Chap 5

FUNCTIONS

1.O/p?

```
main()
{
  int a,b;
  a=sumdig(123);
  b=sumdig(123);
  printf("%d %d",a,b);
}

sumdig(int n)
{
  static int s=0;
  int d;
  if(n!=0)
  {
  d=n%10;
  n=(n-d)/10;
  s=s+d;
  sumdig();
  }
 else
    return(s);
}
```

ans: 6 12;

2.What error would the following function give on compilation.

```
F(int a,int b)
{
 int a;
 a=20;
return a;
}
```

a .missing parenthesis in return statement.
B .The function should be defined as int f(int a,int b)
C . Redeclaratin of a.
d. None of above.

Ans: C.

3.Thee is a mistake in the following code. Add a statement in it to remove it.

```
Main()
{
 int a;
 a=f(10,3.14);
 printf(“%d”,a);
}

f(int aa,float bb)
{
 return((float(aa)+bb);
}
```

ans: Add the following function prototype in main()

```
  float f(int aa,float bb);
```

4. Point error in the following code.

```
Main()
{
 int a=10;
 void f();
 a=f();
 printf(“%d”,a);
}

void f()
{
 printf(“HI”);
}
```

ans. In spite of decelerating that the function will return void the program is trying to collect the value in a variable.

5.Point error if any

```
main()
{
 int b;
 b=f(20);
 printf(“%d”,b);
 }
```

```
int f(int a)
{
  a>20?return(10):return(20);
}
```

ans: Return statement can not be used in format as shown in the conditional operator instead it should be as follows

```
   return(a>20?10:20);
```

6. A function can not be defined inside another function. <true/false>

ans: True.

7.Will the following function work?<yes/no>

```
f1(int a, int b)
{
   return(f2(20));
}

f2(int a)
{
 return(a*a);
}
```

ans:YES.

8.What are following two notations of defining functions commonly known as

```
 int f(int a, float b)
{
  /* some code */
}

int f(a,b)
int a, float b;
{
  /* some code */
}
```

ans: The first one is known as ANSI notation. And the second one is known as Kernighan and Ritche, or simply K & R notation

9.In function two return statements should not occuar. <True/False>

ans: FALSE.

10. In a function two return statements should not occur successively. <True/False>

ans: TRUE.

11.In C all functions except main() can be called recursively.

Ans:FALSE. Any function including main() can be called recursively.

12. Usually recursion works slower than loops. <True/False>

ans: TRUE.

13.Is it true that too many recursive calls may result in stack overflow?

Ans TRUE.

14. How many times the following program prints ‘Jambaree’?

```
main()
{
 printf(“\n Jambaree”);
 main();
}
```

a. infinite.
b. 32767 times
c. 65535 times
d. Till stack doesn’t overflow.

Ans: D