1. What will be the output of the following C code?

```c
#include <stdio.h>
void main()
{
        int x = 5.3 % 2;
printf("Value of x is %d", x);
}
```
(a) Value of x is 2.3
(b) Value of x is 1
(c) Value of x is 0.3
(d) Compile time error

Ans: (d)

2. What will be the output of the following C code?

```c
#include <stdio.h>
void main()
{
    int y = 3;
    int x = 5 % 2 * 3 / 2;
    printf("Value of x is %d", x);
}
```
(a) Value of x is 1
(b) Value of x is 2
(c) Value of x is 3
(d) Compile time error

Ans: (a)

3. Which is the correct C Equivalent Expression of the following Mathematical Expression

$$T = \frac{2m_1 m_2}{m_1 + m_2} \times g$$

(i) T=2*m1*m2*g/(m1+m2)

(ii) T= (2*m1*m2/(m1+m2))*g

(iii) T= 2*m1*m2*(m1+m2)/g

(iv) T= 2*m1*m2/(m1+m2)*g

(a) Only i

(b) Both ii & iv

(c) Both i and ii

(d) i,ii & iii

Ans: (b)

4. Choose an incorrect Statement.

While Evaluating an Arithmetic expressions in C Language

(a) Operands associated with highest priority are operated first.

(b) If the given expression involves parentheses, then the expression inside the parentheses must be evaluated first.

(c) If a unary minus is present in the expression, then the term associated with unary minus must be evaluated before any other expressions.

(d) The parenthesized and unparenthesized expressions dosn't follow the operator precedence.

Ans: (d)

5. What is the precedence of arithmetic operators (from highest to lowest)?
(a) %, *, /, +, −
(b) %, +, /, *, −
(c) +, -, %, *, /
(d) %, +, -, *, /

Ans: (a)

6. Which of the following is not an arithmetic operation?
(a) a * = 10;
(b) a / = 10;
(c) a ! = 10;
(d) a % = 10;

Ans: (c)

7. Which of the following data type will throw an error on modulus operation(%)?
(a) char
(b) short
(c) int
(d) float

Ans: (d)

8. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
int a = 10, b = 5, c = 5;
int d;
d = a == (b + c);
printf("%d", d);
}
```

(a) Syntax error
(b) 1
(c) 10
(d) 5

Ans: (b)

9. Which loop is guaranteed to execute at least one time.

A. for

B. while

C. do while

D. None of the above

Ans: c

10. What is the output of this program?

```
#include <stdio.h>

void main()

{
```

```c
    int a=10;

    if(a=5)

        printf("YES");

    else

        printf("NO");
}
```

A. YES

B. NO

C. Error

D. None of the above

Ans: A

11. How many times the loop will execute ?

```c
for(int i = 0 ; i < 10 ; i++)

{

    i  =  i*2;

    i--;

}
```

A. 10

B. 5

C. 0

D. Infinite

Ans: d

12. What is the output of this program?

```c
#include <stdio.h>

void main()
```

```
{
  int j = -5;
  for(;j;printf("%d ", j++));
}
```

A. -5 to -1

B. -5 to 0

C. -5 to infinity

D. Compilation Error

Ans: a

13. If the two strings are identical, then strcmp() function returns

A. -1

B. 1

C. 0

D. None

Ans: C

14. What is the output of this program?


```
void main()
{
    int a[8] = {1,2,3,4,5};
    printf("%d", a[5]);
}
```

A. 5

B. 6

C. 0

D. Garbage Value

Ans: C

15. The keyword used to transfer control from a function back to the calling function is

A. switch

B. goto

C. return

D. exit

Ans: C

16. What is the default return type if it is not specified in function definition?

A. void

B. int

C. float

D. short int

Ans: B

17. Find the output of below code -

```
#include <stdio.h>

struct cppbuzz{

   };

int main()
{
  printf("%d",sizeof(struct cppbuzz));
  return 0;
}
```
 A.0
 B.1
 C.2

D.4

Ans. A

18. Prior to using pointer variable
        (a) It should be declared
        (b) It should be initialised
        (c) It should be both declared and initialised
        (d) None of the above

    Ans: c

    19. * is known as
    a) Reference operator
    b) Address of operator
    c) Dereferencing operator
    d) None of the above

    Ans: C

20. int * ptr, p;
    a) ptr is a pointer to integer but p is not
    b) ptr and p both are pointers
    c) ptr and p both are not pointer
    d) None of the above

    Ans: A