

Chap 4

FLOATING POINT ISSUES

1. O/P?

```
Main()
{
    float a=0.7;
    if(a<0.7)
        printf("C");
    else
        printf("C++");
}
```

a. C
b. C++
c. Error
d. None of above.

Ans: A.

2.O/p?

```
main()
{
    float a=0.7;
    if(a<0.7f)
        printf("C");
    else
        printf("C++");
}
```

a.C
b. C++
c. Error
e. None of above

Ans: B.

3.O/p?

```
main()
{
    printf("%f",sqrt(36.0));
}
```

a.6.0
b.6
c.6.000000

d. Some absurd result

ans: D

4. Would this program give proper results <yes/no>?

```
main()
{
    printf("%f", log(36.0));
}
```

ans: No. since math.h is not included.

5. Would the following printf()s print the same values for any value of a <yes/no>?

```
main()
{
    float a;
    scanf("%f", &a);
    printf("%f", a+a+a);
    printf("%f", 3*a);
}
```

ans: No.

6. We want to round off x, a float, to an int value. The correct way to do so would be

- a. `y=(int)(x+0.5);`
- b. `y=int(x+0.5);`
- c. `y=(int) x+0.5;`
- d. `y=(int)((int)x+0.5);`

ans: A.

7. Which error are you likely to get when you run the following program.

```
Main()
{
    struct emp
    {
        char name[20];
        float sal;
    };
    struct emp e[10];
    int x;
```

```

for(x=0;x<9;x++)
    scanf("%s %f", e[x].name,e[x].sal);
}

```

- a. Suspicious pointer conversion.
- b. Floating point formats not linked
- c. Can not use scanf() for structure
- d. Strings can not be nested inside structure

Ans: B

8. What causes the error of problem 7 above to occur and how would you rectify the error in the above program.

Ans: What causes the “floating point format not linked” error to occur? When the compiler encounters a reference to the address of the float, it sets a flag to have the linker link in the floating point emulator a floating point emulator is used to manipulate floating point numbers in runtime library functions like scanf() and atof(). There are some cases in which the reference to a float is a bit obscure and the compiler does not detect the need for the emulator.

These situations usually occur during the initial stages of program development. Normally, once the program is fully developed, the emulator is used in such a fashion that the compiler can accurately determine when to link in the emulator.

To force linking of the floating point emulator into an application just include the following function in your program

```

Void LinkFloat(void)
{
    float a=0, b=&a; /* cause the emulator to be linked. */
    a=*b; /* suppress warning “var not used” */
}

```

There is no need to call this function from your program.

9. Which are the three different types of real data types available in C and what are the format specifiers used for them?

Ans:

```

Float    4 bytes  %f
Double  8 bytes  %lf
Long double 10 bytes %Lf

```

10. By default any real number is treated as

- a float
- b double
- c long double
- b depends on the memory model that you are using

ans: B.

11. What should you do to treat the constant 3.14 as a float?

Ans: Use 3.14f

12. What should you do to treat the constant 3.14 as long double?

Ans: Use 3.14l

13. O/p?

main()

```
{  
    printf(“%d %d %d”,sizeof(3.14f),sizeof(3.14),sizeof(3.14l));  
}
```

- a. 4 4 4
- b. 4 garbage value garbage value
- c. 4 8 10
- d. error

ans: C

14. The binary equivalent of 5.375 is

- a 101.101110111
- b 101.011
- c 101011
- d none of above

ans: B.

15. How floats are stored in binary form?

Ans: Floating point numbers are represented in IEEE format. The IEEE format for floating point storage uses a sign bit, a mantissa and an exponent for representing the power of two. The sign bit denotes the sign of the number (0- positive) and (1- negative) the mantissa is represented in binary after converting it into its normalized form. The normalized form results in a mantissa whose most significant bit is always 1. The IEEE format takes the advantage of this by not storing this bit at all. The exponent is an integer stored in an unsigned binary format after adding a positive integer bias. This ensures that the stored exponent is always positive. The value of bias is 127 for floats and 1023 for doubles

16. A float occupies 4 bytes. If the hexadecimal equivalent of each of these bytes is A, B, C, and D, then when this float is stored in memory these bytes get stored in the order.

- a. ABCD
- b. DCBA
- c. 0xABCD
- d. 0xDCBA

ans: B.

17. If the binary equivalent of 5.375 in normalized form is 0100 0000 1010 1100 0000 0000 0000 0000, what would be the output of the following program?

```
Main()
{
    float a=5.375;
    char *p;
    int x;
    p=(char*)&a;
    for(x=0;x<=3;x++)
        printf("%02x", (unsigned char)p[x]);
}
```

- a 40 AC 00 00
- b 04 CA 00 00
- c 00 00 AC 40
- d 00 00CA 04

ans: C.