; i <= 10; i++) {
```

```
 sum = sum + i;
 S.o.p(sum);
}
```

for

Adv: Efficiency  
int i = 1; sum  
 while ( i <= 10 ) {

while

```
 sum = sum + i;
 i++; // incrementation
```

```
 S.o.p(sum);
}
```

Q. 5. Find remainder when 10101010 is divided by 101.

10101010

$$101 \overline{) 10101010} = 101010$$

From ABCB report

Concave  $\rightarrow$  real images (upside down);

```
char c : {endl, chard(c); // Thank you
```

$$= \frac{1}{2} \left( \frac{1}{2} \right) e_j$$


© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 105–112


$$I_n = \{0, i \in \mathbb{N}, i \leq n\}$$
$$x_n(j; 0; j \leq n-1; j+n)$$
$$f_p(u) = \frac{1}{p} \log \left( \frac{1}{p} \right)$$
$$\left[ f_m(x, 0; k \leq \frac{i}{2}; k+1) \right]$$


Shape : Pascal Pyramid/Triangle


$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$L(j:0, j+1, j+1)$$

50, (1962),

$$L(k_0, k_1, k_2)$$

5.7 (











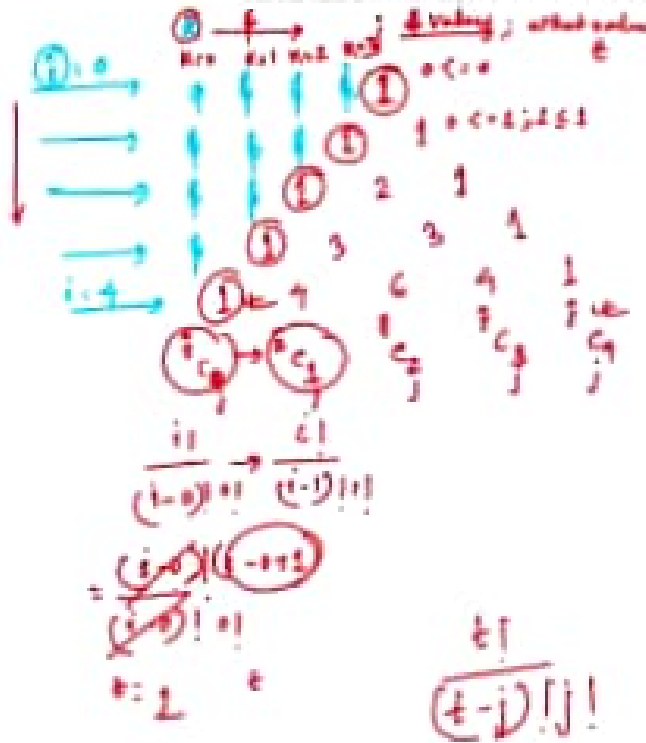


1944 - 1945

$$1 = 1 + i$$


174

- no. of values
- what are the values

[illegible]

Complexity % :

Public class DemoPerceptron

{  
public static void main(String[] args)

{  
Scanner sc = new Scanner(System.in);  
System.out.print("Enter value of n : ");  
int n = sc.nextInt();

int i = 1;

for (int j = 0; j < n; j++)

{  
for (int k = 0; k < n - j; k++)

{  
System.out.print(" ");

for (int l = 0; l < n - j - k; l++)

{  
if (j == 0 && l == 0)

{  
i = 1; // starting value

{  
i = i \* (n - j - k - l);

System.out.print(i + " ");

System.out.println();

System.out.print(i);

System.out.println();

System.out.println();

System.out.println();

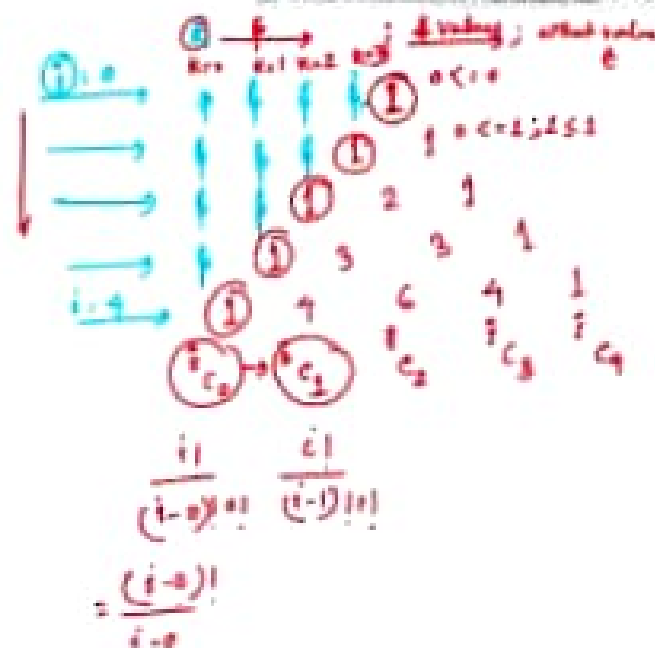
System.out.println();

System.out.println();

System.out.println();

System.out.println();

System.out.println();

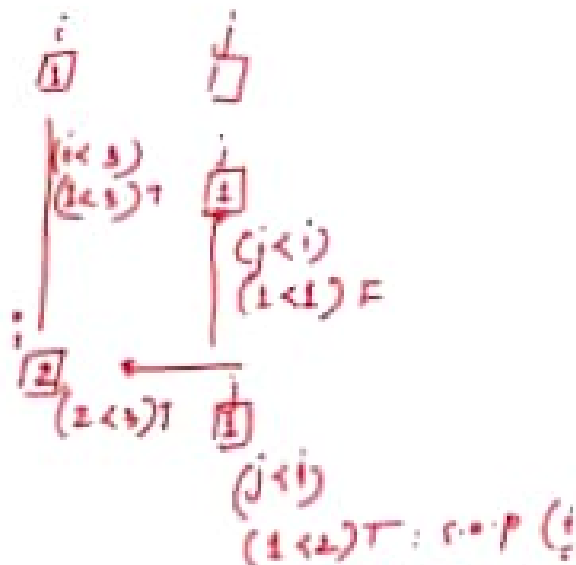


```

public class temp {
 public static void main(String ar[]) {
 int n = 3; // number of local & local
 for (int i = 1; i <= n; i++) // outer for i row
 for (int j = 1; j <= n; j++) // inner for j col
 System.out.print(i * " * " + j + " ");
 }
}

```

2 1



$$\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 1 & 2 \\ \hline \end{array}$$

$$\begin{array}{c} \frac{2}{3} \\ (2+i < 3) \text{ T} \\ \frac{3}{4} \\ (3+i < 3) \text{ F} \end{array} \quad \text{s.op (k+i) } \left( \frac{2+i}{3} \right) = 2;$$

$$\begin{array}{l} \text{do while (j+i < 3)} \\ \quad (2+i < 3) \text{ T} \\ \quad \quad \text{while (i+i < 3)} \\ \quad \quad \quad 4 < 3 \text{ F} \end{array}$$

$$\begin{array}{l} \text{do while (j+i < 3)} \\ \quad (1+i < 3) \text{ T} \\ \quad \quad \text{while (i+i < 3)} \end{array}$$



00:00:00.000 00:00:00.000

```
public class Loop9
{
 public static void main(String args[])
 {
 a1:
 for(int i=0; i<3; i++)
 {
 a2:
 for(int j=0; j<1; j++)
 {
 if(i==j)
 continue a1;

 System.out.println(i+"*"+j+"=");
 }
 }
 }
}
```

```

public class Longs {
 public static int validMatch(String s) {
 int i = 0;
 while (i < s.length()) {
 if (s.charAt(i) == '('') {
 // push onto stack
 stack.push(s.charAt(i));
 } else if (s.charAt(i) == ')') {
 // pop from stack
 if (stack.isEmpty()) {
 return false;
 }
 char c = stack.pop();
 if (c != '('') {
 return false;
 }
 }
 i++;
 }
 return stack.isEmpty();
 }
}

```

$$\begin{array}{c|c|c}
 2 & 1 & 1 \\
 \hline
 2 & 2 & 2
 \end{array}$$

$$\left. \begin{array}{l} (i < 3) \\ (1 < 3) \end{array} \right\} T; \boxed{1}$$

$$\left. \begin{array}{l} (j \leq i) \\ (1 \leq 1) \end{array} \right\} T$$

$$\left. \begin{array}{l} (j = i) \\ 1 = 1 \end{array} \right\} \text{True}$$

s.o.p. (i, j) || 1 1

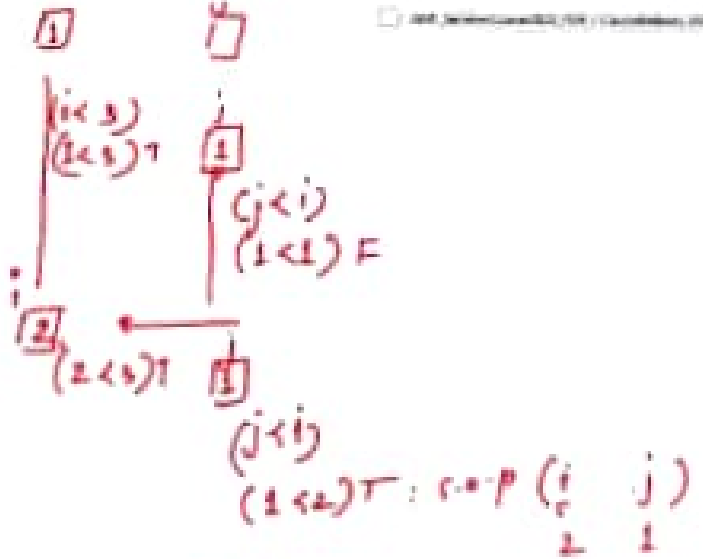
$$\left. \begin{array}{l} (i < 3) \\ 2 < 3 \end{array} \right\} T; \boxed{2}$$

$$\left. \begin{array}{l} (j \leq i) \\ (1 \leq 2) \end{array} \right\} T$$

$$\left. \begin{array}{l} (j = i) \\ (1 = 2) \end{array} \right\} \text{False}$$

```
public class temp {
 public static void main(String arr[]) {
 // Local i & local j
 for (int i = 0; i < arr.length; i++) {
 for (int j = i; j < arr.length; j++) {
 System.out.print(i + " " + j + " ");
 }
 }
 }
}
```

Output: 0 1 2 1 2 3 0 1 2 3 4



**Abstract**



Copyright © 2006 John Wiley & Sons, Ltd.

1

**WILEY**

| Age Group | Total (%) | Male (%) | Female (%) | Unknown (%) |
|-----------|-----------|----------|------------|-------------|
| 18-24     | 10        | 10       | 10         | 10          |
| 25-34     | 25        | 25       | 25         | 25          |
| 35-44     | 35        | 35       | 35         | 35          |
| 45-54     | 20        | 20       | 20         | 20          |
| 55-64     | 10        | 10       | 10         | 10          |
| 65+       | 10        | 10       | 10         | 10          |

1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 26



$$\begin{array}{r} 255 \\ - 9 \\ \hline 246 \end{array}$$

Ques. Write a program to find the sum of all the elements in an array.

public static void main(String args[])

{  
 int arr[] = {1, 2, 3, 4, 5};

int sum = 0;  
 for (int i = 0; i < arr.length; i++)

{  
 sum += arr[i];  
 System.out.println("Sum of array is: " + sum);  
 }

}

Ans

Ques. Write a program to find the sum of all the elements in an array.

public static void main(String args[])

{  
 int arr[] = {1, 2, 3, 4, 5};

int sum = 0;  
 for (int i = 0; i < arr.length; i++)

{  
 sum += arr[i];  
 System.out.println("Sum of array is: " + sum);  
 }

}

Ans  
 Sum of array is: 15

Thank you!  
 Rishabh

Ques. Write a program to find the sum of all the elements in an array.

Ans

```
class Demo
{
 public static void main(String args[])
 {
 int x=3;
 {
 int x=7;
 System.out.println(x);
 }
 System.out.println(x);
 }
}
```

o/p Error : Duplicate local variable



## N.7

**Abstract**

```

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
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040

```

1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 26

Copyright © 2004 John Wiley & Sons, Inc.

$\mu \leq 3$  ;

Sp. Brain | Typenart.  
-8





## 6.7

27 July 2023

DS-53

```
class Demo(
 public static void main(String f[])
{
 Demo x=new Demo();
 x.fun1();
}
static void fun()
{
 System.out.println("Hello!");
}
void fun1()
{
 System.out.println("Hi");
 fun();
 Demo.fun();
 Demo x=new Demo ();
 x.fun();
}
}
```

o/p

Hi  
Hello  
Hello  
Hello .

```

// Demo101.java
// constructor
//
// System.out.println("Hello, World!");
//
// public static void main(String a[]) {}
//
// {
// Demo d = new Demo();
// }

```

o/p  
≠ error

| Precedence | Operator                               | Type                                                                                                                                                   | Associativity |
|------------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| 15         | ()<br>[]                               | Parentheses<br>Array subscript<br>Member selection                                                                                                     | Left to Right |
| 14         | ++<br>--                               | Unary post-increment<br>Unary post-decrement                                                                                                           | Right to left |
| 13         | ++<br>--<br>+<br>-<br>!<br>~<br>(type) | Unary pre-increment<br>Unary pre-decrement<br>Unary plus<br>Unary minus<br>Unary logical negation<br>Unary bitwise complement<br>Unary type cast       | Right to left |
| 12         | *<br>/<br>%                            | Multiplication<br>Division<br>Modulus                                                                                                                  | Left to right |
| 11         | +<br>-                                 | Addition<br>Subtraction                                                                                                                                | Left to right |
| 10         | <<<br>>><br>>>>                        | Bitwise left shift<br>Bitwise right shift with sign extension<br>Bitwise right shift with zero extension                                               | Left to right |
| 9          | <<br><=<br>><br>>=<br>instanceof       | Relational less than<br>Relational less than or equal<br>Relational greater than<br>Relational greater than or equal<br>Type comparison (objects only) | Left to right |
| 8          | =<br>!=                                | Relational is equal to<br>Relational is not equal to                                                                                                   | Left to right |
| 7          | &                                      | Bitwise AND                                                                                                                                            | Left to right |
| 6          | ^                                      | Bitwise exclusive OR                                                                                                                                   | Left to right |
| 5          |                                        | Bitwise inclusive OR                                                                                                                                   | Left to right |
| 4          | &&                                     | Logical AND                                                                                                                                            | Left to right |
| 3          |                                        | Logical OR                                                                                                                                             | Left to right |
| 2          | ?:                                     | Ternary conditional                                                                                                                                    | Right to left |
| 1          | =<br>+=<br>-=<br>*=<br>/=<br>%=        | Assignment<br>Addition assignment<br>Subtraction assignment<br>Multiplication assignment<br>Division assignment<br>Modulus assignment                  | Right to left |

2/2      hi ✓  
             Stelle ✓  
             Hahn  
             Hahn

17. class Item:

```
 public static void main(String[] args) {
 int n=10, p=100;
 int[] arr = new int[n];
 System.out.println(arr);
 }
}
```

18. class Item:

```
 public static void main(String[] args) {
 int n=10, p=100;
 int[] arr = new int[n];
 System.out.println(arr);
 }
}
```

19. class Item:

```
 public static void main(String[] args) {
 int n=10, p=100;
 int[] arr = new int[n];
 System.out.println(arr);
 }
}
```

20. class Item:

```
 public static void main(String[] args) {
 int n=10, p=100;
 int[] arr = new int[n];
 System.out.println(arr);
 }
}
```



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course

Java 2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course



Java

2023 Summer Course

1999-2000 2000-2001 2001-2002

Downloaded from <http://www.jstor.org/stable/2346002> on Tue, 27 Jun 2017 12:00:00 UTC

Received 10 November 2004; accepted 12 January 2005



« « » »

Left  $\rightarrow$  Right

$$cc/32 \xrightarrow{\text{Lap}} \text{Exp}$$

→ Eye to Left

134

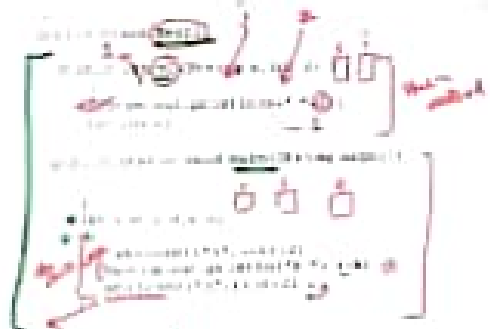


$$2^{10} \rightarrow 2^5$$

22

Ans. ALL THANKS

[illegible]



Handwritten notes in green ink:

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

Handwritten code snippets in red ink:

```

while (true) {
 // ...
 if (x < 2) {
 // ...
 } else {
 // ...
 }
}

```

Handwritten notes in red ink:

1.  $x = 1$   
 $(1 < 2) \rightarrow \text{True} \rightarrow \text{loop}$   
 $x = 2$   
 $(2 < 2) \rightarrow \text{False} \rightarrow \text{exit}$

2.  $x = 2$   
 $(2 < 2) \rightarrow \text{False} \rightarrow \text{exit}$

3.  $x = 3$   
 $(3 < 2) \rightarrow \text{False} \rightarrow \text{exit}$

4.  $x = 4$   
 $(4 < 2) \rightarrow \text{False} \rightarrow \text{exit}$

5.  $x = 5$   
 $(5 < 2) \rightarrow \text{False} \rightarrow \text{exit}$

6.  $x = 6$   
 $(6 < 2) \rightarrow \text{False} \rightarrow \text{exit}$

7.  $x = 7$   
 $(7 < 2) \rightarrow \text{False} \rightarrow \text{exit}$

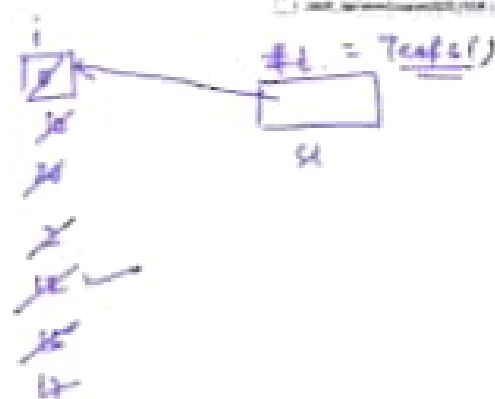
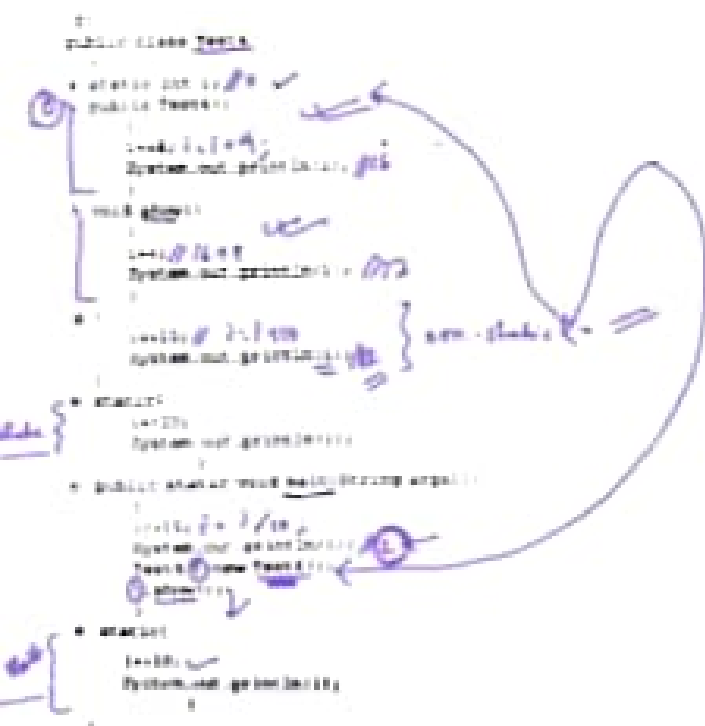
8.  $x = 8$   
 $(8 < 2) \rightarrow \text{False} \rightarrow \text{exit}$

9.  $x = 9$   
 $(9 < 2) \rightarrow \text{False} \rightarrow \text{exit}$

10.  $x = 10$   
 $(10 < 2) \rightarrow \text{False} \rightarrow \text{exit}$







```
1 //
2 public class Test6
3 {
4 static int i;
5 public Test6()
6 {
7 i+=4;
8 System.out.println(i);
9 }
10 void show()
11 {
12 i++;
13 System.out.println(i);
14 }
15 {
16 i+=10;
17 System.out.println(i);
18 }
19 static(
20 i+=10;
21 System.out.println(i);
22)
23 public static void main(String args[])
24 {
25 i+=10;
26 System.out.println(i);
27 Test6 t=new Test6();
28 t.show();
29 }
30 static(
31 i+=10;
32 System.out.println(i);
33)
34 }
```

```
 int i=1;
 public static Test5()
 {
 int i=1;
 static int j;
 public Test5()
 {
 i+=2;
 j++;
 System.out.println(i+"*"+j);
 }
 }
 i+=1;
 j+=2;
 System.out.println(i+"*"+j);
}
public Test5(int i)
{
 this.i=i;
 public
 {
 System.out.println(i+j);
 }
}
Test5 t;
Test5 t1;
Test5 t2;
public static void main(String args[])
{
 new Test5(1);
 new Test5(1);
}
}
```

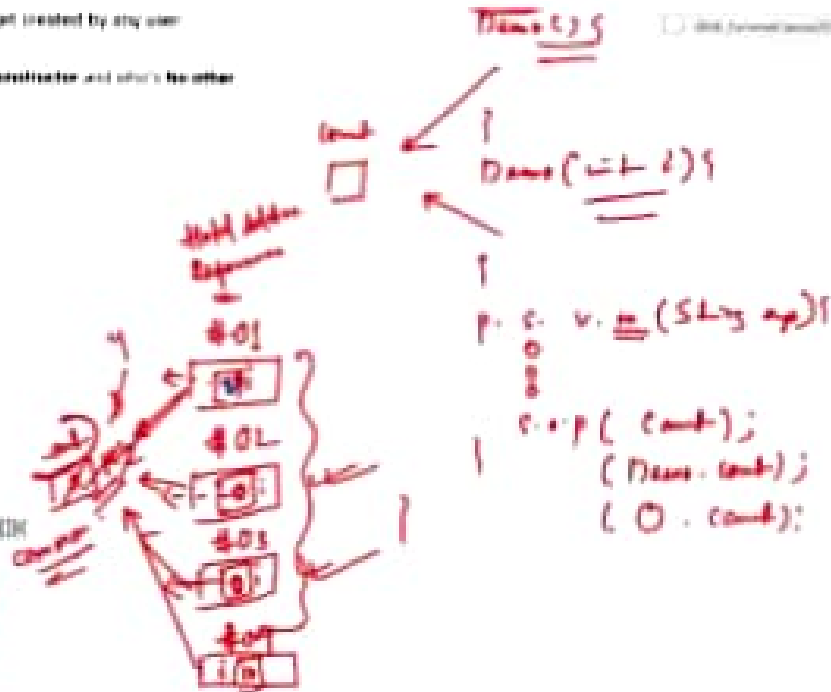
Q. Design a class such that only one object of that class can get created by any user.

Q. Write an object of a class, which is having only one private constructor and where no other constructor can be defined.

```

Class Demo {
 int i;
 Static {int count = 0; // 0
 Demo() {
 count = count + 1;
 }
 Demo(int x) {
 i = x;
 count = count + 1;
 }
 Public static void main(String args[]){
 Demo d1 = new Demo(1);
 Demo d2 = new Demo(2);
 Demo d3 = new Demo(3);
 Demo d4 = new Demo(10);
 }
}

```



```

101
public class Demo2
{

```

```

 • int i;
 • Demo2()
 {
 System.out.println("Demo2");
 }
 • Demo2(int i)
 {
 this.i = i;
 System.out.println(i);
 }
 • public static void main(String args[])
 {
 new Demo2(1);
 }
}

```

if java 4  
 error : this()

this → refers to the current class object  
 this() → Empty argument based call

