1. What will be the output of the program?

```
#include<stdio.h>
int main()
{
    float a=0.7;
    if(a < 0.7)
        printf("C\n");
    else
        printf("C++\n");
    return 0;
}
```

A. C

B. C++

C. Compiler error

D. Non of above

Answer: Option **A**
Explanation:
`if(a < 0.7)` here `a` is a float variable and `0.7` is a double constant. The float variable `a` is less than double constant `0.7`. Hence the `if` condition is satisfied and it prints `'C'`
Example:

```
#include<stdio.h>
int main()
{
    float a=0.7;
    printf("%.10f %.10f\n",0.7, a);
    return 0;
}
```

Output:
0.7000000000 0.6999999881
View Answer Discuss in Forum Workspace Report

---

2. What will be the output of the program?

```
#include<stdio.h>
int main()
{
    float *p;
    printf("%d\n", sizeof(p));
    return 0;
}
```

A. 2 in 16bit compiler, 4 in 32bit compiler

B. 4 in 16bit compiler, 2 in 32bit compiler

C. 4 in 16bit compiler, 4 in 32bit compiler

D. 2 in 16bit compiler, 2 in 32bit compiler

Answer: Option **A**
Explanation:
`sizeof(x)` returns the size of `x` in bytes.
`float *p` is a pointer to a `float`.
In 16 bit compiler, the pointer size is always 2 bytes.
In 32 bit compiler, the pointer size is always 4 bytes.
View Answer Discuss in Forum Workspace Report

---

3. What will be the output of the program?

```
#include<stdio.h>
int main()
{
    float fval=7.29;
    printf("%d\n", (int)fval);
    return 0;
}
```

A. 0

B. 0.0

C. 7.0

D. 7

View Answer Discuss in Forum Workspace Report

---

4. What will be the output of the program?

```
#include<stdio.h>
#include<math.h>
int main()
{
    printf("%f\n", sqrt(36.0));
    return 0;
}
```

A. 6.0

B. 6

C. 6.000000

D. Error: Prototype `sqrt()` not found.

Answer: Option **C**
Explanation:
`printf("%f\n", sqrt(36.0));` It prints the square root of 36 in the float format(i.e 6.000000).
Declaration Syntax: `double sqrt(double x)` calculates and return the positive square root of the given number.

5. What will be the output of the program?

```
#include<stdio.h>
#include<math.h>
int main()
{
    printf("%d, %d, %d\n", sizeof(3.14f), sizeof(3.14), sizeof(3.14l));
    return 0;
}
```

A. 4, 4, 4

B. 4, 8, 8

C. 4, 8, 10

D. 4, 8, 12

Answer: Option **C**
Explanation:
`sizeof(3.14f)` here '3.14f' specifies the `float` data type. Hence size of float is 4 bytes.
`sizeof(3.14)` here '3.14' specifies the `double` data type. Hence size of float is 8 bytes.
`sizeof(3.14l)` here '3.14l' specifies the `long double` data type. Hence size of float is 10 bytes.
Note: If you run the above program in Linux platform (GCC Compiler) it will give 4, 8, 12 as output. If you run in Windows platform (TurboC Compiler) it will give 4, 8, 10 as output. Because, C is a machine dependent language.

6. What will be the output of the program?

```
#include<stdio.h>
int main()
{
    float f=43.20;
    printf("%e, ", f);
    printf("%f, ", f);
    printf("%g", f);
    return 0;
}
```

A. 4.320000e^{+01}, 43.200001, 43.2

B. 4.3, 43.22, 43.21

C. 4.3e, 43.20f, 43.00

D. Error

Answer: Option **A**
Explanation:

`printf("%e, ", f);` Here '%e' specifies the "Scientific Notation" format. So, it prints the 43.20 as 4.320000e^{+01}.
`printf("%f, ", f);` Here '%f' specifies the "Decimal Floating Point" format. So, it prints the 43.20 as 43.200001.
`printf("%g, ", f);` Here '%g' "Use the shorter of %e or %f". So, it prints the 43.20 as 43.2.
View Answer Discuss in Forum Workspace Report

7. What will be the output of the program?

```
#include<stdio.h>
int main()
{
    float a=0.7;
    if(a < 0.7f)
        printf("C\n");
    else
        printf("C++\n");
    return 0;
}
```

A. C

B. C++

C. Compiler error

D. Non of above

Answer: Option **B**
Explanation:
`if(a < 0.7f)` here `a` is a float variable and `0.7f` is a float constant. The float variable `a` is not less than `0.7f` float constant. But both are equal. Hence the `if` condition is failed and it goes to `else` it prints `'C++'`
Example:

```
#include<stdio.h>
int main()
{
    float a=0.7;
    printf("%.10f %.10f\n",0.7f, a);
    return 0;
}
```

Output:
```
0.6999999881 0.6999999881
```
View Answer Discuss in Forum Workspace Report

8. What will be the output of the program?

```
#include<stdio.h>
#include<math.h>
int main()
{
    float n=1.54;
    printf("%f, %f\n", ceil(n), floor(n));
    return 0;
}
```

A. 2.000000, 1.000000

B. 1.500000, 1.500000

C. 1.550000, 2.000000

D. 1.000000, 2.000000

Answer: Option **A**
Explanation:
`ceil(x)` round up the given value. It finds the smallest integer not < `x`.
`floor(x)` round down the given value. It finds the smallest integer not > `x`.
`printf("%f, %f\n", ceil(n), floor(n));` In this line `ceil(1.54)` round up the 1.54 to 2 and `floor(1.54)` round down the 1.54 to 1.
In the `printf("%f, %f\n", ceil(n), floor(n));` statement, the format specifier "%f %f" tells output to be float value. Hence it prints 2.000000 and 1.000000.
View Answer Discuss in Forum Workspace Report

9. What will be the output of the program?

```
#include<stdio.h>
int main()
{
    float d=2.25;
    printf("%e,", d);
    printf("%f,", d);
    printf("%g,", d);
    printf("%lf", d);
    return 0;
}
```

A. 2.2, 2.50, 2.50, 2.5

B. 2.2e, 2.25f, 2.00, 2.25

C. 2.250000e+000, 2.250000, 2.25, 2.250000

D. Error

Answer: Option **C**
Explanation: