

Introduction to Programming – Surprise Test

Name:

Roll Number:

+++++

Select the correct o/p of the following programs (without execution) and state reasons:

1.

```
int Trump = 'R', Obama = 'D', Biden = 'D';
Trump = Obama == Biden;
printf ("%d", Trump);
```

i) 68, ii) 1, iii) 0, iv) 82.
2.

```
int counter = 101;
for (counter--;counter--;counter--);
printf ("%d",counter);
```

i) 0, ii) 1, iii) -1, iv) None.
3.

```
printf ("ISI" "MTech" "%d", printf ("to"));
```

i) ISI, ii) ISIMTech, iii) ISIMTech2, iv) toISIMTech2.
4.

```
int no = 2, *ptr = &no;
no *= 3;
printf ("%d", *ptr**ptr);
```

i) 36, ii) 6, iii) 4, iv) 12.
5.

```
int x = 3, y, z;
z = y = x;
z *= y = x*x;
```

```
printf ("%d %d %d", x, y, z);
```

i) 3 9 9, ii) 3 3 3, iii) 3 9 27, iv) None.
6.

```
printf ("%d",10++);
```

i) 1, ii) 10, iii) 11, iv) Error.
7.

```
#define triple(x) 3*x
...
int m = 6, n = 6;
m = m / triple(2);
n /= triple(2);
printf ("%d %d", m, n);
```

i) 1 1, ii) 4 1, iii) 1 4, iv) 4 4.
8.

```
int i = printf ("MTech") > printf ("ISI");
printf ("%d", i);
```

i) 1, ii) 0, iii) MTechISI1, iv) Error.
9.

```
for (;;)
printf ("Programming");
```

i) Error, ii) Warning & ∞ loop, iii) Programming, iv) None.
10.

```
int a[] = {-5, -3, -1, 1, 3, 5}, *b = &a[2];
printf ("%d", b["%d",a[2]]);
```

i) -3, ii) -3-1, iii) -1, iv) Error.
11.

```
int a[] = {10, 20, 30, 40, 50};
int *p1 = &a[1], *p2 = &a[4];
printf ("%d", (p2-p1));
```

i) 30, ii) 6, iii) 3, iv) 4.

12. `printf (printf ("li")+ "mili");`
 i) lili, ii) limili, iii) mili, iv) 2mili.
13. `char *a = "yes""no""0""ok";`
`printf ("%s", a);`
 i) yesnook, ii) yesno, iii) yes, iv) Error.
14. `printf ("%c", "don"[1]);`
 i) d, ii) o, iii) n, iv) odon.
15. `void a = 10;`
`printf ("%d", a);`
 i) Error, ii) 10, iii) 0, iv) None.
16. `char ch = 291;`
`printf ("%d", ch);`
 i) 291, ii) 127, iii) 35, iv) None.
17. `printf ("%d", sizeof (234L));`
 i) 2, ii) 4, iii) 8, iv) Compiler-fixed.
18. `float problem = 0.8;`
`printf ("%0.8f",problem);`
 i) 0.80000000, ii) 0.79999999, iii) 0.80000001, iv) None.
19. `fprintf (stdout, "%i", 48);`
 i) Error, ii) '0', iii) 0, iv) 48.
20. `printf("by""%d",printf("Two"));`
 i) by3Two, ii) byTwo, iii) Twoby3, iv) Error.
21. `# define x - -5`
`int main(){`
`printf ("%d", x);`
`return 0;`
`}`
 i) -5, ii) 5, iii) 4, iv) Error.
22. `#define sqr(n) ++n * ++n`
`int main(){`
`int x = 3, y;`
`y = ++x * ++x;`
`x -= 3;`
`printf ("%d %d", sqr(x), y);`
`return 0;`
`}`
 i) 16 16, ii) 25 25, iii) 16 25, iv) 25 16.
23. `char subject[20] = "Intro" "2" "Program";`
`printf (subject);`
 i) Intro, ii) Program, iii) Intro 2 Program, iv) Intro2Program.
24. `char name[20];`
`scanf ("%s %s", name); // Let the input be "Tim Berners-Lee"`
`printf ("%s", name);`
 i) Tim, ii) Berners-Lee, iii) Tim Berners-Lee, iv) Lee.
25. `#define fun(x) (sizeof (x) / sizeof (*x))`
`int main(){`
`int digit[10] = {0,1,2,3,4,5, 6, 7, 8, 9};`

- ```

char *str[20] = {"aa", "ab", "ac", "ad", "ae", "af", "ag", "ah", "ai", "aj"};
printf ("%d %d\n", fun(digit), fun(str));
return 0;
}

```
26. i) 1 1,                      ii) 10 10,                      iii) 10 20,                      iv) None.
26.    `int i = 2^3;`  
       `printf ("%d", i);`  
       i) 0,                      ii) 1,                      iii) 8,                      iv) 9.
27.    `if(!--1)`  
       `printf ("Santa");`  
       `if((~7 & 0x000f) == 8)`  
       `printf ("Banta");`  
       `printf ("**");`  
       i) Santa\*,                      ii) Banta\*,                      iii) SantaBanta\*,                      iv) \*.
28.    `int a = 2;`  
       `if(a >> 1)`  
       `printf ("Value = %d", a);`  
       i) Value = 0,                      ii) Value = 1,                      iii) Value = 2,                      iv) Value =.
29.    `int a = 5, b = 10;`  
       `a ^= b ^= a ^= b;`  
       `printf ("%d %d", a, b);`  
       i) 5 10,                      ii) 10 5,                      iii) 5 5,                      iv) 10 10.
30.    `int i = 2, j = 2;`  
       `while(i+1 ? --j : j++)`  
       `printf ("%d", i);`  
       i) 1,                      ii) 2,                      iii) 3,                      iv) Error.
31.    `int x = 3, y = 5;`  
       `printf ("%d", (x+y)++);`  
       i) 8,                      ii) 9,                      iii) 10,                      iv) Error.
32.    `printf ("Brouhaha"+21/4-1);`  
       i) haha,                      ii) a,                      iii) Brouhaha,                      iv) Error.
33.    `int var[][3] = {1, 2, 3, 4, 5, 6};`  
       `int (*ptr)[3] = var;`  
       `printf ("%d %d ", (*ptr)[1], (*ptr)[2]);`  
       `++ptr;`  
       `printf ("%d %d", (*ptr)[1], (*ptr)[2]);`  
       i) 2 3 3 4,                      ii) 2 3 4 5,                      iii) 2 3 5 6,                      iv) Error.
34.    `int num = 16;`  
       `if((num & (num - 1)) == 0)`  
       `printf ("Yes");`  
       else  
       `printf ("No");`  
       i) Yes,                      ii) No,                      iii) YesNo,                      iv) Error.
35.    `int a = 3, b = 8;`  
       `(a & b) ? printf("True") : printf("False");`  
       i) True,                      ii) False,                      iii) Error,                      iv) None.
36.    `long int i = 1; // A long integer consumes 8 bytes of memory`  
       `printf ("%lu", -sizeof(i));`  
       i) 8,                      ii) -8,                      iii) -1,                      iv) None.

37. `int (*ptr)[20][30];  
printf ("%lu", sizeof(*ptr)); // An integer consumes 4 bytes of memory`  
 i) 4,                                   ii) 600,                                   iii) 2400,                                   iv) None.
38. `printf ("%d", printf ("%d", printf ("%d", printf ("%s", "CisSOfunny"))));`  
 i) CisSOfunny,                                   ii) 1210 CisSOfunny,                                   iii) CisSOfunny1021,                                   iv) Error.
39. `static int a[] = {0, 1, 2, 3, 4};  
static int *p[] = {a, a+1, a+2, a+3, a+4};  
int **ptr = p;  
ptr++, printf ("%d", **ptr);  
*ptr++, printf ("%d", **ptr);  
++*ptr, printf ("%d", **ptr);  
**ptr, printf ("%d", **ptr);`  
 i) 1234,                                   ii) 1223,                                   iii) 1222,                                   iv) 1233.
40. `int num = 5;  
while(num --> 0)  
    printf("%d ", num);`  
 i) 5 4 3 2 1,                                   ii) 4 3 2 1,                                   iii) 4 3 2 1 0,                                   iv) Error.