STRUCTURES, UNIONS & ENUMERATED STRUCTURES

Exercises:

1)
```
struct student
{
        int age, rno; char nm[20];
};
void f(struct student *n)
{
        printf("Enter the details of s : ");
        scanf("%d %d %[^\n]", &n->age, &n->rno, &n->nm);
        printf("Details of s : ");
        printf("%d, %d, %s", n->age, n->rno, n->nm);
}
void main()
{
        struct student s;
        f(&s);
}
```

```
};
```

2)

```
struct student
{
    int age, rno; char nm[20];
};
void f(struct student *n)
{
    printf("Enter the details of s : ");
    scanf("%d %d %[^\n]", &n->age, &n->rno, &n->nm);
}
void main()
{
    struct student s;
    f(&s);
    printf("Details of s : ");
    printf("%d, %d, %s", s.age, s.rno, s.nm);
}
```

3)

```
union uni
{
    int a;
    char c[2];
};
```

```
void main()
{
        union uni u;
        u.c[0]=38;
        u.c[1]=2;
        printf("%d ", u.a);
}
```

Aptitude Questions:

1) 1 0.000000

2) Error! A structure is just declaration of a new datatype, no memory is reserved for it. Hence, we can’t assign values to data members along with the structure definition.

3)
   (Blank – no output)

4) Error! This is because, in C we can’t keep a static variable inside a structure. The compiler allocates memory for given structure contiguously, and hence the same memory segment. *The static storage class has been discussed in detail in Chapter 20.*

5) 18 bytes