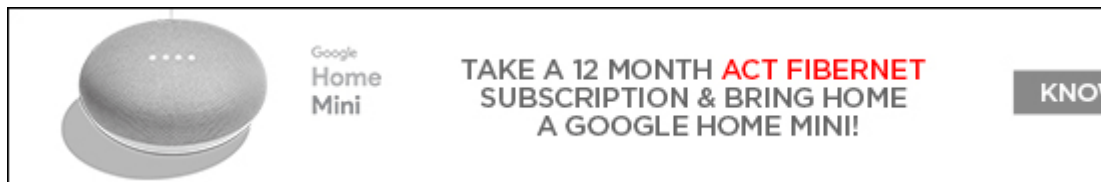




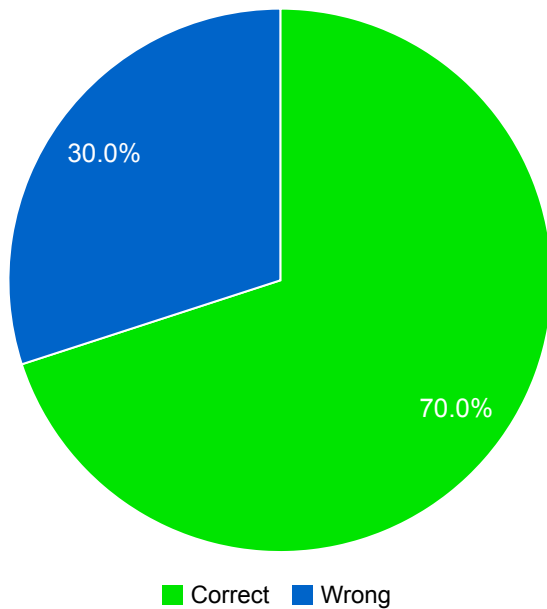
(/index.jsp)



Result

Descriptions	Status
Attempted Question	20
Un-Attempted Question	0
Total Correct Answer	14
Total Wrong Answer	6
Result	Pass
Performance	Good





Ques 1 : What will be printed as the result of the operation below.

```
main()
{
    int x=20,y=35;
    x=y++ + x++;
    y= ++y + ++x;
    printf("%d%d",x,y);
}
```

- (A) 5 8 9 4
 - ✓(B) 5 7 9 4
 - (C) 5 8 9 5
 - (D) 5 7 8 4
- Answer :** 5 7 9 4

Ques 2 : What will be printed as the result of the operation below:

```
main()
{
    int x=5;
    printf("%d,%d,%d",x,x<<2,x>>2);
}
```

- (A) 5,21,1
 - ✓(B) 5,20,1
 - (C) 5,19,0
 - (D) 5,19,1
- Answer :** 5,20,1

Ques 3 : Given the following program fragment

```
main ()
{
    int i, j, k;
    i = 3;
    j = 2*(i++);
    k = 2*(++i);
}
```

which one of the given option is correct?

(A) j = 6, k = 10.

(B) i = 5, k = 6.

✗(C) j = 6, k = 8.

(D) i = 4, j = 6.

Answer : j = 6, k = 10.

Description :

Explanation : In the expression `j = 2 * (i++)` the value of `i` is used before incrementing and in expression `k = 2*(++i);` will get incremented first and then used in the expression

Ques 4 : How many times the below loop will run

```
main()
{
    int i;
    i=0;
    do
    {
        --i;
        printf("%d",i);
        i++;
    }
    while(i>=0);
}
```

(A) 1

✓(B) Infinite

(C) 0

(D) Compilation Error

Answer : Infinite

Description :

In every iteration value of `i` is decremented and then incremented so remains 0 and hence a Infinite Loop

Ques 5 : switch(option)

```
{
    case 'H' : printf("Hello");
    case 'W' : printf("Welcome");
    case 'B' : printf("Bye");
    break;
}
```

what would be the output if option = 'H' ?

- (A) Hello
- (B) Hello Welcome
- ✓(C) Hello Welcome Bye
- (D) None of the above

Answer : Hello Welcome Bye

Description :

If option = H then the first case is true so "Hello" gets printed but there is no break statement after this case to come out of the switch statement so the program execute all other case statements also and Hello Welcome Bye get printed.

Ques 6 : Suppose a,b,c are integer variables with values 5,6,7 respectively. What is the value of the expression:

$!((b+c)>(a+10))$

- ✓(A) 1
- (B) 6
- (C) 15
- (D) 0

Answer : 1

Description :

1. $!((b+c)>(a+10))$
2. $!((6 + 7) > (5+10))$
3. $!(13 > 15)$ 13 is less than 15 so it will return False (0).
4. $!(0)$. Not of 0 is 1.

Ques 7 : Consider the following program,

```
main ()
{
    int i, j;
    for (i=0, j=5; j >0, i < 10; i ++, j--)
        printf("pskills.org");
}
```

How many times "pskills.org" will get printed

- (A) 5
- (B) Compilation Error
- ✓(C) 10

(D) None of the above

Answer : 10

Description :

Condition part of for loop (j>0, i<10) is separated by commas which means that compiler will use the value which is at right hand side of comma i.e of i<10 so the loop will execute till the value of i is less than 10.

Ques 8 : What value of c will get printed

```
main()
{
    int a,b,c;
    a=10;
    b=20;
    c=printf("%d",a)+ ++b;
    printf("\n%d",c);
}
```

✓(A) 23

(B) 22

(C) 30

(D) Compilation Error

Answer : 23

Description :

printf() will return no. of bytes it printed

Expression becomes

c = 2 + ++b;

then value of b is incremented before addition

Ques 9 : How many times the below loop will get executed?

```
main()
{
    int i;
    for(i=9;i;i=i-2)
    {
        printf("\n%d",i);
    }
}
```

✗(A) 5

(B) 6

(C) Compilation Error

(D) Infinite

Answer : Infinite

Description :

Above loop will iterate till i have non zero value. Initial value of i is 9 and it get decremented by 2 in every iteration (9, 7, 5, 3, 1, -1, -3, -5, -7, -9, -11, -13, -15, -17, -19, -21, -23, -25, -27, -29, -31, -33, -35, -37, -39, -41, -43, -45, -47, -49, -51, -53, -55, -57, -59, -61, -63, -65, -67, -69, -71, -73, -75, -77, -79, -81, -83, -85, -87, -89, -91, -93, -95, -97, -99, -101, -103, -105, -107, -109, -111, -113, -115, -117, -119, -121, -123, -125, -127, -129, -131, -133, -135, -137, -139, -141, -143, -145, -147, -149, -151, -153, -155, -157, -159, -161, -163, -165, -167, -169, -171, -173, -175, -177, -179, -181, -183, -185, -187, -189, -191, -193, -195, -197, -199, -201, -203, -205, -207, -209, -211, -213, -215, -217, -219, -221, -223, -225, -227, -229, -231, -233, -235, -237, -239, -241, -243, -245, -247, -249, -251, -253, -255, -257, -259, -261, -263, -265, -267, -269, -271, -273, -275, -277, -279, -281, -283, -285, -287, -289, -291, -293, -295, -297, -299, -301, -303, -305, -307, -309, -311, -313, -315, -317, -319, -321, -323, -325, -327, -329, -331, -333, -335, -337, -339, -341, -343, -345, -347, -349, -351, -353, -355, -357, -359, -361, -363, -365, -367, -369, -371, -373, -375, -377, -379, -381, -383, -385, -387, -389, -391, -393, -395, -397, -399, -401, -403, -405, -407, -409, -411, -413, -415, -417, -419, -421, -423, -425, -427, -429, -431, -433, -435, -437, -439, -441, -443, -445, -447, -449, -451, -453, -455, -457, -459, -461, -463, -465, -467, -469, -471, -473, -475, -477, -479, -481, -483, -485, -487, -489, -491, -493, -495, -497, -499, -501, -503, -505, -507, -509, -511, -513, -515, -517, -519, -521, -523, -525, -527, -529, -531, -533, -535, -537, -539, -541, -543, -545, -547, -549, -551, -553, -555, -557, -559, -561, -563, -565, -567, -569, -571, -573, -575, -577, -579, -581, -583, -585, -587, -589, -591, -593, -595, -597, -599, -601, -603, -605, -607, -609, -611, -613, -615, -617, -619, -621, -623, -625, -627, -629, -631, -633, -635, -637, -639, -641, -643, -645, -647, -649, -651, -653, -655, -657, -659, -661, -663, -665, -667, -669, -671, -673, -675, -677, -679, -681, -683, -685, -687, -689, -691, -693, -695, -697, -699, -701, -703, -705, -707, -709, -711, -713, -715, -717, -719, -721, -723, -725, -727, -729, -731, -733, -735, -737, -739, -741, -743, -745, -747, -749, -751, -753, -755, -757, -759, -761, -763, -765, -767, -769, -771, -773, -775, -777, -779, -781, -783, -785, -787, -789, -791, -793, -795, -797, -799, -801, -803, -805, -807, -809, -811, -813, -815, -817, -819, -821, -823, -825, -827, -829, -831, -833, -835, -837, -839, -841, -843, -845, -847, -849, -851, -853, -855, -857, -859, -861, -863, -865, -867, -869, -871, -873, -875, -877, -879, -881, -883, -885, -887, -889, -891, -893, -895, -897, -899, -901, -903, -905, -907, -909, -911, -913, -915, -917, -919, -921, -923, -925, -927, -929, -931, -933, -935, -937, -939, -941, -943, -945, -947, -949, -951, -953, -955, -957, -959, -961, -963, -965, -967, -969, -971, -973, -975, -977, -979, -981, -983, -985, -987, -989, -991, -993, -995, -997, -999, -1001, -1003, -1005, -1007, -1009, -1011, -1013, -1015, -1017, -1019, -1021, -1023, -1025, -1027, -1029, -1031, -1033, -1035, -1037, -1039, -1041, -1043, -1045, -1047, -1049, -1051, -1053, -1055, -1057, -1059, -1061, -1063, -1065, -1067, -1069, -1071, -1073, -1075, -1077, -1079, -1081, -1083, -1085, -1087, -1089, -1091, -1093, -1095, -1097, -1099, -1101, -1103, -1105, -1107, -1109, -1111, -1113, -1115, -1117, -1119, -1121, -1123, -1125, -1127, -1129, -1131, -1133, -1135, -1137, -1139, -1141, -1143, -1145, -1147, -1149, -1151, -1153, -1155, -1157, -1159, -1161, -1163, -1165, -1167, -1169, -1171, -1173, -1175, -1177, -1179, -1181, -1183, -1185, -1187, -1189, -1191, -1193, -1195, -1197, -1199, -1201, -1203, -1205, -1207, -1209, -1211, -1213, -1215, -1217, -1219, -1221, -1223, -1225, -1227, -1229, -1231, -1233, -1235, -1237, -1239, -1241, -1243, -1245, -1247, -1249, -1251, -1253, -1255, -1257, -1259, -1261, -1263, -1265, -1267, -1269, -1271, -1273, -1275, -1277, -1279, -1281, -1283, -1285, -1287, -1289, -1291, -1293, -1295, -1297, -1299, -1301, -1303, -1305, -1307, -1309, -1311, -1313, -1315, -1317, -1319, -1321, -1323, -1325, -1327, -1329, -1331, -1333, -1335, -1337, -1339, -1341, -1343, -1345, -1347, -1349, -1351, -1353, -1355, -1357, -1359, -1361, -1363, -1365, -1367, -1369, -1371, -1373, -1375, -1377, -1379, -1381, -1383, -1385, -1387, -1389, -1391, -1393, -1395, -1397, -1399, -1401, -1403, -1405, -1407, -1409, -1411, -1413, -1415, -1417, -1419, -1421, -1423, -1425, -1427, -1429, -1431, -1433, -1435, -1437, -1439, -1441, -1443, -1445, -1447, -1449, -1451, -1453, -1455, -1457, -1459, -1461, -1463, -1465, -1467, -1469, -1471, -1473, -1475, -1477, -1479, -1481, -1483, -1485, -1487, -1489, -1491, -1493, -1495, -1497, -1499, -1501, -1503, -1505, -1507, -1509, -1511, -1513, -1515, -1517, -1519, -1521, -1523, -1525, -1527, -1529, -1531, -1533, -1535, -1537, -1539, -1541, -1543, -1545, -1547, -1549, -1551, -1553, -1555, -1557, -1559, -1561, -1563, -1565, -1567, -1569, -1571, -1573, -1575, -1577, -1579, -1581, -1583, -1585, -1587, -1589, -1591, -1593, -1595, -1597, -1599, -1601, -1603, -1605, -1607, -1609, -1611, -1613, -1615, -1617, -1619, -1621, -1623, -1625, -1627, -1629, -1631, -1633, -1635, -1637, -1639, -1641, -1643, -1645, -1647, -1649, -1651, -1653, -1655, -1657, -1659, -1661, -1663, -1665, -1667, -1669, -1671, -1673, -1675, -1677, -1679, -1681, -1683, -1685, -1687, -1689, -1691, -1693, -1695, -1697, -1699, -1701, -1703, -1705, -1707, -1709, -1711, -1713, -1715, -1717, -1719, -1721, -1723, -1725, -1727, -1729, -1731, -1733, -1735, -1737, -1739, -1741, -1743, -1745, -1747, -1749, -1751, -1753, -1755, -1757, -1759, -1761, -1763, -1765, -1767, -1769, -1771, -1773, -1775, -1777, -1779, -1781, -1783, -1785, -1787, -1789, -1791, -1793, -1795, -1797, -1799, -1801, -1803, -1805, -1807, -1809, -1811, -1813, -1815, -1817, -1819, -1821, -1823, -1825, -1827, -1829, -1831, -1833, -1835, -1837, -1839, -1841, -1843, -1845, -1847, -1849, -1851, -1853, -1855, -1857, -1859, -1861, -1863, -1865, -1867, -1869, -1871, -1873, -1875, -1877, -1879, -1881, -1883, -1885, -1887, -1889, -1891, -1893, -1895, -1897, -1899, -1901, -1903, -1905, -1907, -1909, -1911, -1913, -1915, -1917, -1919, -1921, -1923, -1925, -1927, -1929, -1931, -1933, -1935, -1937, -1939, -1941, -1943, -1945, -1947, -1949, -1951, -1953, -1955, -1957, -1959, -1961, -1963, -1965, -1967, -1969, -1971, -1973, -1975, -1977, -1979, -1981, -1983, -1985, -1987, -1989, -1991, -1993, -1995, -1997, -1999, -2001, -2003, -2005, -2007, -2009, -2011, -2013, -2015, -2017, -2019, -2021, -2023, -2025, -2027, -2029, -2031, -2033, -2035, -2037, -2039, -2041, -2043, -2045, -2047, -2049, -2051, -2053, -2055, -2057, -2059, -2061, -2063, -2065, -2067, -2069, -2071, -2073, -2075, -2077, -2079, -2081, -2083, -2085, -2087, -2089, -2091, -2093, -2095, -2097, -2099, -2101, -2103, -2105, -2107, -2109, -2111, -2113, -2115, -2117, -2119, -2121, -2123, -2125, -2127, -2129, -2131, -2133, -2135, -2137, -2139, -2141, -2143, -2145, -2147, -2149, -2151, -2153, -2155, -2157, -2159, -2161, -2163, -2165, -2167, -2169, -2171, -2173, -2175, -2177, -2179, -2181, -2183, -2185, -2187, -2189, -2191, -2193, -2195, -2197, -2199, -2201, -2203, -2205, -2207, -2209, -2211, -2213, -2215, -2217, -2219, -2221, -2223, -2225, -2227, -2229, -2231, -2233, -2235, -2237, -2239, -2241, -2243, -2245, -2247, -2249, -2251, -2253, -2255, -2257, -2259, -2261, -2263, -2265, -2267, -2269, -2271, -2273, -2275, -2277, -2279, -2281, -2283, -2285, -2287, -2289, -2291, -2293, -2295, -2297, -2299, -2301, -2303, -2305, -2307, -2309, -2311, -2313, -2315, -2317, -2319, -2321, -2323, -2325, -2327, -2329, -2331, -2333, -2335, -2337, -2339, -2341, -2343, -2345, -2347, -2349, -2351, -2353, -2355, -2357, -2359, -2361, -2363, -2365, -2367, -2369, -2371, -2373, -2375, -2377, -2379, -2381, -2383, -2385, -2387, -2389, -2391, -2393, -2395, -2397, -2399, -2401, -2403, -2405, -2407, -2409, -2411, -2413, -2415, -2417, -2419, -2421, -2423, -2425, -2427, -2429, -2431, -2433, -2435, -2437, -2439, -2441, -2443, -2445, -2447, -2449, -2451, -2453, -2455, -2457, -2459, -2461, -2463, -2465, -2467, -2469, -2471, -2473, -2475, -2477, -2479, -2481, -2483, -2485, -2487, -2489, -2491, -2493, -2495, -2497, -2499, -2501, -2503, -2505, -2507, -2509, -2511, -2513, -2515, -2517, -2519, -2521, -2523, -2525, -2527, -2529, -2531, -2533, -2535, -2537, -2539, -2541, -2543, -2545, -2547, -2549, -2551, -2553, -2555, -2557, -2559, -2561, -2563, -2565, -2567, -2569, -2571, -2573, -2575, -2577, -2579, -2581, -2583, -2585, -2587, -2589, -2591, -2593, -2595, -2597, -2599, -2601, -2603, -2605, -2607, -2609, -2611, -2613, -2615, -2617, -2619, -2621, -2623, -2625, -2627, -2629, -2631, -2633, -2635, -2637, -2639, -2641, -2643, -2645, -2647, -2649, -2651, -2653, -2655, -2657, -2659, -2661, -2663, -2665, -2667, -2669, -2671, -2673, -2675, -2677, -2679, -2681, -2683, -2685, -2687, -2689, -2691, -2693, -2695, -2697, -2699, -2701, -2703, -2705, -2707, -2709, -2711, -2713, -2715, -2717, -2719, -2721, -2723, -2725, -2727, -2729, -2731, -2733, -2735, -2737, -2739, -2741, -2743, -2745, -2747, -2749, -2751, -2753, -2755, -2757, -2759, -2761, -2763, -2765, -2767, -2769, -2771, -2773, -2775, -2777, -2779, -2781, -2783, -2785, -2787, -2789, -2791, -2793, -2795, -2797, -2799, -2801, -2803, -2805, -2807, -2809, -2811, -2813, -2815, -2817, -2819, -2821, -2823, -2825, -2827, -2829, -2831, -2833, -2835, -2837, -2839, -2841, -2843, -2845, -2847, -2849, -2851, -2853, -2855, -2857, -2859, -2861, -2863, -2865, -2867, -2869, -2871, -2873, -2875, -2877, -2879, -2881, -2883, -2885, -2887, -2889, -2891, -2893, -2895, -2897, -2899, -2901, -2903, -2905, -2907, -2909, -2911, -2913, -2915, -2917, -2919, -2921, -2923, -2925, -2927, -2929, -2931, -2933, -2935, -2937, -2939, -2941, -2943, -2945, -2947, -2949, -2951, -2953, -2955, -2957, -2959, -2961, -2963, -2965, -2967, -2969, -2971, -2973, -2975, -2977, -2979, -2981, -2983, -2985, -2987, -2989, -2991, -2993, -2995, -2997, -2999, -3001, -3003, -3005, -3007, -3009, -3011, -3013, -3015, -3017, -3019, -3021, -3023, -3025, -3027, -3029, -3031, -3033, -3035, -3037, -3039, -3041, -3043, -3045, -3047, -3049, -3051, -3053, -3055, -3057, -3059, -3061, -3063, -3065, -3067, -3069, -3071, -3073, -3075, -3077, -3079, -3081, -3083, -3085, -3087, -3089, -3091, -3093, -3095, -3097, -3099, -3101, -3103, -3105, -3107, -3109, -3111, -3113, -3115, -3117, -3119, -3121, -3123, -3125, -3127, -3129, -3131, -3133, -3135, -3137, -3139, -3141, -3143, -3145, -3147, -3149, -3151, -3153, -3155, -3157, -3159, -3161, -3163, -3165, -3167, -3169, -3171, -3173, -3175, -3177, -3179, -3181, -3183, -3185, -3187, -3189, -3191, -3193, -3195, -3197, -3199, -3201, -3203, -3205, -3207, -3209, -3211, -3213, -3215, -3217, -3219, -3221, -3223, -3225, -3227, -3229, -3231, -3233, -3235, -3237, -3239, -3241, -3243, -3245, -3247, -3249, -3251, -3253, -3255, -3257, -3259, -3261, -3263, -3265, -3267, -3269, -3271, -3273, -3275, -3277, -3279, -3281, -3283, -3285, -3287, -3289, -3291, -3293, -3295, -3297, -3299, -3301, -3303, -3305, -3307, -3309, -3311, -3313, -3315, -3317, -3319, -3321, -3323, -3325, -3327, -3329, -3331, -3333, -3335, -3337, -3339, -3341, -3343, -3345, -3347, -3349, -3351, -3353, -3355, -3357, -3359, -3361, -3363, -3365, -3367, -3369, -3371, -3373, -3375, -3377, -3379, -3381, -3383, -3385, -3387, -3389, -3391, -3393, -3395, -3397, -3399, -3401, -3403, -3405, -3407, -3409, -3411, -3413, -3415, -3417, -3419, -3421, -3423, -3425, -3427, -3429, -3431, -3433, -3435, -3437, -3439, -3441, -3443, -3445, -3447, -3449, -3451, -3453, -3455, -3457, -3459, -3461, -3463, -3465, -3467, -3469, -3471, -3473, -3475, -3477, -3479, -3481, -3483, -3485, -3487, -3489, -3491, -3493, -3495, -3497, -3499, -3501, -3503, -3505, -3507, -3509, -3511, -3513, -3515, -3517, -3519, -3521, -3523, -3525, -3527, -3529, -3531, -3533, -3535, -3537, -3539, -3541, -3543, -3545, -3547, -3549, -3551, -3553, -3555, -3557, -3559, -3561, -3563, -3565, -3567, -3569, -3571, -3573, -3575, -3577, -3579, -3581, -3583, -3585, -3587, -3589, -3591, -3593, -3595, -3597, -3599, -3601, -3603, -3605, -3607, -3609, -3611, -3613, -3615, -3617, -3619, -3621, -3623, -3625, -3627, -3629, -3631, -3633, -3635, -3637, -3639, -3641, -3643, -3645, -3647, -3649, -3651, -3653, -3655, -3657, -3659, -3661, -3663, -3665, -3667, -3669, -3671, -3673, -3675, -3677, -3679, -3681, -3683, -3685, -3687, -3689, -3691, -3693, -3695, -3697, -3699, -3701, -3703, -3705, -3707, -3709, -3711, -3713, -3715, -3717, -3719, -3721, -3723, -3725, -3727, -3729, -3731, -3733, -3735, -3737, -3739, -3741, -3743, -3745, -3747, -3749, -3751, -3753, -3755, -3757, -3759, -3761, -3763, -3765, -3767, -3769, -3771, -3773, -3775, -3777, -3779, -3781, -3783, -3785, -3787, -3789, -3791, -3793, -3795, -3797, -3799, -3801, -3803, -3805, -3807, -3809, -3811, -3813, -3815, -3817, -3819, -3821, -3823, -3825, -3827, -3829, -3831, -3833, -3835, -3837, -3839, -3841, -3843, -3845, -3847, -3849, -3851, -3853, -3855, -3857, -3859, -3861, -3863, -3865, -3867, -3869, -3871, -3873, -3875, -38

Ques 10 : What the below statement will print if a=10

```
printf("%d %d",a, !a++);
```

✓(A) 11 0

(B) 10 10

(C) 10 0

(D) 0 10

Answer : 11 0

Description :

Values in the function get passed from right to left. First !a++ get processed which pass zero as argument and make a equal to 11

Ques 11 : What will be the output

```
main()
```

```
{
    int i;
    i = 10;
    if(i == 20 || 30)
    {
        printf("True");
    }
    else
    {
        printf("False");
    }
}
```

✓(A) True

(B) False

(C) Syntax Error

(D) Run time Error

Answer : True

Description :

i==20 is a expression which will return TRUE or FALSE depending on the value of i. In this program it will return 0 so the statement become

If (0 || 30)

30 is a nonzero value which means TRUE (1) in C when ORed with 0 will result TRUE.

Ques 12 : What will be the output

```
main()
{
    if(1,0)
    {
        printf("True");
    }
    else
    {
        printf("False");
    }
}
```

- (A) True
- ✓(B) False
- (C) Compilation Error
- (D) Run time Error

Answer : False

Description :

comma(,) operator returns the value which at the right hand side of , . if statement become if(0)

Ques 13 : What will be the output

```
main()
{
    int i, j, *ptr, *ptr1;
    i = 10;
    j = 10;
    ptr = &i;
    ptr1 = &j;
    if(ptr == ptr1)
    {
        printf("True");
    }
    else
    {
        printf("False");
    }
}
```

- (A) True
- ✓(B) False
- (C) Syntax Error
- (D) Run time Error

Answer : False

Description :

In this program we are comparing the addresses contained by ptr & ptr1 not the value at those addresses and pointers ptr and ptr1 have the addresses of different variables so above condition is false

Ques 14 : How many times the below loop will get executed?

```
main()
{
    int i;
    for(i=20, i=10; i<=20; i++)
    {
        printf("\n %d", i);
    }
}
```

- (A) 1
- (B) Run time Error
- ✓(C) 11
- (D) Compilation Error

Answer : 11

Description :

i will start from 10

Ques 15 : How many times main() will get called?

```
main()
{
    printf("\n Main Called Again");
    main();
}
```

- (A) 1
- (B) 100
- ✗(C) main can not be called recursively
- (D) Infinite

Answer : Infinite

Description :

There is no condition in the main() to stop the recursive calling of the main() hence it will be called infinite no of times.

Ques 16 : What will be the output of the following program if the base address of array is 100.

```
main()
{
    int gyan[] = { 1,2,3,4,5 };
    int i,*ptr ;
    ptr = gyan;
    for(i = 0; i<=4 ; i++)
    {
        printf("\n %d", *ptr++);
    }
}
```


- (A) 1 2 3 4 5
✗(B) 2 3 4 5
(C) 100 101 102 103
(D) 101 102 103 104
Answer : 1 2 3 4 5

Description :

ptr contains the base address of the array and printf() is printing the value at the current address contained by ptr and then incrementing the pointer to point to the next array element address.

Ques 17 : Which of the following program structure/component/statement is not an example for implementation of modularization ?

- ✗(A)** DLL
(B) Functions
(C) type casting
Answer : type casting

Description :

Option c) type casting. DLL and Functions help in modularization of a program while typecasting just converts from one data type to another.

Ques 18 : What will be output of following program?

```
#include<stdio.h>
int main()
{
    int a = 320;
    char *ptr;
    ptr = (char *)&a;
    printf("%d", *ptr);
    return 0;
}
```

- (A) 2
(B) 320
✓(C) 64
(D) Compilation Error

Answer : 64

Description :

As we know int is two byte data type while char is one byte data type.

Character pointer can keep the address one byte at a time.

Binary value of 320 is 00000001 01000000 (In 16 bit) Memory

representation of int a = 320 is:

So ptr is pointing only first 8 bit which value is 64 and Decimal value is 64.

Ques 19 : What will be output when you will execute following c code?

```
#include<stdio.h>
{
    char arr[11]="The African Queen";
    printf("%s", arr);
    return 0;
}
```

Choose all that

- (A) The African
- ☒ (B) The
- (C) Compilation
- (D) None of the above

Answer : Compilation

Description :

Size of any character array cannot be less than the number of characters in any string which it has assigned. Size of an array can be equal (excluding null character) or reater than but never less than

Ques 20 : What will be output of the following c program?

```
#include "stdio.h"
int main()
{
    int _ = 5;
    int __ = 10;
    int ___;
    ___ = _ + __;
    printf("%i", ___);
    return 0;
}
```

- (A) 5
- (B) 10
- ☒ (C) 15
- (D) Compilation Error

Answer : 15

Description :

Variable name can have only underscore

Save Your Result

Tutorials

Core Java Tutorial (<http://www.pskills.in/java/index.jsp>)

Jdbc Tutorial (<http://www.pskills.in/jdbc/index.jsp>)

Servlet Tutorial (<http://www.pskills.in/servlet/index.jsp>)

Jsp Tutorial (<http://www.pskills.in/jsp/index.jsp>)

Interview Question

Java Interview (</javainterview.jsp>)

C Interview (</cinterview.jsp>)

Php Interview (</phpinterview.jsp>)

Html Interview (</htmlinterview.jsp>)

.Net Interview (</netinterview.jsp>)

HR Interview (</hrinterview.jsp>)

Online Test

Core Java Test (</corejava.jsp>)

Jdbc Test (</jdbc.jsp>)

Servlet Test (</servlet.jsp>)

Jsp Test (</jsp.jsp>)

Php Test (</php.jsp>)

C Test (</c.jsp>)

C# Test (</csharp.jsp>)

Asp Test (</asp.jsp>)

Html Test (</html.jsp>)

Javascript & JQuery Test (</jquery.jsp>)

Aptitude Test (</aptitude.jsp>)

Other Links

Get Test Result (</certificatesearch.jsp>)

Download Software (</downloadsoftware.jsp>)

Download Books (</viewFiles.jsp>)

Java Posts (</blog.jsp>)

Faq's (</faq.jsp>)

Interview Questions (</interview.jsp>)

About Us (</aboutus.jsp>)

© Copyright 2015. All Rights Reserved. Contact (</contactus.jsp>)