1. What is the output of the program given below

```
#include<stdio.h>
main()
{
        char i=0;
        for(;i>=0;i++) ;
        printf("%d\n",i);
}
```

2. Which one MUST be correct in the following statements ?

   a) All Identifiers are keywords
   b) All Keywords are Identifiers
   c) Keywords are not Identifiers
   d) Some keywords are Identifiers

3. How much memory is allocated by the following definition ?

```
int (*x)[10];
```

4. What is the memory allocated by the following definition ?

```
int (*x)();
```

5. What is the Output of the Program ?

```
main()
{
        int i = 1 ;
        printf(i ?"one" : "zero") ;
}
```

   a) one
   b) zero
   c) error
   d) both and b

6. What is the Output of the Program ?

```
main()
{
     int i = 1 ;
     printf("%d",i ? 1 : 0) ;
}
```

a) 1

b) 0

c) error

d) none of the above

7. What is the output for the program given below

```
typedef enum grade{GOOD,BAD,WORST,BAD};

main()
{
       BAD g1;
       g1=1;
       printf("%d",g1);
}
```

8. Give the output for the following program.

```
#define STYLE1 char
main()
{
        typedef char STYLE2; STYLE1 x;
        STYLE2 y;
        clrscr();
        x=255;
        y=255;
        printf("%d %d\n",x,y);
}
```

9. Which of 'Arrays' or 'pointers' are faster ?

a) Arrays
b) pointers
c) Both take same time
d) Can't say

10. In the following program

```
#include<stdio.h>
main()
{
      char *pDestn,*pSource="I Love India";
      pDestn=malloc(strlen(pSource));
      strcpy(pDestn,pSource);
      printf("%s",pDestn); free(pDestn);
}
```

a) Free() fails
b) Strcpy() fails
c) prints I love India <Garbage>
d) error

11. What is the output for the following program

```
    #include<stdio.h>
    main()
    {
          char a[5][5],flag; a[0][0]='A';
          flag=((a==*a)&&(*a==a[0]));
          printf("%d\n",flag);
     }
```

12. main()

```
    {
          int i;
          i=(2,3);
          printf("%d",i);
     }
```

a) 2

b) 3

c) Compiler error

d) Syntax error.

13. main()

```
     {
          char str[]="GESL";
          printf("%d %d",sizeof(str),strlen(str));
     }
```

a) 5,5

b) 4,4

c) 5,4

d) 4,5

14.
```
main()
{
     int i ;
     for(i=0;i++;i<100)
      printf("hello world\n");
}
```

a) 100 times
b) 0 times
c) Infinite loop
d) None of the above.

15.
```
main()     {
     int i;
     for(i=1;i++;i<100)
       printf("hello world\n");
}
```

a) 100
b) 0 times
c) Infinite loop
d) None of the above.

16.
```
main()
{
     char c;
     scanf("%s",c);
}
```

a) Compiler dependent
b) Runtime error
c) Compiler error
d) scans the i/p.

17. main()

```
{
    int k=5;
    if(++k<5 && k++/5 || ++k<8);
    printf("%d\n",k);
}
```

a) 5
b) 6
c) 7
d) 8

18. main()

```
{
    int *ptr1,*ptr2;
    ptr1=(int *)malloc(sizeof(int));
    ptr2=func(20,10,ptr1);
    printf("%d %d\n",*ptr1,*ptr2);
}
int *func(int a, int b, int *c)
{
    int x=a+b;  *c=a-b;
    return(&x);
}
```

a) Bug in the code.
b) No Bugs prints correctly
c) Error
d) None of the above

19.
```
int main()
{
    int i = 10, j ;
    if ( ( j = ~i ) < i )
      printf ( "True" ) ;
    else
      printf ( "False" ) ;
}
```

a) True

b) False

c) Compiler Dependent

d) None of the above.

20. How many bytes are required to create a 3*3 matrix using pointer pointer to object of type T?

21. Assume the sizes of int, float and char as 4, 8 and 1 respectively.

```
main()
{
    FILE *fp;
    printf("%d\n",sizeof(fp) );
}
```

a) 2

b) 4

c) Compiler dependent

d) Error

```
22. main()
    {
       int a = 10,b = 20; a^=b^=a^=b;
       printf("%d\n %d\n",a, b);
    }
```

a) a=20,b=10

b) a=10,b=20

c) Syntax error

d) Unpredictable

```
23. main()

    {
       int i=10;
       switch(i)
       {
       case 10: printf("Hello ");
               {
       case 1 : printf("World ");
               }
       case 5: printf("Hello World ");
       }
    }
```

a) Hello

b) Hello

c) Hello World Hello World

d) Syntax Error

```
24. main()
   {
       char str1[]="Hello";
       char str2[]="Hello";
       if ( str1==str2 )  printf("True\n");
       else
         printf("False\n");
     }
```

a) True
b) False
c) Error
d) Unpredictable.

```
25. main()
    {
       # include <stdio.h>
       int i = 10 ;
       printf("%d\n", i/2 );
    }
```

a) 10
b) 5
c) error
d) warning.

```
26. #include <stdio.h>   /* Size of int is 4 bytes */
    # pragma pack(2)
    struct SIZE
    {
       int i; char ch ;  double db ;
    }
    main ()
    {

        printf ( "%d\n",sizeof(struct SIZE) );
    }
```

a) 12

b) 14

c) 16

d) 8

```
27. main()
    {
       int arr[]={ 1,2,3,4 };
       int *ptr ;;;;
       ptr++ = arr;
       printf("%d,%d",ptr[2],arr[2]);
       return 0;
     }
```

what is the output :

a) compile time error :multiple termination statements for pointer

b) lvalue required for ptr

c) prints 3 3

d) printd 4 3

```
28. main()
    {
       char s[10];
       scanf ("%s",s); printf(s);
    }
```

what is the output if input is abcd :

a) prints abcd
b) compiler error
c) prints abcd and 6 junk characters
d) printd s

```
29. main()
    {
       char c = 255;
       printf ("%d",c); return 0;
    }
```

what is the output

a) illegal character assignment
b) prints -1
c) prints 2
d) prints 255

```
30. main()
    {
        int i;
        for (i=7;i<=0;i)
            printf ("hello\n");
    }
```

what is the output ?

a) prints hello 7 times

b) prints hello 8 times

c) prints hello once

d) prints nothing

```
31. main()
    {
       printf( printf ("world") );
    }
```

a) prints world

b) prints printf ("world")

c) prints nothing

d) Runtime error

32. What is the output of the following code ?

```
int  main( )
{
    for( ; ;);
    printf("Hello\n");
}
```

a) give compilation error

b) prints Hello infinite times

c) Runs in an infinite loop without printing anything.

d) prints Hello once.

33. Output of the code?

```
FUNC (int *p)

{

    p = (int *)malloc(100);

    printf("p:%x",p);

}

int main( )

{

    int *ptr; FUNC(ptr);

    printf("Ptr:%x",ptr);

    return 0;

}
```

a) Both printf statements prints same values.

b) Both print different values.

c) Gives compile time error.

d) Gives run time error.

34. Output of the code?

```
int main()

{

    char a[] = "world";

    printf("%d  %d\n",strlen(a),sizeof(a));

    return 0;

}
```

a) 5,5

b) 6,5

c) 5,6

d) 6,6

35. What is the output of the following program ?

```
main()
{
      int a;
      #define y 10
      a=y;
      printf("%d",a);
}
```

a) 10

b) Compiler error

c) Run-time error

d) Garbage value

36. Interpret the given declaration

```
char ( * ( f ( ) ) [ ] ) (  )
```

a) f is a pointer to function returning char

b) f is a function returning an array of pointers to function returning char

c) Invalid declaration

d) f is a function returning pointer to array[] of pointer to function returning char.

37. what is the o/p ?

```
void main()
{
    char *mess[]={"Have","a","nice","day","Bye");
    printf("%d \t %d",sizeof(mess),sizeof(mess[1]));
 }
```

a) 16 4

b) 5   4

c) 20 4

38. what is the o/p of the following programe?

```
void main()
{
    int i,count=0; char *p1="abcdefghij";
    char *p2="alcmenfoip";
    for(i=0;i<=strlen(p1);i++)
{
     if(*p1++ == *p2++) count+=5;
    else  count-=3;
}
    printf("count=%d\n",count);
}
```

a) 15
b) 6
c) 12
d) compiler error

39. what does main return on successful execution?

a) 1
b) 0
c) -1
d)Nonzero

```
40.  main(int argc,char *argv[])
    {
        printf((argc > 1 ? "%c" : "%c",*++argv));
    }
```

If the i/p string is "CRANES Bangalore".

a) C
b) R
c) B
d) CRANES

41. How do u declare a pointer to an array of pointers to int?

a) int *ptr[5];
b) int **ptr[5];
c) int *(*ptr)[5];
d) cannot declare

```
42. main()
    {
        int a; char *p;
        a = sizeof(int) * p;
        printf("%d\n",a);
    }
```

a) compile error
b) run time error
c) 4
d) compiler dependent

```
43. #define SIZE sizeof(int)
    main()
   {
          int i=-1;
          if( i < SIZE ) printf("True\n");
          else
               printf("False\n");
    }
```

a) True

b) False

c) can't predict

d) None of these

```
44. int (*fun())[]
```

a) function returning a pointer to an array of int

b) function returning an array of pointers

c) pointer to a funtion which takes array as asrument

d) Compiler error

```
45. main()
   {
        int a=8,d;
        int *p; p=&a; d=a/(*p);
        print("%d\n",d);
    }
```

a) 1

b) 0

c) compiler error

d) run time error

```
46. main()
    {
        char *a="Hello";
        *a++ = 'h';
        printf("%s\n",a);
     }
```

a) hello

b) ello

c) runtime error

d) compiler error

```
47. main()
   {
        char p[]="Hello";
        p[0]='h'; printf("%s\n", p);
    }
```

a) hello

b) Hello

c) compiler error

d) run time error

```
48. #define mysizeof(a) (&(a)+1) - &(a))
    main()
    {
        float d;
        printf("%d\n", mysizeof(d) );
    }
    note:  assume sizeof float is 8 bytes
```

a) 8
b) 4
c) 1
d) compiler error

```
49. main()
     {
        int *p=10;
        printf("%d\n",*p);
    }
      a) 10
      b) run time error
      c) compiler error
      d) 5
```

```
50. main()
    {
        int i=-1;
        i<<=2;
        printf("%d\n",i);
    }
      a) -1
      b) -2
      c) -4
      d) 0
```

```
51. main()
   {
        int i= 0xffffffff;
        printf("%d\n",i);
     }
    note: size of int is 4 bytes
      a) -1
      b) 65635
      c) 100
      d) error
```

```
52. #include<stdio.h>
    main()
    {
         scanf("%d");
         printf();
    }
```

which of the following is correct?

a) compilation error
b) Run time error
c) No output
d) depends on the compiler

```
53. #include<stdio.h>
    #define islower(c)  ('a'<=(c) && (c)<='z')
    #define toupper(c)  (islower(c)?(c)-('a'-'A'):(c))
    main()
    {
        char *p="i am fine";
        while(*p)  printf("%c",toupper(*p++));
    }
```

a) bcd
b) AFE
c) aFe
d) BCd

54.
```
#include<stdio.h>
main()
{
    200; printf("tricky problem");
}
```

a)warning message

b)compilation error

c)run time error

d)tricky problem

55. which is the null statement?

a) ;

b) {}

c) '\0';

d) all of these

56. what is the correct prototype of printf function ?

a) printf(char *p,...);

b) printf(const *char *p,...);

c) printf(const char *p,...);

d) printf(const *char p,...);

57.
```
main()
{
    int *p ;
    p=(int *)malloc(-10);
}
```

a) returns NULL

b) returns a valid address always

c) compilation error

d) Runtime error

```
58. main()
    {
        for( printf("a") ;
        printf("b") ;
        printf("c") ) ;
    }
```

a) abc

b) abc abc abc .....(infinite times)

c) a bc bc bc ....(infinite times)

d) Error

```
59. main()
    {
        int i= 10 * fun() ;
        printf("%d",i);
    }
    fun()
    {
        return 10 ;
    }
```

a) 0

b) 10

c) 100

d) Error

```
60. int i= 10 * fun();
    main()
    {
        printf("%d",i);
    }
    fun() {
        return 10 ;
    }
```

a) 0

b) 10

c) 100

d) Error

61. Assume size of int to be 2 bytes :

```
    main()
    {
        int i = 100 ;
        printf("%d ", sizeof(i++));
        printf("%d ",i) ;
    }
```

a) 2 100

b) 2 101

c) 100   101

d) 101   100

```
62. main()
   {
       int A=1,B=2;
       if(A==B < printf("Hello "))
         printf("world\n");
       else
         printf("Bangalore\n");
    }
```

What is the o/p?

a) world

b) Hello bangalore

c) bangalore

d) Hello world.

```
63. main()
   {
       int i;
       for(i=0; i< 10; i++)
       {
         int j=10; j++;
         printf("j= %d\n", j);
       }
    }
```

what is o/p ?

a) 10 to 19

b) error j undeclared

c) 10 times 11

d) 10 - 18

```
64. union test
    {
        int a; union test *p;
    };
    main(){
        union test q;
        printf(" a= %d\n ", q.a);
    }
```

what is o/p?

a) 0
b) syntax error
c) garbage value
d) run time error

```
65. register int a,b;
    main()
    {
        for(a=0 ; a<5 ; a++) b++;
    }
```

a) 5
b) 4
c) 0
d) error

```
66. # define dprint(expr)  printf(" expr = %d \n ", expr)
     main()
    {
       int i=10,j=2;
       dprint(i / j) ;
    }
```

a) 5

b) expr = 5

c) i / j= 5

d) error.

67. What is the Output of the Program ?

```
    #define p a

    main()
    {
       int a = 1;
       printf("%d %d ",a++,p++) ;
    }
```

a) 1, 0

b) 2, 0

c) 1 2

d) 2,1

68. #include<stdio.h>

```
    main()
    {
        #include<stdio.h>
        int a = 90 ;
        printf("%d",a) ;
    }
```

a) 90
b) compilation error
c) linker error
d) runtime error

69. What is the Output of the Program ?

```
    main()
    {
        main() ;
    }
```

a) compilation error
b) Infinite loop
c) executes until the stack overflows and leads to runtime error
d) none of the above

70. What is the Output of the Program ?

```
    #define max "hello"
     main()
     {
           printf(max) ;
     }
```

a) compilation error
b) Preprocessing error
c) runtime error
d) hello

71. What is the Output of the Program ?

```
#define max main()
main()
{
    max ;
    printf("hello wolrd\n ") ;
}
```

a) compilation error

b) Preprocessing error

c) Infinite loop

d) executes until the stack overflows and leads to Runtime Error

72. What is the Output of the Program ?

```
typedef int *p ;
main()
{
    int a = 90 ; p p1 ;
    p1 = &a ; printf("%d",*p1) ;
}
```

a) 90

b) compilation error

c) runtime error

d) none of the above

73. Select the choice which is wrong ?

a)'volatile' is a keyword only

b)'volatile' is a qualifier

c)'volatile' is a modifier

d)'volatile' is a Identifier

74. Consider the following Program

```
main()
{
    int i,j;
    i = 06;  j = 09;
    printf ("%d %d\n",i,j);
}
```

a) 6 9

b) 6 11

c) 06 09

d) Compilation Error

75. What is the Output of the Program ?

```
main()
{
    int a = 90 , b = 100 ;
    a++ ;   a = (a ^ b) ^ (a = b );
    b = a^b^a ;a ;printf("%d %d",a++,b++) ;
}
```

a) 90 100

b) 100 90

c) 101 91

d) 91 100

76 What is the Output of the Program ?

```
main()
{
    int a = 10 , b = 100 ;
    swap(&a , &b) ;
    printf("%d %d",a,b) ;
}
 swap(int *a , int *b)
{
    *a = *a + *b ;
    *b = *a - *b ;*a = *a - *b ;
    swap1(&a , &b) ;
}
 swap1(int **a , int **b)
{
    **a = **a + **b ;
    **b = **a - **b ;
    **a = **a - **b ;
}
```

a) 100 10
b) 10 100
c) lvalue is required in fun main
d) error !!

77. What is the Output of the Program ?

```
main()
{
    void *ptr ;
    int a = 10 ;
    ptr = &a ;
    printf("%d",*ptr) ;
}
```

a) error

b) 10

c) 20

d) none

78. What is the Output of the Program ?

```
main()
{
    void *ptr ;int a = 90 ;
    char *ptr1 = "hello" ;
    ptr = &a ;ptr = ptr1 ;
}
```

a) executes w/o any error

b) compilation error

c) runtime error

d) none

79. What is the Output of the Program ?

```
main()
{
    char *pSrc = "helloo" ;
    char *pDst = "strcat" ;
    while((*(pSrc++)  = *(pDst++)) != '\0')    ;
}
```

a) Runtime error
b) address is copied
c) contents are copied
d) none

80. What is the Output of the Program ?

```
int global = 10 ;
main()
{
    int global = 10 ;
    printf("%d",global) ;
}
int global ;
```

a) 10
b) 11
c) error
d) none

81. What is the Output of the Program ?

```
main()

{

    int a = 1 ;

    int b = 0 ;

    a = a++ + b * a++ ;

    printf("%d",a) ;

}
```

a) error

b) 3

c) 1

d) 2

82. What is the Output of the Program ? (size of int is 4 bytes)

```
struct s

{

    int si;

    union u

    {

        float uf;

        char  uc;

    };

};

main()

{

    printf("%d",sizeof(struct s));

}
```

a) 8
b) 3
c) 6
d) 7

83. What is the Output of the Program ?

```
struct st
{
    int a;
    char b;
}
main()
{
}
```

a) Compilation error
b) struct st is considered as return type of main
c) main is a variable of struct st
d) Run time error

84. What is the Output of the Program ?

```
typedef struct info
{
    int i;
    char b;
}
    node;
main()
{
    struct info node1;node1.i=55;
    printf("%d",node1.i);
}
```

a) 55
b) Not possible to use struct info
c) Compilation error
d) Garbage value.

85. What is the Output of the Program ?

```
struct item
{
    int i;
    int display()
{
    printf("hello world\n");
}
};
main()
{
    strcut item vara; vara.display();
}
```

a) hello

b) hello world

c) Compile time error

d) garbage

86. What is the Output of the Program ?

```
struct item
{
    int (*ptr)();
};
int display()
{
    printf("Dennis Ritchie\n");
}
main()
{
    struct item object;
    object.ptr=display;
    object.ptr();
}
```

a) Dennis Ritchie
b) Address of display function
c) address of object
d) Error

87. What is the output of the program ?

```
main()

{

    #define x 5  int b;

    b = x;

    printf("%d",b);

}
```

  a) Compiler Error

  b)Runtime error

  c) Garbage value

  d) 5

88. What is the Output of the Program ?

```
union tag
{
    int i;  char x; char y;
}name;
(Assume Storage is Little Endian technique)
int main()
{
    name.i=258;
    printf("\n x = %d    y = %d  ",name.x,name.y);
}
```

```
 a)   x = 1    y = 1
 b)   x = 2    y = 2
 c)   x = 1    y = 2
 d)   x = 2    y = 1
```

89. Consider the Program,

```
int main()
{
    int a[20];
    int *p,*q,val;
    p = &a[0];
    q = &a[10];
    val = q - p;
    printf("q %u  p %u  val %d  ", q, p, val);
}
```

Assume p = 1000, sizeof(int) = 2, what is the value of q and val?

a) q = 1020 p = 1000 val = 20

b) q = 1020 p = 1000 val = 10

c) q = 1010 p = 1000 val = 10

d) q = 1010 p = 1000 val = 20

90. Consider the Program, Alignment is 4 bytes, sizeof(int) is 4 bytes

```
struct key
{
    char *word[2];
    int count;
    char c;
}
    abc;
 int main()
{
    printf("\nsize %d",sizeof(abc));
}
```

What is the size of abc?

a) 8

b) 7

c) 16

d) 5

91. What is the output of the following program ?

```
main()
{
    int a;
    fun();
    printf("%d",a);
    a=50;
}
    fun()
{
    int i;
    *(&i+4) =  100;
}
```

a) 50

b) Garbage value

c) 100

d) Compiler error

92.What is the Output of the Program ?

```
typedef int *ABC;

typedef ABC XYZ[10];

int main()

{

    XYZ var;

}
```

A. var is an array of 10 integer pointers.

B. var is a pointer to an integer array. Options:

a) only A is correct.

b) only B is correct.

c) both A and B are correct.

d) typedef statements are in wrong order.

93.What is the output generated?

```
main()
{
    char *s = "Hello";
    printf("%s", s + 1);
}
```

a) Hello
b) ello
c) e
d) none of these.

94. What will be printed on the screen ?

```
#define s -50
main()
{
    int s;
    #ifdef s
    printf("Hell\n");
    #else
    printf("Heaven\n");
    #endif
}
```

a)Hell
b) Heaven
c) Compilation error
d) HellHeaven

95. Give the output for the following program segment (macro.c)

If it was compiled as , gcc macro.c –o macro DTRUE

```
#ifdef TRUE

int I = 0;

#endif

main() { /* K&R style */

    int j = 0;

    printf("%d %d\n",i,j);

}
```

96.How many times can a comment be nested ?

a)comment_nest_limit times
b)comment_limit times
c)one time
d)Not even Once

97. What is the output of the following program #include<stdio.h>

```
main() {    /* K&R style */

    int i=0;

    fork();

    printf("%d",i++);

    fork();

    printf("%d",i++);

    fork();

    wait();

 }
```

98. In the following program segment. Assume **odd(b)** is macro.

```
#include<stdio.h>
main()
{
    int a=2; int b=9; int c=1;
    while(b)
    {
      if(odd(b))  c=c*a;  a=a*a; b=b/2;
    }
    printf("%d\n",c);
}
```

How many times is c=c*a calculated?

99. In the program segment in question 98 what is the value of **a** at the end of the while loop?

100. What happens when we compile this program ?

```
# undef  __FILE_
# define __FILE__  "CRANES"
main()
{
    printf("%s\n",__FILE__);
}
```

a)Compilation Error

b)Run-Time Error

c)Compiles But gives a Warning

d)Compiles Normal

101. If a global object is not defined(just declared), what happens?

102.What happens when we compile this program ?

```
# define LINE

# define NAME "CSIL"

main()

{

     printf("%d %s\n",LINE,NAME);

}
```

a) Compilation Error

b) Compiles but Warns

c) Runtime Error

d) Compiles Normally

103.
```
int main()

{

     int i = 5;

     if(1)

     {

          static int i;

          i++;printf("%d", i);

     }

     printf("%d", i);

}
```

a) error

b) 5,0

c) 5,1

d) 1,5

```
104.int main()
     {
          int a[4] = {23, 67, 90};
          printf(" %d", a[3]);
     }
```

a) junk

b) error

c) 0

d) 1

```
105.int main()
     {
          int i = 1, 2;
          printf("%d", i);
     }
```

a) 1

b) 2

c) error

d) none

```
106)int main()
     {
          int i;
          for( i=0; ; i++)
          {
               i = i+2;
               break;
               printf("%d", i);
          }
     }
```

a) 0

b) 2

c) error

d) none

```
107)int main()
     {
          int i;
          i = 1, 2;
          printf("%d", i);
     }
     a) 1

     b) 2

     c) error

     d) none
```

```
108)#include<stdio.h>

     int i =20;
     int maxlen = i;
     int main()
     {
          int j = i;
          printf("i=%d , j=%d\n", i , j);
     }
     a) i=20 , j=20

     b) i=20 , j=junk

     c) error

     d) none
```

```
109)int main()
     {

          int i =10;

          printf("%d", j);

          printf("%d",i);

     }
     int j = 20;

     a) j=20 , i=10

     b) j=junk , i=10

     c) compile time error

     d) runtime error
```

```
110)int i =20;
     int i,j=10;
     main()
     {
          int j =20;
          printf("i=%d , j=%d\n", i, j);
     }
     a) redeclaration error
     b) i=20 , j=10
     c) i=20 , j=20
     d) none
```

```
111) int main()
     {
          int k = 2, i = 10;
          while(k)
          {
               printf("%d\n",disp(i));
          }
     }
     int disp(int k)
     {
          static int i=0;
          return i=i+k;
     }
     a) 10, 10
     b) 10, 20
     c) 20, 10
     d) Infinite loop
```

112) Header files usually contains

a)only definitions

b)only declarations

c)both

d)compiled code for functions

113)
```
int main()
{
     int i =3;
     while(i)
     {
          int i =10;     printf("%d",i);
     }
}
```

a) 10, 9, 8, 7, ...1

b) 10, 10, 10, 10, ... Infinite times

c) 10, 10, 10

d) none

114)
```
char s[] = "hello\0 world";
printf("%s...%d",s,strlen(s));
```

What is the output?

a) hello...5

b) hello\0 world...12

c) hello...12

d) compile time error

115)
```
printf("%%%s","hello");
```

What is the output?

a) %%%s

b) %%%

c) %hello

d) hello

116 What does fgetc return

a) char

b) int

c) unsigned int

d) void

117) int i = 24;

printf("%xd",i);

What is the output?

a) 18

b) 24

c) 18d

d)compile time error

118) What is return type of freopen

a) int*

b) FILE *

c) int

d) void

119) struct node

```
{
     int i;
} ;
main()
{
     struct node n1;
     printf("%d",n1.i);
}
```

o/p of the program:

a) 0

b) Garbage value

c) error.

4) warning

```
120) struct node_tag      /*    (On 32-bit machine) */
     {
          int i;struct node_tag *ptr;
     } ;
     main()
     {
          printf("%d",size(node_tag));
     }
```

o/p of the program:

a) 8

b) 6

c) Garbage value

d) error

```
121)typedef struct node_tag
     {
          int i=0;int j;
     }
          node;main()
     {
          node n1;printf("%d",n1.i);
     }
```

o/p of the program:

a) 0

b) warning

c) Garbage value

d) error

```
122) struct
     {
          int i;
     }
          node ;
     main()
     {
          printf("%d",node.i);
     }
```

o/p of the program:

a) 0

b) Garbage value

c) error.

d) warning

```
123) struct node_tag

     {

          int a;struct node_tag *pt;

     } ;

     main()

     {

          struct node_tag n1;n1.pt=&n1; n1.pt->a=5;

          printf("%d",n1.a);

     }
```

o/p of the program:

a) error

b) warning

c) 5

d) Garbage value

124) int n;scanf("%d",n);

what is the output?

a) read 1 integer value

b) compile time error

c) runtime error

d) reads 0

125)strchr(s,c)

what this will do?

a) return pointer to first 'c' in 's' or NULL if not present

b) return pointerto last 'c' in 's' or NULL if not present

c) concatenate c to s in the beginning

d) concatenate c to s at the end

126) When calloc() is called memory is initialised to

a) Garbage

b) NULL

c) 0

d) -1

127) (void *) is called

a)pointer to void

b)pointer to any data type

c)generic pointer

d)None of the above

128)What the putchar() will return on error

a) 0

b) EOF

c) NULL

d) doesn’t return

129)what is the output of the following ?

```
i=5;

i=i++ * i++;

printf("%d",i);
```

a) 30

b) 49

c) 25

d) 27

130) what is the output of the following ?

```
i=5;

printf("%d",i++ * i++);
```

a)30

b)49

c)25

d)37

131)

```
#include<stdlib.h>

int main(void)

{

     putchar("0123456789ABCDEFGHIJKL" [16 & 17 ] );

     return EXIT_SUCCESS;

}
```

a) Error

b) No Output

C) Garbage

d) G

```
132)#include<stdio.h>

     int main()

     {

          char *p = "What is the definition of ERROR?\n";

          *(p+25);fprintf(stderr,"%s",p);

          return 'c';

     }

     a) prints "ERROR" to stderr.
     b) Error
     C) Garbage
     d) prints "What is the definition of ERROR?" to screen & stderr
```

```
133)#include<stdio.h>

     int main(void)
     {
          puts("hello\0world");
     }

     a) Error
     b) hello$^@$S
     c) hello
     d) world$%^#^
```

```
134)#include<stdio.h>

     typedef char  (*PFI) () ;

     char main(int argc,char *argv[],char *environ[])

     {

          PFI a = main;

          printf("%s",a);

     }

     a)Compile Time Error
     b)Infinite Loop
     c)Prints some garbage
     d)Run Time error
```

135) union u

```
{
     int ival;float fval;char *sval;
}
```

size of u is

a) 8 bytes
b) 4 bytes
c) compile time error
d) 12

136) struct x

```
{
     int i; int j;int k;
} ;
     struct x *p;
     struct x arr[3];
     p =&arr[0];
     p++;
```

what is p pointing to

a) pointing to i of arr[0]
b) pointing to j of arr[0]
c) pointing to k of arr[1]
d) pointing to i of arr[1]

137) struct a

```
{
     int b;
};
struct b
{
     int b;
};
int main()
{
     struct a first;
     struct b second;
     first.b =10;
     second = first;
     printf("%d",second.b);
}
```

a) 10
b) garbage
c) compile error
d) run time error

138) struct a

```
{

int x;float y; double z;struct a b;

};
```

a) no error
b) compile error
c) run time error
d) none of the above

139) struct a

```
{
     struct b
     {
          int a;int b;
     }c;
     int *ptr;
}d;

d.ptr=&d.c.a;
```

a) compile error
b) syntax error
c) Both a and c
d) No error

140)#include<stdio.h>

```
int main(void)
{
     int *intPtr ;
     intPtr = (char*)malloc(sizeof(10));
     printf("\n The start address is %d \n ",intPtr);return 0;
}
```

a) Compilation Error
b) Runtime Error
c) Will give a Warning , but run any way
d) neither warning nor error

```
141)#include<stdio.h>

     int main(void)

     {

          FILE *fp1,*fp2;

          int c;    fp1 = fopen("testing","a");

          fp2 = fopen("testing","w");

          while( (c = getchar()) != '\n'){fputc(c,fp1);}return 0;

     }
```

a) Compilation Error
b) Runtime Error
c) contents of file testing are appended
d) contents of file testing are overwritten

```
142.#include<stdio.h>

     int main(void)

     {

          int intNum1,intNum2,num = 1,i;

          printf("\nEnter first number \n");

          scanf("%d",&intNum1);

          printf("\nEnter second number \n");

          scanf("%d",intNum2);for(i = 0;i<=3;i++){

          num = intNum1 * intNum2 * num;

     }

          printf("\n num = %d " , num);

          return 0;

     }
```

a) Compilation Error
b) Runtime Error
c) Successful execution
d) Junk Value

```
143) #include<stdio.h>

     int main(void)

     {

          char str[5];

          char *newstr;

          printf("\nEnter first string \n"); scanf("%s",&str);

          printf("\n The string you have entered is %s ",str);

          newstr = gets(str);

          printf("\n num = %s " , newstr);

          printf("\n the new string is %s ",newstr);

          return 0;

     }
```

a) Compilation Error
b) Runtime Error
c) Dangerous code. **gets** has got a bug.
d) Junk Value

```
144)#include<stdio.h>

     int main(void)

     {

          FILE *fp;char *str ;

          char *newstr;

          fp = fopen("source","r");

          newstr = fgets(str,5,fp);

          printf("\n The new str is %s " ,newstr);

          return 0;

     }
```

a) Compilation Error
b) Runtime Error
c) Successful execution
d) Segmentation Fault

145)

```
int a=1,b=0, x;
x = a++ && ++b;
printf("%d %d %d ",a,b,x );
```

output ?

a) 1 1 2
b) 2 1 0
c) 2 0 2
d) 2 1 1

146)

```
char *fn();
main()
{
     char *s;s = fn();
     printf("%s\n", s);
}
char *fn()
{
     return "Hello";
}
```

output is ?

a) null
b) garbage
c) Hello
d) Compilation Error

147)

```
int i;
for( i=0; i<10-1; i+=2 );
i+= 2;
printf("i = %d\n", i );
```

output is ?

a) 12
b) 11
c) 10
d) 13

148) what is the output of the following program ?

```
main()

{

     int i;

     i = f();

     printf("%d", i);

}

f()
{

     return 1,2,3;

}
```

a) 1
b) Compilation error
c) 2
d) 3

149) What is the difference between ++*ip and *ip++ ?

a) both increment value
b) ++*ip increment value and *ip++ increment address
c) both increment address
d) ++*ip increment address and *ip++ increment value

150) What is the output of the following program ?

```
# include <stdio.h>

int main ( void )

{

     int x, y, z;   x = 2, y = 4;z = x && y;

     printf("z = %d\n", z );

}
```

a) 1
b) 0
c) None of these
d) 8

151) What is the output of the following program ?

```
# include <stdio.h>

int main ( void )

{

     int x = 48;

     printf("x = %s\n", x );

}
```

a) 10
b) 0
c) Run Time Error
d) Compilation Error

152) What is the output of the following program ?

```
# include <stdio.h>

# define ONE 1

# define TWO 2

# define ONE TWO

# define TWO ONE

int main ( void )

{

     printf("ONE = %d, TWO = %d\n", ONE, TWO );

     return 0;

}
```

a) ONE = 1, TWO = 2
b) TWO = 1, ONE = 2
c) Compilation Error
d) None of these

153) If the command line arguments for the following program are

<a.out> <CranesSoftwareLtd>,

what is the output of the program ?

```
# include <stdio.h>

int main( int argc, char **argv )

{

     printf("output = %s\n", *argv[1] );

}
```

a) CranesSoftwareLtd
b) G
c) Compilation Error
d) Run Time Error

154)What is the output of the following ?

```
# include <stdio.h>

void fun( int, int );

int main ( void )

{

     fun( 12, ( 13, ( 14, 15 ) ) );

     return 0;

}

void fun( int x, int y )

{

     printf("x = %d,y = %d\n", x, y );

}
```

a) x = 12, y = 13
b) x = 14, y = 15
c) x = 12, y = 15
d) Syntax Error(Too Many Arguments to fun())

155)
```
#define min((a),(b))  ((a)<(b))?(a):(b)

     main()

     {

          int i=0,a[20],*ptr;

          ptr=a;

          while(min(ptr++,&a[9])<&a[8]) i=i+1;

          printf("i=%d\n",i);

     }
```

a) 5
b) Pre-processing Error
c) Compiler Error
d) None

156)
```
main()

     {

          char b[10]={1,2,3,100,101,102};    int x;

          for(x=0;x<4;x++)    {

               b[x]=x+'a';

          }    printf("%s",b);

     }
```

157)
```
char a =0xAA ;

     int b ;

     b = (int) a ;

     b = b >> 4 ;

     printf("%x",b);
```

What is the output of the above program segment ?

```
158)  struct s1 {
          struct {
               struct {
                    int x;
               }s2;
          }s3;
      }y;
```

How does one access x in the above given structure definition ?

159) What is the size of the array declared as double * X[5] ?

```
160) void f(int y)
     {
          struct s *ptr;
          ptr = malloc (sizeof (struct s)+99*sizeof(int));
     }
     struct s
     {
          int i;float p;
     };
```

when free (ptr) is executed, what will happen?

```
161)enum day = { jan = 1 ,feb=4, april, may};
```

what is the value of may?

a)4 b)5 c)6 d)11e)none

```
162) main()
     {
          int x,j,k;
          j=k=6;x=2;
          x=j*k;
          printf("%d", x);
     }                                 -
```

```
163) f(x)
     {
          if(x<=0) return;
          else f(x-1)+x;
     }
     what is the value of f(5)?
```

```
164) i=20,k=0;
     for(j=1;j<i;j=1+4*(i/j))
     {
     k+=j<10?4:3;
     }
     printf("%d", k);
```

```
165. int i =10;
      main()
     {
          int i =20,n;
          for(n=0;n<=i;){
               int i=10; n++;
          }
          printf("%d", i);
     }
```

```
166) int y=10;

     if( y++>9 && y++!=10 && y++>10)

             printf("%d\n %d", y, i);

     else

             printf("%d\n %d", y, i);
```

```
167) f=(x>y)?x:y

     a) f points to max of x and y
     b) f points to min of x and y
     c)error
     d) ........
```

```
168) int x;

     main()

     {

          int x=0;

          {

               int x=10;x++;

               change_value(x);

               x++;modify_value();

               printf("First output: d \n",x);

          }

          x++;

          change_value(x);

          printf("Second Output : %d\n",x);

          modify_value();

          printf("Third Output : %d\n",x);

     }

     modify_value()
     {
        return (x+=10);
     }
     change_value()
     {
        return(x+=1);

     }
```

```
169) main()
     {
          int x=20,y=35;
          x = y++ + x++;
          y = ++y + ++x;
          printf("%d %d\n",x,y);
     }
```

```
170) main()
     {
          char *p1="Name";
          char *p2;
          p2=(char *)malloc(20);
          while(*p2++=*p1++);
          printf("%s\n",p2);
     }
```

```
171)  main()
     {
          int x=5;
          printf("%d %d %d\n",x,x<<2,x>>2);
     }
```

```
172) #define swap1(a,b) a=a+b;b=a-b;a=a-b;

     main()

     {

             int x=5,y=10;

             swap1(x,y);

             printf("%d %d\n",x,y);

             swap2(x,y);

             printf("%d %d\n",x,y);

     }

     int swap2(int a,int b)

     {

             int temp;

             temp=a;

             b=a;

             a=temp;

             return;

     }
```

```
173) main()

     {

        char *ptr = "Ramco Systems";

        (*ptr)++;

        printf("%s\n",ptr);

        ptr++;

        printf("%s\n",ptr);

     }
```

```
174) #include<stdio.h>

     main()
     {
             char s1[]="Ramco";
             char s2[]="Systems";
             s1=s2;
             printf("%s",s1);
     }
```

```
175) #include<stdio.h>
     main()
     {
             char *p1;
             char *p2;
             p1=(char *) malloc(25);
             p2=(char *) malloc(25);
             strcpy(p1,"Ramco");
             strcpy(p2,"Systems");
             strcat(p1,p2);
             printf("%s",p1);
     }
```

```
176) For the following C program
     #define AND &&
     #define ARRANGE (a>25 AND a<50)
     main()
     {

          int a = 30;
          if (ARRANGE)
              printf("within range");
          else
              printf("out of range");

     }

     What is the output?
```

177) For the following C program

```
#define AREA(x)(3.14*x*x)

main()

{

     float r1=6.25,r2=2.5,a;

     a=AREA(r1);

     printf("\n Area of the circle is %f", a);

     a=AREA(r2);

     printf("\n Area of the circle is %f", a);

}
```

What is the output?

178) What do the following statements indicate. Explain.

```
Int (*p)[10]

Int *f()

Int (*pf)()

Int *p[10]
```

179. typedef struct

```
{

     char *ptr;

     nodeptr next;

} * nodeptr;
```

what does nodeptr stand for?

180) What do you mean by int   *x[]() ?

```
181) struct list
     {
          int x;
          struct list *next;
      }*head;
     struct head.x =100
     above is correct / wrong
```

```
182. o/p=?
       int i;
       i=1;
       i=i+2*i++;
       printf(%d,i);
```

```
183.  FILE *fp1,*fp2;
      fp1=fopen("one","w")
      fp2=fopen("one","w")
      fputc('A',fp1)
      fputc('B',fp2)
      fclose(fp1)
      fclose(fp2)}
      Is it correct/Error ?
```

```
184. #include<malloc.h>
      char *f()
      {
          char *s=malloc(8);
          return strcpy(s,"goodbye");
     }
     main()
     {
          char *f();
          printf("%c",*f()='A');
     }
```

o/p=?

```
185) #define MAX(x,y) (x)>(y)?(x):(y)
          main()
        {
               int i=10,j=5,k=0;
               k= MAX(i++,++j);
               printf("%d %d %d %d",i,j,k);
               int a=10,b=5, c=3,d=3;
               if((a<b)&&(c=d++))
                    printf("%d %d %d %d", a,b,c,d);
               else
                    printf("%d %d %d %d", a,b,c,d);
     }
```

```
186) main()
      {
           int i = 10;
           printf(" %d %d %d \n", ++i, i++, ++i);
      }
```

```
187) #include<stdio.h>
      main()
      {
           int *p, *c, i;
           i = 5;
           p = (int*) (malloc(sizeof(i)));
           printf("\n%d",*p);
           *p = 10;
           printf("\n%d  %d",i,*p);
           c = (int*) calloc(2);
           printf("\n%d\n",*c);
      }
```

```
188. #define MAX(x,y) (x) >(y)?(x):(y)
     main()
     {
          int i=10,j=5,k=0;
          k=  MAX(i++,++j);
          printf("%d..%d..%d",i,j,k);
     }
```

```
189. #include <stdio.h>

     main()

     {

             enum _tag{ left=10, right, front=100, back};

             printf("left is %d, right is %d, front is %d, back is

             %d",left,right,front,back);

     }


190. main()

     {

             int a=10,b=20;

             a>=5?(b=100):(b=200);

             printf("%d\n",b);

     }


191) #define PRINT(int) printf(#int‟ = %d\n” ,int)

      main()

      {

           int x,y,z;

           x=03;y=02;z=01;

           PRINT(x^x);

           z<<=3;PRINT(x);

           y>>=3;PRINT(y);

     }
```

192. Supposing that each integer occupies 4 bytes and each charactrer 1 byte , what is the output of the following programme?

```
#include<stdio.h>
main()
{
     int a[] ={ 1,2,3,4,5,6,7};
     char c[] = {' a','x','h','o','k'};
     printf("%d\t %d ", (&a[3]-&a[0]),(&c[3]-&c[0]));
}
```

193. what is the output of the program?

```
#include<stdio.h>
main()
{
     struct s1 {int i; };
     struct s2 {int i; };
     struct s1 st1;
     struct s2 st2;
     st1.i =5;
     st2 = st1;
     printf(" %d " , st2.i);
}
```

ANSWER KEYS

1. -128
2. c
3. 4bytes
4. 4bytes
5. a
6. a
7. error **(cant define twice BAD)**
8. -1-1
9. a
10. c
11. 1 **(with Warning)**
12. b
13. c
14. b
15. c
16. b
17. c
18. a **(Returning the address of a local variable!)**
19. a
20. 12 + 4 = 16 (**3 pointers to T** + **1 pointer to pointer to T**)
21. b
22. a
23. c
24. b
25. b
26. b
27. b
28. a
29. b
30. d

31. d **(First printf return 5, which is the argument to the Second!)**
32. c
33. b
34. c
35. a
36. b
37. c
38. b
39. b
40. d
41. c
42. a **(Invalid operands to binary *)**
43. b **(signed int is converted to unsigned & then compared…)**
44. a
45. a
46. c (Modifying the String contents in undefined!)
47. a
48. c **(Cast &a to char * in both places to get the correct size)**
49. b
50. c
51. a
52. a **(printf call violates it's prototype!)**
53. b
54. d
55. a
56. c
57. a
58. c
59. c
60. d
61. a
62. d

63. c **(As q is a local objet!)**

64. c

65. d

66. b

67. d

68. a

69. c

70. d

71. d

72. a

73. b

74. d

75. d

76. b

77. a

78. a

79. a

80. a

81. b

82. a

83. b

84. a

85. c

86. a

87. a **('b' undeclared…)**

88. b **(Bit number 8 and 1 are enabled)**

89. b

90. c

91. c

92. a

93. b

94. c

95. 0 0 **(-D is the switch of GCC/CC, which defines the macro TEST.)**

96. d

97. Runtime Error

98. Two times

99. 65536

100. c

101. Linker throws an error: Undefined reference to the **object**

102. Ans (a)

103. Ans (d)

104. Ans (c)

105. Ans (c)

106. Ans (d)

107. Ans.(a)

108. Ans (c)

109. Ans (c) (**There is no declaration for 'j' before its use)**

110. Ans (c) **The second 'i' is considered as declaration, So not an error**

111. Ans (d) **(local 'k' of main is always TRUE)**

112. Ans (b)

113. Ans (b)

114. Ans (a)

115. Ans : (c)

116. Ans : (b)

117. Ans : (c)

118. Ans : (b)

119. Ans : (b)

120. Ans : (d) **(Just tag name doesn't represent the data type)**

121. Ans : (d) **(Can't initialize inside structure!!)**

122. Ans : (a)

123. Ans : (c)

124. Ans : (c)

125. Ans : (a)

126. Ans : (c)

127. Ans : (c)

128. Ans : (b)

129. Ans : (d)

130. Ans : (c)

131. Ans : (d) **[String name gives base address, 16 & 17**
So, "0123456789ABCDEFGHIJKL"[17] is the character G]

132. Ans : (d)

133. Ans : (c)

134. Ans : (c)

135. Ans : (b)

136. Ans : (d)

137. Ans : (c) **(Two different objects can't be assigned)**

138. Ans : (b)

139. Ans : (d)

140. Ans : (c)

141. Ans : (d) **(Recent mode of opening is write-only)**

142. Ans : (b) **(In second scanf)**

143. Ans : (c) **(gets doesn't how much to read!)**

144. Ans : (b) **(No memory is reserved for the string)**

145. Ans : (d)

146. Ans : (c) **[The string "Hello" isn't local, so you can return it]**

147. Ans : (a)

148. Ans : (d) **[Comma operator has Left-Right associativity.**
In the statement Expr1, Expr2
Expr1 is evaluated first, then Expr2 is evaluated. The result of the expression is the Expr2.So, in the expression return (1, 2, 3), 3 is the resultant.]

149. Ans : (b)

150. Ans : (a)

151. Ans : (a)

152. Ans : (a)

153. Ans : (d)

[

From the declaration, char **argv;

We note that, argv is a char **

***argv (or) argv[] is a char ***

****argv (or) *argv[] (or) argv[][] is a char.**

So, *argv[1] is a character, not an address. The format specifier %s expects the address of a character array (char *) ended with null character.]

154. Ans : (c)

[

As described earlier regarding the Comma operator,

```
(14, 15)  results in 15,
(13, 15)  also results in 15,
(12, 15)  are passed to the function fun.
```

]

155. Ans : (a)

156. Ans : abcdef

157. Ans: fffffffa

158. Ans: y.s3.s2.x

159. Ans. 5 * sizeof (double *)

160. Ans: struct s is of Incomplete type.

161. Ans: (c)

162. Ans:36

163. Ans: unpredictable

164. 4

165. 10

166. 13 11

167. Ans : a

168. Ans: d   1   1

```
169. Ans :57,94
170. Ans :Prints Null character
171. Ans :5 20 1
172. Ans :10 5
          10 5
173. Ans :Run time Error (Segmentation fault)
174. Ans :Compiler Error (array name is not an lvalue!)
175. Ans :RamcoSystems
176. Ans :within range
177. Ans. Area of the circle is 122.656250
           Area of the circle is  19.625000
178. Ans: p is  a pointer to array 10 of an int
          f is a function returning pointer to an int
          pf is a pointer to function returning an int
          p is an array of pointers to an int
179. Ans: Compiler error inside the structure
180. expl: Elments of an array can't be functions.
181. Ans: WRONG
```

[

Before using the ptr type struct variable we have to give memory to that. And also when access the member of the structure using

"->" opearator when you have a pointer to a structure.

]

```
182. Ans: 4
183. Ans: no error. But It will over writes on same file.
184. Ans: A
185. Ans: 12 6 11 0 10 5 3 3
186. Ans: 13 11 11
```

```
187. Ans: 0
          5  10
          0
188. Ans: 12..6..11
189. Ans: left is 10, right is 11, front is 100, back is
         101
190. Ans: 100
191. Ans: x^x = 0  x = 3  y = 0
192. Ans : 3  3
193. Ans: error
```

[

Diff struct variables should not assigned using "=" operator.

]