```
#include<stdio.h>
int main0(void)
{
char ch = 65;
int x = 9;
printf("%d, %d,", sizeof(ch), sizeof('A'));
printf("%d, %d, ", sizeof(65), sizeof(++x));
printf("%d", x);
return 0;
}
//1,4,4,4,9


int main1(void)
{
int a = 3, b = 6;
printf("%d, ",a = b);
printf("%d, ",a == b);
printf("%d, ",a != b);
printf("%d, ",a=!b);
return 0;
}
//6,1,0,0

int main2(void)
{
int a = -10, b = 3, c = 0, d;
d = a++ || ++b && c++;
printf("%d, %d, %d, %d, ", a, b, c, d);
a = -10, b = 3, c = 0;
d = c++ && ++b || a++;
printf("%d, %d, %d, %d", a, b, c, d);
return 0;
}
//-9, 3, 0, 1, -9, 3, 1, 1

int main3(void)
{
int a = (1, 2, 3);
int b = (++a, ++a, ++a);
int c = (b++, b++, b++);
printf("\n%d, %d, %d", a, b, c);
return 0;
}
//6,9,8

/*
int main4()
{
printf("%d", ++4);
return 0;
}
//compile time error
*/
```

```
int main5(void)
{
int a =3, b = 6;
if(a = b)
printf("%d, %d", a, b);
else
printf("%d != %d", a, b);
return 0;
}
//6, 6

int main6(void)
{
int a =3, b = 6;
printf("%d ", a!=b?0?b:a:b);
return 0;
}
//3

int main7(void)
{
int k=1;
switch(k++)
{
default : printf("0");
case 1: case 2:printf("2");
case 3*0: printf("1");
}
return 0;
}
//21

int main8(void)
{
float i=1.5;
switch((int)i)
{
case 1  : printf("1");
case 2  : printf("2");
default : printf("0");
}
return 0;
}
//120

int main9(void)
{
int a =3;
while(a--)
{
printf("%d, ", a);
}
return 0;
}
//2, 1, 0,

int main10(void)
{
int a = 1000000L;
for(a=1; a<10; ++a);
{
printf("%d, ", a);
}
return 0;
}
//10,

int main11(void)
{
char i = 0;
do
{
printf("%d, ", i);
}while(i--);
return 0;
}
//0,

int main12(void)
{
int i, j;
for(i = 1; i <= 3; i+=2)
for(j = 1; j <= 3; j+=2)
printf(" %d, ", i+j);
return 0;
}
// 2,  4,  4,  6,

//int x = 3;
int main13(void)
{
int x = 4, i;
for(i = 0; i < x; ++i)
{
int x = 0; printf("%d ,", x++);
}
return 0;
}
//0 ,0 ,0 ,0 ,

int main14(void)
{
int main = 3;
printf("%o", main);
return 0;
}
//3
```

```
/*
int main = 3;
int main15(void)
{
printf("%o", main);
return 0;
}
error: 'main' redeclared as different kind of symbol
*/

/*
int main = 3;
int main(void)
{
printf("%o", main);
return 0;
}
//error: 'main' redeclared as different kind of symbol
*/

int num;
int function(int n)
{
int num = 10;
return num;
}
int main15(void)
{
printf("%d %d\n", num, function(20));
return 0;
}
//0,10

int no1 = 71, no2 = 17;
void swapping(int no1, int no2)
{
int temp = no2;
no2 = no1;
no1 = temp;
}
int main16(void)
{
int no1 = 17, no2 = 71;
printf("%d %d ", no1 , no2);
swapping(no1, no2);
printf("%d %d\n", no1, no2);
return 0;
}
//17 71 17 71

int main17(void)
{
int a = 1;
int *p = &a;
int *q = p;
*p = *p + *q;
printf("%d %d %d",*p,a,*q);
return 0;
}
//2 2 2


//int sunbeam(int **q)
//{
//int a = /*$$$*/;
//return a;
//}
//int main(void)
//{
//int a = 3;
//int *p = &a;
//a=sunbeam(&p);
//printf("%d", a);
//return 0;
//}
//error: expected expression before ';' token

int main18(void)
{
int a = 1;
printf("%d", ++a);
main18();
return 0;
}
//22222.......infinite loop

/*
int main19(void)
{
register char reg = 20;
char *ptr_reg = NULL;
ptr_reg = &reg;
*ptr_reg = 40;
printf("%d ", reg);
return 0;
}
// error: address of register variable 'reg' requested
*/

/*
int var = 1;
int main20(void)
{
auto int var, var1;
var = init();
for(var1 = 1 ; var1 <= 3 ; var1++)
{
```

```
printf("%d ", first(var));
printf("%d ", last(var));
}
return 0;
}
int init(void)
{
return (var);
}
int first(int var)
{
return var = var++;
}
int last(int var1)
{
static int var = 5;
return var1 = var--;
}
//22222.....2222222.....infinite loop after some time exit from loop automatically
*/

/*
//#define area(x) (x * x)
int main19(void)
{
printf("%d ",area(6 + 3));
return 0;
}
//27
*/

#define print_mul(x, y) printf("%d * %d = %d", x,y,x*y)
int main20(void)
{
int a = 3;
print_mul(a, 3);
return 0;
}
//3 * 3 = 9

#define greater(a, b) (a > b)? a : b
int main21(void)
{
int x = 3, y = 4;
if(greater(x, y))
printf("sun");
else
printf("beam");
return 0;
}
//sun

#define max(a, b) (a > b)? a : b
int main22()
```

```
{
int x = 3, y = 4, z;
z = max(++x, y++);
printf("%d, %d, %d", x, y, z);
return 0;
}
//4, 6, 5

/*
#define int char
int main23( void )
{
int* i = 65, j=56;
printf("sizeof(i)=%d sizeof(j)=%d",sizeof(i),sizeof(j));
return 0;
}
//sizeof(i)=8 sizeof(j)=1
*/

int main24(void)
{
int a[] = {1, 2, 3};
printf("%d, %d, ", sizeof(a), sizeof(a[-1]));
print_size(a);
return 0;
}
int print_size(int a[])
{
printf("%d, %d, ", sizeof(a), sizeof(a[3]));
return 0;
}
//12, 4, 8, 4,

/*
int main()
{
int a[] = {1, 2, 3};
++a;
printf("%d", a[2]);
return 0;
}
//error: lvalue required as increment operand
*/

int main25()
{
int a[] = {1, 2, 3};
f(a);
return 0;
}
int f(int a[])
{
++a;
printf("%d", a[-1]);
return 0;
}
//1

#include <string.h>
int main26()
{
char *s = "SunBeam"; int i; char * p = s;
for(i = 0; i <strlen(s); ++i, ++p)
printf("%c", p[-i]);
return 0;
}
//SSSSSSS

#include <string.h>
int main27()
{
char *s = "SunBeam"; int i; char * p = s;
for(i = 0; i <strlen(p); ++i, ++p)
printf("%c", *p++);
return 0;
}
//Sne

int main28(void)
{
char *s = "SunBeam";char p[7] = "SunBeam";
printf("%d, %d, %d, %d", sizeof(s), sizeof(p), sizeof(*p),
sizeof("sunbeam"));
return 0;
}
//8, 7, 1, 8

int main29(void)
{
char *arr[] = {"SunBeam", "DAC","WiMC", "Pune", "Karad"};
char **ptr = arr + 2;
printf("%s", ++ptr[arr-ptr] - 1);
return 0;
}
//SunBeam

/*
int main30(void)
{
int a[][2]={ 1, 2, 3, 4, 5, 6};
int *p = (int *)a;
printf("%d", p[5] – p[1]);
return 0;
}
// error: stray '\342' in program
*/

/*
```

```
int main31()
{
char s[3][128] = {"SunBeam", "DAC", "WIMC"};
char *p = s[1];
s[0] = p;
printf("%s", s[1]);
return 0;
}
//error: assignment to expression with array type
*/

/*
int main32(void)
{
char s[3][128] = {"SunBeam", "DAC", "WIMC"};
char *p = (char *)s[0];
printf("%s", s[++p - s]);
return 0;
}
//error: invalid operands to binary - (have 'char *' and 'char (*)[128]')
*/

int main33(void)
{
char s[3][128] = {"SunBeam", "DAC", "WIMC"};
char *p = (char *)s[2];
printf("%c%s", *p + *(p-1) - *p, ++p);
return 0;
}
//WIMC

int main34(void)
{
char s[3][128] = {"SunBeam", "DAC", "W I M C"};
char *p = (char *)s[2];
printf("%s", s[strlen(p) - strlen(s)]);
return 0;
}
//SunBeam

struct student
{
char *name[2];
int s;
char b;
}s = {"Sun", "Beam", 0, 1};
int main35()
{
printf("%d",sizeof(s));
return 0;
}
//24

/*
```

```
struct x
{
int i; int j; int k;
};
int main36(void)
{
struct x *p, arr[3];
p = &arr[0];
++p;
printf("%d", &(arr[0].j) - p);
return 0;
}
//error: invalid operands to binary - (have 'int *' and 'struct x *')
*/

int main37(void)
{
struct a
{
int a;
struct b
{
int a; char b;
} b;
} x[] = {1, 'a', 'x', 'A', 4, 'b'};
printf("%d", -x[1].a + x[0].b.a + x[1].b.a);
return 0;
}
//36

struct s1
{
struct {
int x;
struct s { int x;}s2;
struct s *s;}s3;
}y={ 10 };
int main38(void)
{
y.s3.s = &y.s3.s2;
printf("%d", y.s3.s->x);
return 0;
}
//0

union u
{
int x[3];
char y[5];
}u[3];
int main39()
{
printf("%d", sizeof(u));
return 0;
}
//36

/*
struct s
{
int x[10];
};
union u
{
int *x;
struct s *y[2];
}u;
int main40(void)
{
printf("%d", sizeof(u));
return 0;
}
//16
*/

enum color {red,green,yellow=1,dark};
int main41(void)
{
enum color x = green % dark;
printf("%d", yellow + dark/x - green * red);
return 0;
}
//3

int main42(void)
{
enum choice {CH1, CH2, CH3};
enum choice ch1, ch2, ch3;
ch1 = CH1;
ch2 = CH3;
ch3 = CH2;
printf("%d, %d, %d,", ch1, ch2, ch3);
return 0;
}
//0, 2, 1,

int main43(void)
{
enum choice {CH1, CH2, CH3};
enum choice ch1, ch2, ch3;
ch1 = CH1;
ch2 = CH3;
ch3 = ch2-ch1;
printf("%d %d", sizeof(enum choice), ch3);
return 0;
}//4 2

#include<stdlib.h>
```

```
int main44(void)
{
void *ptr=NULL;
ptr = malloc(10);
return 0;
}
//Any type of data

int main45(void)
{
char *ptr=NULL;
ptr = (char *)calloc(20,sizeof(char));
printf("%d bytes\n", sizeof(ptr));
free(ptr); ptr=NULL;
return 0;
}
//8 bytes

int main46(void)
{
char *ptr=NULL;
ptr = (char *)calloc(1,10);
strcpy(ptr, "Sunbeam");
ptr = (char *)realloc(ptr,20);
strcat(ptr," IT PARK");
printf("%c", (ptr[0]>=65 && ptr[0]<=90)? ptr[14]+32 : ptr[0]-32);
free(ptr);
ptr=NULL;
return 0;
}
//k

int main47()
{
int a = 10;
unsigned b;
b = sizeof(++a + a++ + ++a) + a;
printf("%d %d\n", a, b);
return 0;
}
//10 14

int main48()
{
int a, b;
printf("%d %d\n", printf("%d %d ", a, b), scanf("%d%d", &a, &b));
return 0;
}
//10 100 7 2

int main49()
{
int a = 0, b = 3, c = 6, d;
d = a++ && ++b || c++;
printf("%d %d %d %d\n", a, b, c, d);
return 0;
}
//1 3 7 1

int main50()
{
int a=3,b=4,c;
c=a+b/a*a*a;
printf("%d\n",c);
return 0;
}
//12

/*
int main(void)
{
int x = 10;
(x < 0) ? int x = 100 : {int x = 1000};
printf("%d", x);
return 0;
}
//Compile time error
*/

int main51(void)
{
int a=3, b=5, c=50;
a=b==c;
printf("%d",a);
}
//0

int main(void)
{
int a=14;
a += 7;
a -= 5;
a *= 7;
a ^= a;
printf("%d", a);
return 0;
}
//0
```

Computer Programming “C language”

1. Which of the following language is predecessor to C Programming Language?
a) A
b) B
c) BCPL
d) C++
Ans: c

2. C programming language was developed by
a) Dennis Ritchie
b) Ken Thompson
c) Bill Gates
d) Peter Norton
Ans: a

3. C was developed in the year ___
a) 1970
b) 1972
c) 1976
d) 1980
Ans: b

4. C is a ___ language
a) High Level
b) Low Level
c) Middle Level
d) Machine Level
Ans: c

5. C language is available for which of the following Operating Systems?
a) DOS
b) Windows
c) Unix
d) All of these
Ans: d

6. Which of the following symbol is used to denote a pre-processor statement?
a) !
b) #
c) ~
d) ;
Ans: b

7. Which of the following is a Scalar Data type
a) Float b) Union
c) Array d) Pointer
Ans: a

8. Which of the following are tokens in C?
a) Keywords b) Variables
c) Constants d) All of the above
Ans: d

9. What is the valid range of numbers for int type of data?
a) 0 to 256
b) -32768 to +32767
c) -65536 to +65536
d) No specific range
Ans: b

10. Which symbol is used as a statement terminator in C?
a) !
b) #
c) ~
d) ;
Ans: d

11. Which escape character can be used to begin a new line in C?
a) \a
b) \b
c) \m
d) \n
Ans: d

12. Which escape character can be used to beep from speaker in C?
a) \a
b) \b
c) \m
d) \n
Ans: a

13. Character constants should be enclosed between ___
a) Single quotes
b) Double quotes
c) Both a and
b d) None of these
Ans: a

14. String constants should be enclosed between ___
a) Single quotes
b) Double quotes
c) Both a and b
d) None of these
Ans: b

15. Which of the following is invalid?
a) ‘’ b) ““
c) ‘a’ d) ‘abc’
Ans: d

16. The maximum length of a variable in C is ___
a) 8 b) 16
c) 32 d) 64
Ans: a

17. What will be the maximum size of a float variable?
a) 1 byte
b) 2 bytes
c) 4 bytes
d) 8 bytes
Ans: c

18. What will be the maximum size of a double variable?
a) 1 byte
b) 4 bytes
c) 8 bytes
d) 16 bytes
Ans: c

19. A declaration float a,b; occupies ___ of memory
a) 1 byte
b) 4 bytes
c) 8 bytes
d) 16 bytes
Ans: c

20. The size of a String variable is
a) 1 byte
b) 8 bytes
c) 16 bytes
d) None of these
Ans: d

21. Which of the following is an example of compounded assignment statement?
a) a=5
b) a+=5
c) a=b=c
d) a=b
Ans: b

22. The operator && is an example for ___ operator.
a) Assignment
b) Increment
c) Logical
d) Rational
Ans: c

23. The operator & is used for
a) Bitwise AND
b) Bitwise OR
c) Logical AND
d) Logical OR
Ans: a

24. The operator / can be applied to

a) integer values	b) float values
c) double values	d) All of these

Ans: b

25. The equality operator is represented by
a) :=
b) .EQ.
c) =
d) ==
Ans: d

26. Operators have hierarchy. It is useful to know which operator
a) is most important
b) is used first
c) is faster
d) operates on large numbers
Ans: b

27. The bitwise AND operator is used for
a) Masking
b) Comparison
c) Division
d) Shifting bits
Ans: a

28. The bitwise OR operator is used to
a) set the desired bits to 1
b) set the desired bits to 0
c) divide numbers
d) multiply numbers
Ans: a

29. Which of the following operator has the highest precedence?
a)*
b) ==
c) =>
d) +
Ans: d

30. The associativity of ! operator is
a) Right to Left
b) Left to Right
c) (a) for Arithmetic and (b) for Relational
d) (a) for Relational and (b) for Arithmetic
Ans: a

31. Which operator has the lowest priority?
a) ++ b) %
c) + d) ||
Ans: d

32. Which operator has the highest priority?
a) ++ b) %
c) + d) ||
Ans: a

33. Operators have precedence. Precedence determines which operator is
a) faster
b) takes less memory
c) evaluated first
d) takes no arguments
Ans: c

34. Integer Division results in
a) Rounding the fractional part
b) Truncating the fractional part
c) Floating value
d) An Error is generated
Ans: b

35. Which of the following is a ternary operator?
a) ?:
b) *
c) sizeof
d) ^
Ans: a

36. What will be the output of the expression 11 ^ 5?
a) 5
b) 6
c) 11
d) None of these
Ans: d

37. The type cast operator is
a) (type)
b) cast()
c) (;;)
d) // " "
Ans: a

38. Explicit type conversion is known as
a) Casting
b) Conversion
c) Disjunction
d) Separation
Ans: a

39. The operator + in a+=4 means
a) a=a+4
b) a+4=a
c) a=4
d) a=4+4
Ans: a

40. p++ executes faster than p+1 because
a) p uses registers b) p++ is a single instruction
c) ++ is faster than + d) None of these
Ans: b

41. Which of the following statements is true?
a) C Library functions provide I/O facilities
b) C inherent I/O facilities
c) C doesn’t have I/O facilities
d) Both (a) and (c)
Ans: a

42. Header files in C contain
a) Compiler commands
b) Library functions
c) Header information of C programs
d) Operators for files
Ans: b

43. Which pair of functions below are used for single character I/O.
a) Getchar() and putchar()
b) Scanf() and printf()
c) Input() and output()
d) None of these
Ans: a

44. The printf() function retunes which value when an error occurs?
a) Positive value
b) Zero
c) Negative value
d) None of these
Ans: c

45. Identify the wrong statement
a) putchar(65)
b) putchar('x')
c) putchar("x")
d) putchar('\n')
Ans: c

46. Which of the following is charecter oriented console I/O function?
a) getchar() and putchar()
b) gets() and puts()
c) scanf() and printf()
d) fgets() and fputs()
Ans: a

47. The output of printf("%u", -1) is
a) -1 b) minimum int value
c) maxium int value d) Error message
Ans: c

48. An Ampersand before the name of a variable denotes
a) Actual Value b) Variable Name
c) Address d) Data Type
Ans: c

49. Symbolic constants can be defined using
a) # define
b) const
c) symbols
d) None of these
Ans: b

50. Null character is represented by
a) \n
b) \0
c) \o
d) \e
Ans: b

51. Which header file is essential for using strcmp() function?
a) string.h
b) strings.h
c) text.h
d) strcmp.h
Ans: a

52. malloc() function used in dynamic allocation is available in which header file?
a) stdio.h
b) stdlib.h
c) conio.h
d) mem.h
Ans: b

53. File manipulation functions in C are available in which header file?
a) streams.h
b) stdio.h
c) stdlib.h
d) files.h
Ans: d

54. C supports how many basic looping constructs
a) 2
b) 3
c) 4
d) 6
Ans: b

55. A statement differs from expression by terminating with a
a) ;
b) :
c) NULL
d) .
Ans: a

56. What should be the expression return value for a do-while to terminate
a) 1 b) 0
c) -1 d) NULL
Ans: b

57. Which among the following is a unconditional control structure
a) do-while
b) if-else
c) goto
d) for
Ans: c

58. Continue statement is used
a) to go to the next iteration in a loop
b) come out of a loop
c) exit and return to the main function
d) restarts iterations from beginning of loop
Ans: a

59. Which operator in C is called a ternary operator
a) if..then
b) ++
c) ?:
d) ()
Ans: c

60. Which of the following header file is required for strcpy() function?
a) string.h
b) strings.h
c) files.h
d) strcsspy()
Ans: a

61. The meaning of conversion character for data input is
a) Data item is a long integer
b) Data item is an unsigned decimal integer
c) Data item is a short integer
d) None of the above
Ans: c

62. The conversion characters for data input means that the data item is
a) An unsigned decimal integer
b) A short integer
c) A hexadecimal integer
d) A string followed by white space
Ans: b

63. An expression contains relational, assign. ment and arithmetic operators. If Parenthesis are not present, the order will be
a) Assignment, arithmetic, relational
b) Relational, arithmetic, assignment
c) Assignment, relational, arithmetic
d) Arithmetic, relational, assignment
Ans: d

64. Which of the following is a key word is used for a storage class
a) printf
b) external
c) auto
d) scanf
Ans: c

65. In the C language 'a’ represents
a) a digit
b) an integer
c) a character
d) a word
Ans: c

66. The number of the relational operators in the C language is
a) Four
b) Six
c) Three
d) One
Ans: b

67. A compound statement is a group of statements included between a pair of
a) double quote
b) curly braces
c) parenthesis
d) a pair of /’s
Ans: a

68. A Link is
a) a compiler
b) an active debugger
c) a C interpreter
d) a analyzing tool in C
Ans: d

69. The continue command cannot be used with
a) for
b) switch
c) do
d) while
Ans: a

70. In C, a Union is
a) memory location
b) memory store
c) memory screen
d) None of these
Ans: b

71. When the main function is called, it is called with the arguments
a) argc
b) argv
c) None of these
d) both a & b
Ans: d

72. A multidimensional array can be expressed in terms of
a) array of pointers rather than as pointers to a group of contiguous array
b) array without the group of contiguous array
c) data type arrays
d) None of these
Ans: a

73. C allows arrays of greater than two dimensions, who will determined this
a) programmer
b) compiler
c) parameter
d) None of these
Ans: b

74. A pointer to a pointer in a form of
a) multiple indirection
b) a chain of pointers
c) both a and b
d) None of these
Ans: c

75. Pointers are of
a) integer data type
b) character data type
c) unsigned integer data types
d) None of these
Ans: d

76. Maximum number of elements in the array declaration int a[5][8] is
a) 28
b) 32
c) 35
d) 40
Ans: d

77. If the size of the array is less than the number of initializers then,
a) Extra values are being ignored
b) Generates an error message
c) Size of Array is increased
d) Size is neglected when values are given
Ans: a

78. Array subscripts in C always start at
a) -1
b) 1
c) 0
d) Value provided by user
Ans: c

79. A Structure
a) can be read as a single entity
b) cannot be read as a single entity
c) can be displayed as a single entity
d) has member variables that cannot be read individually
Ans: b

80. Which is the correct way to declare a pointer?
a) int_ptr; b) int *ptr;
c) *int ptr; d) None of these.
Ans: b

81. If you want to exchange two rows in a two dimensional array, the fastest way is to:
a) Exchange the elements of the 2rows
b) Exchange the address of each element in the two row
c) Silence the address of the rows in an array of pointer and exchange the pointer
d) None of these.
Ans: c

82. A type cast is used to
a) Define a new data type
b) Force a value to be a particular variable type
c) Rename an old type
d) None of these.
Ans: b

83. Operator precedence determines which operator
a) Operators on the largest number
b) Is used first
c) Is most important
d) None of these.
Ans:b

84. If you don't initialize a static array, what will be the element set to?
a) Zero
b) A floating point
c) An undetermined value
d) None of these.
Ans: a

85. Which is more appropriate for reading in a multi-word string?
a) gets( )
b) Printf( )
c) scanf( )
d) puts ( ).
Ans: a

86. The process of translating a source program into machine language is a function of:

a) Compiler	b) Translator
c) Assembler	d) None of these.

Ans: a

87. Function argument can be

a) A structure member	b) A pointer variable
c) A complete structure	d) All of the above.

Ans: d

88. A "switch" statement is used to:
a) Switch between user defined functions in a program
b) Switch from one variable to another variable
c) Jump from one place to another in a program.
d) None of these.
Ans: d

89. Consider the foll statement: "using C language programmers can write their own library functions".
a) True
b) False
c) May be
d) None of these.
Ans: a

90. C is a ________ level programming language?
a) Low
b) High
c) Middle
d) None of these.
Ans: c

91. A function is a subroutine that may include one or more ____________ designed to perform a specific task.
a) Functions
b) Statements
c) Libraries
d) Data types.
Ans:b

92. What is used as a terminator in C?
a) ?
b) ;
c) :
d) _
Ans:b

93. Which function is necessary to exist in each & every program?
a) void
b) sum
c) main
d) None of these.
Ans: c

94. What is the answer of: 7%3
a) 2.5
b) 1
c) 2
d) 3
Ans: b

95. The _______ chars have values from -128 to 127.
a) signed b) unsigned
c) long d) none
Ans: a

96. What is the control character for "a single character".
a) %c b) %d
c) %i d) %p
Ans: a

97. What is the control character for "a decimal integer".
a) %c
b) %d
c) %i
d) %p
Ans: b

98. What is the control character for "a floating point number".
a) %c
b) %d
c) %i
d) %f
Ans: d

99. C supports the ______ statement to branch unconditionally from one point to another in the program.
a) continue
b) goto
c) break
d) for
Ans: b

100. The _______ is used to break out of the case statements.
a) continue
b) break
c) default
d) case
Ans: b

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

```
    #include <stdio.h>
    int main()
    {
        float f1 = 0.1;
        if (f1 == 0.1)
            printf("equal\n");
        else
            printf("not equal\n");
    }
```

a) equal
b) not equal
c) output depends on the compiler
d) error
View Answer
Answer: b
Explanation: 0.1 by default is of type double which has different representation than float resulting in inequality even after conversion.
Output:

```
$ cc pgm4.c
$ a.out
not equal
```

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

```
    #include <stdio.h>
    int main()
    {
        float f1 = 0.1;
        if (f1 == 0.1f)
            printf("equal\n");
        else
            printf("not equal\n");
    }
```

a) equal
b) not equal
c) output depends on compiler
d) error
View Answer
Answer: a
Explanation: 0.1f results in 0.1 to be stored in floating point representations.
Output:

```
$ cc pgm5.c
$ a.out
equal
```

3. What will be the output of the following C code on a 32-bit machine?

```
    #include <stdio.h>
    int main()
    {
        int x = 10000;
        double y = 56;
        int *p = &x;
        double *q = &y;
        printf("p and q are %d and %d", sizeof(p), sizeof(q));
        return 0;
    }
```

a) p and q are 4 and 4
b) p and q are 4 and 8
c) compiler error
d) p and q are 2 and 8
View Answer
Answer: a
Explanation: Size of any type of pointer is 4 on a 32-bit machine.
Output:

```
$ cc pgm6.c
$ a.out
p and q are 4 and 4
```

4. Which is correct with respect to the size of the data types?
a) char > int > float
b) int > char > float
c) char < int < double
d) double > char > int
View Answer
Answer: c
Explanation: char has less bytes than int and int has less bytes than double in any system

5. What will be the output of the following C code on a 64 bit machine?

```
    #include <stdio.h>
    union Sti
    {
        int nu;
        char m;
    };
    int main()
    {
        union Sti s;
        printf("%d", sizeof(s));
        return 0;
    }
```

a) 8
b) 5
c) 9
d) 4
View Answer
Answer: d
Explanation: Since the size of a union is the size of its maximum data type, here int is the largest data type. Hence the size of the union is 4.
Output:

```
$ cc pgm7.c
$ a.out
4
```

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

```
    #include <stdio.h>
    int main()
    {
        float x = 'a';
        printf("%f", x);
        return 0;
    }
```

a) a
b) run time error
c) a.0000000
d) 97.000000
View Answer
Answer: d
Explanation: Since the ASCII value of a is 97, the same is assigned to the float variable and printed.
Output:

```
$ cc pgm8.c
$ a.out
97.000000
```

7. Which of the data types has the size that is variable?
a) int
b) struct
c) float
d) double
View Answer
Answer: b
Explanation: Since the size of the structure depends on its fields, it has a variable size.

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

```
    #include <stdio.h>
    int main()
    {
        enum {ORANGE = 5, MANGO, BANANA = 4, PEACH};
        printf("PEACH = %d\n", PEACH);
    }
```

a) PEACH = 3
b) PEACH = 4
c) PEACH = 5
d) PEACH = 6
View Answer
Answer: c
Explanation: In enum, the value of constant is defined to the recent assignment from left.
Output:

```
$ cc pgm1.c
$ a.out
PEACH = 5
```

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

```
    #include <stdio.h>
    int main()
    {
        printf("C programming %s", "Class by\n%s Sanfoundry",
"WOW");
    }
```

a)

```
C programming Class by

WOW Sanfoundry
```

b) C programming Class by\n%s Sanfoundry
c)

```
C programming Class by

%s Sanfoundry
```

d) Compilation error
View Answer
Answer: c
Explanation: This program has only one %s within first double quotes, so it does not read the string “WOW”.
The %s along with the Sanfoundry is not read as a format modifier while new line character prints the new line.
Output:

```
$ cc pgm2.c
$ a.out
C programming Class by
%s Sanfoundry
```

3. In the following code snippet, character pointer str holds a reference to the string ____________

```
char *str =  "Sanfoundry.com\0" "training classes";
```

a) Sanfoundry.com
b) Sanfoundry.com\0training classes
c) Sanfoundry.comtraining classes
d) Invalid declaration
View Answer
Answer: b
Explanation: ‘\0’ is accepted as a char in the string. Even though strlen will give length of string “Sanfoundry.com”, in memory str is pointing to entire string including training classes.

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

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

a) a = 5
b) a = 10
c) Compilation error
d) Runtime error
View Answer
Answer: c
Explanation: The #define substitutes a with 10 without leaving any identifier, which results in Compilation error.
Output:

```
$ cc pgm3.c
pgm3.c: In function ‘main’:
pgm3.c:5: error: expected identifier or ‘(’ before numeric constant
```

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

```c
#include <stdio.h>
int main()
{
    int var = 010;
    printf("%d", var);
}
```

a) 2
b) 8
c) 9
d) 10
View Answer
Answer: b
Explanation: 010 is octal representation of 8.
Output:

```
$ cc pgm4.c
$ a.out
8
```

6. What will be the output of the following C function?

```c
#include <stdio.h>
enum birds {SPARROW, PEACOCK, PARROT};
enum animals {TIGER = 8, LION, RABBIT, ZEBRA};
int main()
{
    enum birds m = TIGER;
    int k;
    k = m;
    printf("%d\n", k);
    return 0;
}
```

a) 0
b) Compile time error
c) 1
d) 8
View Answer
Answer: d
Explanation: m is an integer constant, hence it is compatible.
Output:
```
$ cc pgm5.c
$ a.out
8
```

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

```c
    #include <stdio.h>
    #define MAX 2
    enum bird {SPARROW = MAX + 1, PARROT = SPARROW + MAX};
    int main()
    {
        enum bird b = PARROT;
        printf("%d\n", b);
        return 0;
    }
```

a) Compilation error
b) 5
c) Undefined value
d) 2
View Answer
Answer: b
Explanation: MAX value is 2 and hence PARROT will have value 3 + 2.
Output:
```
$ cc pgm6.c
$ a.out
5
```

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

```c
    #include <stdio.h>
    #include <string.h>
    int main()
    {
        char *str = "x";
        char c = 'x';
        char ary[1];
        ary[0] = c;
        printf("%d %d", strlen(str), strlen(ary));
        return 0;
    }
```

a) 1 1
b) 2 1
c) 2 2
d) 1 (undefined value)
View Answer

Answer: d
Explanation: str is null terminated, but ary is not null terminated.
Output:
```
$ cc pgm7.c
$ a.out
1 5
```

1. enum types are processed by __________
a) Compiler
b) Preprocessor
c) Linker
d) Assembler
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        printf("sanfoundry\rclass\n");
        return 0;
    }
```

a) sanfoundryclass
b)

```
sanfoundry
class
```

c) classundry
d) sanfoundry
View Answer
Answer: c
Explanation: r is carriage return and moves the cursor back. sanfo is replaced by class.
Output:
```
$ cc pgm8.c
$ a.out
classundry
```

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

```
    #include <stdio.h>
    int main()
    {
        printf("sanfoundry\r\nclass\n");
        return 0;
    }
```

a) sanfoundryclass
b)

```
sanfoundry

class
```

c) classundry
d) sanfoundry
View Answer
Answer: b
Explanation: rn combination makes the cursor move to the next line.
Output:

```
$ cc pgm9.c
$ a.out
sanfoundry
class
```

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

```
    #include <stdio.h>
    int main()
    {
        const int p;
        p = 4;
        printf("p is %d", p);
        return 0;
    }
```

a) p is 4
b) Compile time error
c) Run time error
d) p is followed by a garbage value
View Answer
Answer: b
Explanation: Since the constant variable has to be declared and defined at the same time, not doing it results in an error.
Output:

```
$ cc pgm10.c
pgm10.c: In function ‘main’:
pgm10.c:5: error: assignment of read-only variable ‘p’
```

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

```
    #include <stdio.h>
    void main()
    {
        int k = 4;
        int *const p = &k;
        int r = 3;
        p = &r;
        printf("%d", p);
    }
```

a) Address of k
b) Address of r
c) Compile time error

d) Address of k + address of r
View Answer
Answer: c
Explanation: Since the pointer p is declared to be constant, trying to assign it with a new value results in an error.
Output:

```
$ cc pgm11.c
pgm11.c: In function ‘main’:
pgm11.c:7: error: assignment of read-only variable ‘p’
pgm11.c:8: warning: format ‘%d’ expects type ‘int’, but argument 2 has type ‘int * const’
```

6. Which of the following statement is false?
a) Constant variables need not be defined as they are declared and can be defined later
b) Global constant variables are initialized to zero
c) const keyword is used to define constant values
d) You cannot reassign a value to a constant variable
View Answer
Answer: a
Explanation: Since the constant variable has to be declared and defined at the same time, not doing it results in an error.

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

```
    #include <stdio.h>
    void main()
    {
        int const k = 5;
        k++;
        printf("k is %d", k);
    }
```

a) k is 6
b) Error due to const succeeding int
c) Error, because a constant variable can be changed only twice
d) Error, because a constant variable cannot be changed
View Answer
Answer: d
Explanation: Constant variable has to be declared and defined at the same time. Trying to change it results in an error.
Output:

```
$ cc pgm12.c
pgm12.c: In function ‘main’:
pgm12.c:5: error: increment of read-only variable ‘k’
```

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

```
    #include <stdio.h>
    int const print()
    {
        printf("Sanfoundry.com");
        return 0;
    }
    void main()
    {
        print();
    }
```

a) Error because function name cannot be preceded by const
b) Sanfoundry.com
c) Sanfoundry.com is printed infinite times
d) Blank screen, no output
View Answer
Answer: b
Explanation: None.
Output:
```
$ cc pgm13.c
$ a.out
```

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

```
    #include <stdio.h>
    void foo(const int *);
    int main()
    {
        const int i = 10;
        printf("%d ", i);
        foo(&i);
        printf("%d", i);

    }
    void foo(const int *i)
    {
        *i = 20;
    }
```

a) Compile time error
b) 10 20
c) Undefined value
d) 10
View Answer
Answer: a
Explanation: Cannot change a const type value.
Output:
```
$ cc pgm1.c
pgm1.c: In function ‘foo’:
pgm1.c:13: error: assignment of read-only location ‘*i’
```

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

```
    #include <stdio.h>
    int main()
    {
        const int i = 10;
        int *ptr = &i;
        *ptr = 20;
        printf("%d\n", i);
        return 0;
    }
```

a) Compile time error
b) Compile time warning and printf displays 20
c) Undefined behaviour

d) 10
View Answer
Answer: b
Explanation: Changing const variable through non-constant pointers invokes compiler warning.
Output:
```
$ cc pgm2.c
pgm2.c: In function ‘main’:
pgm2.c:5: warning: initialization discards qualifiers from pointer target type
$ a.out
20
```

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

```
    #include <stdio.h>
    int main()
    {
        j = 10;
        printf("%d\n", j++);
        return 0;
    }
```

a) 10
b) 11
c) Compile time error
d) 0
View Answer
Answer: c
Explanation: Variable j is not defined.
Output:
```
$ cc pgm3.c
pgm3.c: In function ‘main’:
pgm3.c:4: error: ‘j’ undeclared (first use in this function)
pgm3.c:4: error: (Each undeclared identifier is reported only once
pgm3.c:4: error: for each function it appears in.)
```

4. Will the following C code compile without any error?

```
    #include <stdio.h>
    int main()
    {
        for (int k = 0; k < 10; k++);
            return 0;
    }
```

a) Yes
b) No
c) Depends on the C standard implemented by compilers
d) Error
View Answer
Answer: c
Explanation: Compilers implementing C90 do not allow this, but compilers implementing C99 allow it.
Output:
```
$ cc pgm4.c
```

```
pgm4.c: In function 'main':
pgm4.c:4: error: 'for' loop initial declarations are only allowed in C99 mode
pgm4.c:4: note: use option -std=c99 or -std=gnu99 to compile your code
```

5. Will the following C code compile without any error?

```
    #include <stdio.h>
    int main()
    {
        int k;
        {
            int k;
            for (k = 0; k < 10; k++);
        }
    }
```

a) Yes
b) No
c) Depends on the compiler
d) Depends on the C standard implemented by compilers
View Answer
Answer: a
Explanation: There can be blocks inside the block. But within a block, variables have only block scope.
Output:

```
$ cc pgm5.c
```

6. Which of the following declaration is not supported by C?
a) String str;
b) char *str;
c) float str = 3e2;
d) Both String str; & float str = 3e2;
View Answer
Answer: a
Explanation: It is legal in Java, but not in C.

7. Which of the following format identifier can never be used for the variable var?

```
    #include <stdio.h>
    int main()
    {
        char *var = "Advanced Training in C by Sanfoundry.com";
    }
```

a) %f
b) %d
c) %c
d) %s
View Answer
Answer: a
Explanation: %c can be used to print the indexed position.
%d can still be used to display its ASCII value.
%s is recommended.
%f cannot be used for the variable var.

Here is a listing of C questions and puzzles on “Declarations” along with answers, explanations and/or solutions:

1. Which of the following declaration is illegal?
a) char *str = “Best C programming classes by Sanfoundry”;
b) char str[] = “Best C programming classes by Sanfoundry”;
c) char str[20] = “Best C programming classes by Sanfoundry”;
d) char[] str = “Best C programming classes by Sanfoundry”;
View Answer
Answer: d
Explanation: char[] str is a declaration in Java, but not in C.

2. Which keyword is used to prevent any changes in the variable within a C program?
a) immutable
b) mutable
c) const
d) volatile
View Answer
Answer: c
Explanation: const is a keyword constant in C program.

3. Which of the following is not a pointer declaration?
a) char a[10];
b) char a[] = {‘1’, ‘2’, ‘3’, ‘4’};
c) char *str;
d) char a;
View Answer
Answer: d
Explanation: Array declarations are pointer declarations.

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

```
    #include <stdio.h>
    void main()
    {
        int k = 4;
        float k = 4;
        printf("%d", k)
    }
```

a) Compile time error
b) 4
c) 4.0000000
d) 4.4
View Answer
Answer: a
Explanation: Since the variable k is defined both as integer and as float, it results in an error.
Output:

```
$ cc pgm8.c
pgm8.c: In function ‘main’:
pgm8.c:5: error: conflicting types for ‘k’
pgm8.c:4: note: previous definition of ‘k’ was here
```

pgm8.c:6: warning: format ‘%d’ expects type ‘int’, but argument 2 has type ‘double’
pgm8.c:7: error: expected ‘;’ before ‘}’ token

5. Which of the following statement is false?
a) A variable defined once can be defined again with different scope
b) A single variable cannot be defined with two different types in the same scope
c) A variable must be declared and defined at the same time
d) A variable refers to a location in memory
View Answer
Answer: c
Explanation: It is not an error if the variable is declared and not defined. For example – extern declarations.

6. A variable declared in a function can be used in main().
a) True
b) False
c) True if it is declared static
d) None of the mentioned
View Answer
Answer: b
Explanation: Since the scope of the variable declared within a function is restricted only within that function, so the above statement is false.

7. The name of the variable used in one function cannot be used in another function.
a) True
b) False
View Answer
Answer: b
Explanation: Since the scope of the variable declared within a function is restricted only within that function, the same name can be used to declare another variable in another function.

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

```
    #include <stdio.h>
    int main()
    {
        int i = -3;
        int k = i % 2;
        printf("%d\n", k);
    }
```

a) Compile time error
b) -1
c) 1
d) Implementation defined
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int i = 3;
        int l = i / -2;
        int k = i % -2;
        printf("%d %d\n", l, k);
        return 0;
    }
```

a) Compile time error
b) -1 1
c) 1 -1
d) Implementation defined
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int i = 5;
        i = i / 3;
        printf("%d\n", i);
        return 0;
    }
```

a) Implementation defined
b) 1
c) 3
d) Compile time error
View Answer
Answer: b
Explanation: None.

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

```
   #include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d\n", i);
        return 0;
    }
```

a) Implementation defined
b) -1
c) -3
d) Compile time error
View Answer
Answer: b
Explanation: None.

5. What will be the final value of x in the following C code?

```
    #include <stdio.h>
    void main()
    {
        int x = 5 * 9 / 3 + 9;
    }
```

a) 3.75
b) Depends on compiler
c) 24
d) 3
View Answer
Answer: c
Explanation: None.

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

```
    #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
View Answer
Answer: d
Explanation: None.

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

```
    #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
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int a = 3;
        int b = ++a + a++ + --a;
        printf("Value of b is %d", b);
    }
```

a) Value of x is 12
b) Value of x is 13
c) Value of x is 10
d) Undefined behaviour
View Answer
Answer: d
Explanation: None.

2. What is the precedence of arithmetic operators (from highest to lowest)?
a) %, *, /, +, –
b) %, +, /, *, –
c) +, -, %, *, /
d) %, +, -, *, /
View Answer
Answer: a
Explanation: None.

3. Which of the following is not an arithmetic operation?
a) a * = 10;
b) a / = 10;
c) a ! = 10;
d) a % = 10;
View Answer
Answer: c
Explanation: None.

4. Which of the following data type will throw an error on modulus operation(%)?
a) char
b) short
c) int
d) float
View Answer
Answer: d
Explanation: None.

5. Which among the following are the fundamental arithmetic operators, i.e, performing the desired operation can be done using that operator only?
a) +, –
b) +, -, %
c) +, -, *, /
d) +, -, *, /, %
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int a = 10;
        double b = 5.6;
        int c;
        c = a + b;
        printf("%d", c);
    }
```

a) 15
b) 16
c) 15.6
d) 10
View Answer
Answer: a
Explanation: None.

7. 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
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 1, y = 0, z = 5;
        int a = x && y || z++;
        printf("%d", z);
    }
```

a) 6
b) 5
c) 0
d) Varies
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 1, y = 0, z = 5;
        int a = x && y && z++;
        printf("%d", z);
    }
```

a) 6
b) 5
c) 0
d) Varies
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 1, y = 0, z = 3;
        x > y ? printf("%d", z) : return z;
    }
```

a) 3
b) 1
c) Compile time error
d) Run time error
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 1, z = 3;
        int y = x << 3;
        printf(" %d\n", y);
    }
```

a) -2147483648
b) -1
c) Run time error
d) 8
View Answer
Answer: d
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 0, y = 2, z = 3;
        int a = x & y | z;
        printf("%d", a);
    }
```

a) 3
b) 0

c) 2
d) Run time error
View Answer
Answer: a
Explanation: None.

6. What will be the final value of j in the following C code?

```
    #include <stdio.h>
    int main()
    {
        int i = 0, j = 0;
        if (i && (j = i + 10))
            //do something
            ;
    }
```

a) 0
b) 10
c) Depends on the compiler
d) Depends on language standard
View Answer
Answer: a
Explanation: None.

7. What will be the final value of j in the following C code?

```
    #include <stdio.h>
    int main()
    {
        int i = 10, j = 0;
        if (i || (j = i + 10))
            //do something
            ;
    }
```

a) 0
b) 20
c) Compile time error
d) Depends on language standard
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int i = 1;
        if (i++ && (i == 1))
            printf("Yes\n");
        else
            printf("No\n");
    }
```

a) Yes
b) No
c) Depends on the compiler
d) Depends on the standard
View Answer
Answer: b
Explanation: None.

1. Are logical operator sequence points?
a) True
b) False
c) Depends on the compiler
d) Depends on the standard
View Answer
Answer: a
Explanation: None.

2. Do logical operators in the C language are evaluated with the short circuit?
a) True
b) False
c) Depends on the compiler
d) Depends on the standard
View Answer
Answer: a
Explanation: None.

3. What is the result of logical or relational expression in C?
a) True or False
b) 0 or 1
c) 0 if an expression is false and any positive number if an expression is true
d) None of the mentioned
View Answer
Answer: b
Explanation: None.

4. What will be the final value of d in the following C code?

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

a) Syntax error
b) 1
c) 5
d) 10
View Answer
Answer: b
Explanation: None.

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

```
#include <stdio.h>
int main()
{
    int a = 10, b = 5, c = 3;
    b != !a;
    c = !!a;
    printf("%d\t%d", b, c);
}
```

a) 5 1
b) 0 3
c) 5 3
d) 1 1
View Answer
Answer: a
Explanation: None.

6. Which among the following is NOT a logical or relational operator?
a) !=
b) ==
c) ||
d) =
View Answer
Answer: d
Explanation: None.

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

```
#include <stdio.h>
int main()
{
    int a = 10;
    if (a == a--)
        printf("TRUE 1\t");
    a = 10;
    if (a == --a)
        printf("TRUE 2\t");
}
```

a) TRUE 1
b) TRUE 2
c) TRUE 1 TRUE 2
d) Compiler Dependent
View Answer
Answer: d
Explanation: This is a sequence point problem and hence the result will be implementation dependent.

8. Relational operators cannot be used on ____________
a) structure
b) long
c) strings
d) float
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        float x = 0.1;
        if (x == 0.1)
            printf("Sanfoundry");
        else
            printf("Advanced C Classes");
    }
```

a) Advanced C Classes
b) Sanfoundry
c) Run time error
d) Compile time error
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        float x = 0.1;
        printf("%d, ", x);
        printf("%f", x);
    }
```

a) 0.100000, junk value
b) Junk value, 0.100000
c) 0, 0.100000
d) 0, 0.999999
View Answer
Answer: b
Explanation: None.

3. What will be the output of the following C code? (Initial values: x= 7, y = 8)

```
    #include <stdio.h>
    void main()
    {
        float x;
        int y;
        printf("enter two numbers \n", x);
        scanf("%f %f", &x, &y);
        printf("%f, %d", x, y);
    }
```

a) 7.000000, 7
b) Run time error
c) 7.000000, junk
d) Varies
View Answer

Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        double x = 123828749.66;
        int y = x;
        printf("%d\n", y);
        printf("%lf\n", y);
    }
```

a) 0, 0.0
b) 123828749, 123828749.66
c) 12382874, 12382874.0
d) 123828749, 0.000000
View Answer
Answer: d
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 97;
        char y = x;
        printf("%c\n", y);
    }
```

a) a
b) b
c) 97
d) Run time error
View Answer
Answer: a
Explanation: None.

6. When double is converted to float, then the value is?
a) Truncated
b) Rounded
c) Depends on the compiler
d) Depends on the standard
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        unsigned int i = 23;
        signed char c = -23;
        if (i > c)
            printf("Yes\n");
        else if (i < c)
            printf("No\n");
    }
```

a) Yes
b) No
c) Depends on the compiler
d) Depends on the operating system
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int i = 23;
        char c = -23;
        if (i < c)
            printf("Yes\n");
        else
            printf("No\n");
    }
```

a) Yes
b) No
c) Depends on the compiler
d) Depends on the standard
View Answer
Answer: b
Explanation: None.

1. function tolower(c) defined in library <ctype.h> works for ___________
a) Ascii character set
b) Unicode character set
c) Ascii and utf-8 but not EBCDIC character set
d) Any character set
View Answer
Answer: d
Explanation: None.

2. What will be the output of the following C code considering the size of a short int is 2, char is 1 and int is 4 bytes?

```
    #include <stdio.h>
    int main()
    {
        short int i = 20;
        char c = 97;
        printf("%d, %d, %d\n", sizeof(i), sizeof(c), sizeof(c + 
i));
        return 0;
    }
```

a) 2, 1, 2
b) 2, 1, 1
c) 2, 1, 4
d) 2, 2, 8
View Answer
Answer: c
Explanation: None.

3. Which type of conversion is NOT accepted?
a) From char to int
b) From float to char pointer
c) From negative int to char
d) From double to char
View Answer
Answer: b
Explanation: Conversion of a float to pointer type is not allowed.

4. What will be the data type of the result of the following operation?

```
(float)a * (int)b / (long)c * (double)d
```

a) int
b) long
c) float
d) double
View Answer
Answer: d
Explanation: None.

5. Which of the following type-casting have chances for wrap around?
a) From int to float
b) From int to char
c) From char to short
d) From char to int
View Answer
Answer: b
Explanation: None.

6. Which of the following typecasting is accepted by C?
a) Widening conversions
b) Narrowing conversions
c) Widening & Narrowing conversions
d) None of the mentioned
View Answer
Answer: c
Explanation: None.

7. When do you need to use type-conversions?
a) The value to be stored is beyond the max limit
b) The value to be stored is in a form not supported by that data type
c) To reduce the memory in use, relevant to the value
d) All of the mentioned
View Answer

Answer: d
Explanation: None.

1. What is the difference between the following 2 codes?

```
    #include <stdio.h> //Program 1
    int main()
    {
        int d, a = 1, b = 2;
        d =  a++ + ++b;
        printf("%d %d %d", d, a, b);
    }
```

```
    #include <stdio.h> //Program 2
    int main()
    {
        int d, a = 1, b = 2;
        d =  a++ +++b;
        printf("%d %d %d", d, a, b);
    }
```

a) No difference as space doesn't make any difference, values of a, b, d are same in both the case
b) Space does make a difference, values of a, b, d are different
c) Program 1 has syntax error, program 2 is not
d) Program 2 has syntax error, program 1 is not
View Answer
Answer: d
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int a = 1, b = 1, c;
        c = a++ + b;
        printf("%d, %d", a, b);
    }
```

a) a = 1, b = 1
b) a = 2, b = 1
c) a = 1, b = 2
d) a = 2, b = 2
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int a = 1, b = 1, d = 1;
        printf("%d, %d, %d", ++a + ++a+a++, a++ + ++b, ++d + d++ +
 a++);
    }
```

a) 15, 4, 5
b) 9, 6, 9
c) 9, 3, 5
d) Undefined (Compiler Dependent)
View Answer
Answer: d
Explanation: None.

4. For which of the following, “PI++;” code will fail?
a) #define PI 3.14
b) char *PI = “A”;
c) float PI = 3.14;
d) none of the Mentioned
View Answer
Answer: a
Explanation: None.

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

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

a) a = 10, b = 9
b) a = 10, b = 8
c) a = 5, b = 9
d) a = 5, b = 8
View Answer
Answer: c
Explanation: None.

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

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

a) 0
b) 1
c) 2
d) Compile time error
View Answer
Answer: b
Explanation: None.

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

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

a) 6
b) 5
c) 4
d) Compile time error
View Answer
Answer: a
Explanation: None.

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

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

a) = operator is not a sequence point
b) ++ operator may return value with or without side effects
c) it can be evaluated as (i++)+i or i+(++i)
d) = operator is a sequence point
View Answer
Answer: a
Explanation: None.

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

```
•    #include <stdio.h>
•    int main()
•    {
•        int i = 0;
•        int x = i++, y = ++i;
•        printf("%d % d\n", x, y);
•        return 0;
•    }
```

a) 0, 2
b) 0, 1
c) 1, 2
d) Undefined
View Answer
Answer: a
Explanation: None.

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

```
•    #include <stdio.h>
•    int main()
```

```
    {
        int i = 10;
        int *p = &i;
        printf("%d\n", *p++);
    }
```

a) 10
b) 11
c) Garbage value
d) Address of i
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 97;
        int y = sizeof(x++);
        printf("X is %d", x);
    }
```

a) X is 97
b) X is 98
c) X is 99
d) Run time error
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 4, y, z;
        y = --x;
        z = x--;
        printf("%d%d%d", x,  y, z);
    }
```

a) 3 2 3
b) 2 3 3
c) 3 2 2
d) 2 3 4
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 4;
```

```
•           int *p = &x;
•           int *k = p++;
•           int r = p - k;
•           printf("%d", r);
•       }
```

a) 4
b) 8
c) 1
d) Run time error
View Answer
Answer: c
Explanation: None.

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

```
•  #include <stdio.h>
•       void main()
•       {
•           int a = 5, b = -7, c = 0, d;
•           d = ++a && ++b || ++c;
•           printf("\n%d%d%d%d", a,  b, c, d);
•       }
```

a) 6 -6 0 0
b) 6 -5 0 1
c) -6 -6 0 1
d) 6 -6 0 1
View Answer
Answer: d
Explanation: None.

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

```
•       #include <stdio.h>
•       void main()
•       {
•           int a = -5;
•           int k = (a++, ++a);
•           printf("%d\n", k);
•       }
```

a) -4
b) -5
c) 4
d) -3
View Answer
Answer: d
Explanation: None.

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

```
•       #include <stdio.h>
•       int main()
•       {
•           int c = 2 ^ 3;
•           printf("%d\n", c);
```

```
    }
```

a) 1
b) 8
c) 9
d) 0
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        unsigned int a = 10;
        a = ~a;
        printf("%d\n", a);
    }
```

a) -9
b) -10
c) -11
d) 10
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        if (7 & 8)
        printf("Honesty");
            if ((~7 & 0x000f) == 8)
                printf("is the best policy\n");
    }
```

a) Honesty is the best policy
b) Honesty
c) is the best policy
d) No output
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int a = 2;
        if (a >> 1)
           printf("%d\n", a);
    }
```

a) 0
b) 1
c) 2
d) No Output
View Answer
Answer: c
Explanation: None.

5. Comment on the output of the following C code.

```
    #include <stdio.h>
    int main()
    {
        int i, n, a = 4;
        scanf("%d", &n);
        for (i = 0; i < n; i++)
            a = a * 2;
    }
```

a) Logical Shift left
b) No output
c) Arithmetic Shift right
d) Bitwise exclusive OR
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 97;
        int y = sizeof(x++);
        printf("x is %d", x);
    }
```

a) x is 97
b) x is 98
c) x is 99
d) Run time error
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 4, y, z;
        y = --x;
        z = x--;
        printf("%d%d%d", x, y, z);
    }
```

a) 3 2 3
b) 2 2 3
c) 3 2 2
d) 2 3 3
View Answer
Answer: d
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 4;
        int *p = &x;
        int *k = p++;
        int r = p - k;
        printf("%d", r);
    }
```

a) 4
b) 8
c) 1
d) Run time error
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int a = 5, b = -7, c = 0, d;
        d = ++a && ++b || ++c;
        printf("\n%d%d%d%d", a, b, c, d);
    }
```

a) 6 -6 0 0
b) 6 -5 0 1
c) -6 -6 0 1
d) 6 -6 0 1
View Answer
Answer: d
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int a = -5;
        int k = (a++, ++a);
        printf("%d\n", k);
    }
```

a) -3
b) -5
c) 4
d) Undefined
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 2;
        x = x << 1;
        printf("%d\n", x);
    }
```

a) 4
b) 1
c) Depends on the compiler
d) Depends on the endianness of the machine
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = -2;
        x = x >> 1;
        printf("%d\n", x);
    }
```

a) 1
b) -1
c) 2 ^{31} – 1 considering int to be 4 bytes
d) Either -1 or 1
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        if (~0 == 1)
            printf("yes\n");
        else
            printf("no\n");
    }
```

a) yes
b) no
c) compile time error
d) undefined
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = -2;
        if (!0 == 1)
            printf("yes\n");
        else
            printf("no\n");
    }
```

a) yes
b) no
c) run time error
d) undefined
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int y = 0;
        if (1 |(y = 1))
            printf("y is %d\n", y);
        else
            printf("%d\n", y);

    }
```

a) y is 1
b) 1
c) run time error
d) undefined
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int y = 1;
        if (y & (y = 2))
```

```
                printf("true %d\n", y);
            else
                printf("false %d\n", y);

    }
```

a) true 2
b) false 2
c) either true 2 or false 2
d) true 1
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 0;
        if (x = 0)
            printf("Its zero\n");
        else
            printf("Its not zero\n");
    }
```

a) Its not zero
b) Its zero
c) Run time error
d) None
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int k = 8;
        int x = 0 == 1 && k++;
        printf("%d%d\n", x, k);
    }
```

a) 0 9
b) 0 8
c) 1 8
d) 1 9
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        char a = 'a';
        int x = (a % 10)++;
        printf("%d\n", x);
    }
```

a) 6
b) Junk value
c) Compile time error
d) 7
View Answer
Answer: c
Explanation: None.

4. What will be the output of the following C code snippet?

```
    #include <stdio.h>
    void main()
    {
        1 < 2 ? return 1: return 2;
    }
```

a) returns 1
b) returns 2
c) Varies
d) Compile time error
View Answer
Answer: d
Explanation: None.

5. What will be the output of the following C code snippet?

```
    #include <stdio.h>
    void main()
    {
        unsigned int x = -5;
        printf("%d", x);
    }
```

a) Run time error
b) Aries
c) -5
d) 5
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 2, y = 1;
        x *= x + y;
        printf("%d\n", x);
        return 0;
    }
```

a) 5
b) 6
c) Undefined behaviour
d) Compile time error
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 2, y = 2;
        x /= x / y;
        printf("%d\n", x);
        return 0;
    }
```

a) 2
b) 1
c) 0.5
d) Undefined behaviour
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 1, y = 0;
        x &&= y;
        printf("%d\n", x);
    }
```

a) Compile time error
b) 1
c) 0
d) Undefined behaviour
View Answer
Answer: a
Explanation: None.

1. What is the type of the following assignment expression if x is of type float and y is of type int?

```
 y = x + y;
```

a) int
b) float
c) there is no type for an assignment expression
d) double
View Answer

Answer: a
Explanation: None.

2. What will be the value of the following assignment expression?

(x = foo())!= 1 considering foo() returns 2

a) 2
b) True
c) 1
d) 0
View Answer
Answer: a
Explanation: None.

3. Operation “a = a * b + a” can also be written as ___________
a) a *= b + 1;
b) (c = a * b)!=(a = c + a);
c) a = (b + 1)* a;
d) All of the mentioned
View Answer
Answer: d
Explanation: None.

4. What will be the final value of c in the following C statement? (Initial value: c = 2)

```
c <<= 1;
```

a) c = 1;
b) c = 2;
c) c = 3;
d) c = 4;
View Answer
Answer: d
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int a = 1, b = 2;
        a += b -= a;
        printf("%d %d", a, b);
    }
```

a) 1 1
b) 1 2
c) 2 1
d) 2 2
View Answer
6. What will be the output of the following C code?

```
    #include <stdio.h>
    int main()
    {
        int a = 4, n, i, result = 0;
        scanf("%d", n);
        for (i = 0;i < n; i++)
        result += a;
    }
```

a) Addition of a and n
b) Subtraction of a and n
c) Multiplication of a and n
d) Division of a and n
View Answer
Answer: c
Explanation: None.

7. Which of the following is an invalid assignment operator?
a) a %= 10;
b) a /= 10;
c) a |= 10;
d) None of the mentioned
View Answer
Answer: d
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 2, y = 0;
        int z = (y++) ? y == 1 && x : 0;
        printf("%d\n", z);
        return 0;
    }
```

a) 0
b) 1
c) Undefined behaviour
d) Compile time error
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 1;
        int y =  x == 1 ? getchar(): 2;
        printf("%d\n", y);
    }
```

a) Compile time error
b) Whatever character getchar function returns
c) Ascii value of character getchar function returns

d) 2
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 1;
        short int i = 2;
        float f = 3;
        if (sizeof((x == 2) ? f : i) == sizeof(float))
            printf("float\n");
        else if (sizeof((x == 2) ? f : i) == sizeof(short int))
            printf("short int\n");
    }
```

a) float
b) short int
c) Undefined behaviour
d) Compile time error
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int a = 2;
        int b = 0;
        int y = (b == 0) ? a :(a > b) ? (b = 1): a;
        printf("%d\n", y);
    }
```

a) Compile time error
b) 1
c) 2
d) Undefined behaviour
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int y = 1, x = 0;
        int l = (y++, x++) ? y : x;
        printf("%d\n", l);
    }
```

a) 1
b) 2
c) Compile time error
d) Undefined behaviour
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int k = 8;
        int m = 7;
        int z = k < m ? k++ : m++;
        printf("%d", z);
    }
```

a) 7
b) 8
c) Run time error
d) 15
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int k = 8;
        int m = 7;
        int z = k < m ? k = m : m++;
        printf("%d", z);
    }
```

a) Run time error
b) 7
c) 8
d) Depends on compiler
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        1 < 2 ? return 1 : return 2;
    }
```

a) returns 1
b) returns 2

c) Varies
d) Compile time error
View Answer
Answer: d
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 2, y = 0;
        int z = (y++) ? y == 1 && x : 0;
        printf("%d\n", z);
        return 0;
    }
```

a) 0
b) 1
c) Undefined behaviour
d) Compile time error
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 1;
        int y =  x == 1 ? getchar(): 2;
        printf("%d\n", y);
    }
```

a) Compile time error
b) Whatever character getchar function returns
c) Ascii value of character getchar function returns
d) 2
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 1;
        short int i = 2;
        float f = 3;
        if (sizeof((x == 2) ? f : i) == sizeof(float))
            printf("float\n");
        else if (sizeof((x == 2) ? f : i) == sizeof(short int))
            printf("short int\n");
    }
```

a) float
b) short int
c) Undefined behaviour
d) Compile time error
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int a = 2;
        int b = 0;
        int y = (b == 0) ? a :(a > b) ? (b = 1): a;
        printf("%d\n", y);
    }
```

a) Compile time error
b) 1
c) 2
d) Undefined behaviour
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int y = 1, x = 0;
        int l = (y++, x++) ? y : x;
        printf("%d\n", l);
    }
```

a) 1
b) 2
c) Compile time error
d) Undefined behaviour
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int k = 8;
        int m = 7;
        int z = k < m ? k++ : m++;
        printf("%d", z);
    }
```

a) 7
b) 8
c) Run time error
d) 15
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int k = 8;
        int m = 7;
        int z = k < m ? k = m : m++;
        printf("%d", z);
    }
```

a) Run time error
b) 7
c) 8
d) Depends on compiler
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        1 < 2 ? return 1 : return 2;
    }
```

a) returns 1
b) returns 2
c) Varies
d) Compile time error
View Answer
Answer: d
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int k = 8;
        int m = 7;
        k < m ? k++ : m = k;
        printf("%d", k);
    }
```

a) 7
b) 8

c) Compile time error
d) Run time error
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int k = 8;
        int m = 7;
        k < m ? k = k + 1 : m = m + 1;
        printf("%d", k);
    }
```

a) Compile time error
b) 9
c) 8
d) Run time error
View Answer
Answer: a
Explanation: None.

3. What will be the final values of a and c in the following C statement? (Initial values: a = 2, c = 1)

```
 c = (c) ? a = 0 : 2;
```

a) a = 0, c = 0;
b) a = 2, c = 2;
c) a = 2, c = 2;
d) a = 1, c = 2;
View Answer
Answer: a
Explanation: None.

4. What will be the data type of the following expression? (Initial data type: a = int, var1 = double, var2 = float)

```
expression (a < 50)? var1 : var2;
```

a) int
b) float
c) double
d) Cannot be determined
View Answer
Answer: c
Explanation: None.

5. Which expression has to be present in the following?

exp1 ? exp2 : exp3;

a) exp1
b) exp2
c) exp3
d) all of the mentioned
View Answer
Answer: d
Explanation: None.

6. What will be the final value of c in the following C code snippet? (Initial values: a = 1, b = 2, c = 1)

```
        c += (-c) ? a : b;
```

a) Syntax Error
b) c = 1
c) c = 2
d) c = 3
View Answer
Answer: c
Explanation: None.

7. The following C code can be rewritten as _______

```
c = (n) ? a : b;
```

a)

```
    if (!n)c = b;
    else c = a;
```

b)

```
    if (n <;= 0)c = b;
    else c = a;
```

c)

```
    if (n > 0)c = a;
    else c = b;
```

d) All of the mentioned
View Answer
Answer: a
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        reverse(1);
    }
    void reverse(int i)
    {
        if (i > 5)
```

```
            exit(0);
        printf("%d\n", i);
        return reverse(i++);
    }
```

a) 1 2 3 4 5
b) 1 2 3 4
c) Compile time error
d) Stack overflow
View Answer
Answer: d
Explanation: None.

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

```
    #include <stdio.h>
    void reverse(int i);
    int main()
    {
        reverse(1);
    }
    void reverse(int i)
    {
        if (i > 5)
            return ;
        printf("%d ", i);
        return reverse((i++, i));
    }
```

a) 1 2 3 4 5
b) Segmentation fault
c) Compilation error
d) Undefined behaviour
View Answer
Answer: a
Explanation: None.

3. In expression i = g() + f(), first function called depends on __________
a) Compiler
b) Associativiy of () operator
c) Precedence of () and + operator
d) Left to write of the expression
View Answer
Answer: a
Explanation: None.

4. What will be the final values of i and j in the following C code?

```
    #include <stdio.h>
    int x = 0;
    int main()
    {
        int i = (f() + g()) || g();
        int j = g() || (f() + g());
    }
    int f()
    {
        if (x == 0)
            return x + 1;
        else
            return x - 1;
    }
    int g()
    {
        return x++;
    }
```

a) i value is 1 and j value is 1
b) i value is 0 and j value is 0
c) i value is 1 and j value is undefined
d) i and j value are undefined
View Answer
Answer: d
Explanation: None.

5. What will be the final values of i and j in the following C code?

```
    #include <stdio.h>
    int x = 0;
    int main()
    {
        int i = (f() + g()) | g(); //bitwise or
        int j = g() | (f() + g()); //bitwise or
    }
    int f()
    {
        if (x == 0)
            return x + 1;
        else
            return x - 1;
    }
    int g()
    {
        return x++;
    }
```

a) i value is 1 and j value is 1
b) i value is 0 and j value is 0
c) i value is 1 and j value is undefined
d) i and j value are undefined
View Answer
Answer: c
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 2, y = 0;
        int z = y && (y |= 10);
        printf("%d\n", z);
        return 0;
    }
```

a) 1
b) 0
c) Undefined behaviour due to order of evaluation
d) 2
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 2, y = 0;
        int z = (y++) ? 2 : y == 1 && x;
        printf("%d\n", z);
        return 0;
    }
```

a) 0
b) 1
c) 2
d) Undefined behaviour
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 2, y = 0;
        int z;
        z = (y++, y);
        printf("%d\n", z);
        return 0;
    }
```

a) 0
b) 1
c) Undefined behaviour
d) Compilation error
View Answer
Answer: b
Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int x = 2, y = 0, l;
        int z;
        z = y = 1, l = x && y;
        printf("%d\n", l);
        return 0;
    }
```

a) 0
b) 1
c) Undefined behaviour due to order of evaluation can be different
d) Compilation error
View Answer
Answer: b
Explanation: None.

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

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

a) 12
b) 20
c) 4
d) Either 12 or 20
View Answer
Answer: b
Explanation: None.

1. C99 standard guarantees uniqueness of __________ characters for internal names.

a) 31

b) 63

c) 12

d) 14

View Answer

Answer: b

Explanation: ISO C99 compiler may consider only first 63 characters for internal names.

2. C99 standard guarantees uniqueness of ___________ characters for external names.

a) 31

b) 6

c) 12

d) 14

View Answer

Answer: a

Explanation: ISO C99 compiler may consider only first 31 characters for external names.

3. Which of the following is not a valid variable name declaration?

a) int __a3;

b) int __3a;

c) int __A3;

d) None of the mentioned

View Answer

Answer: d

Explanation: None.

4. Which of the following is not a valid variable name declaration?

a) int _a3;

b) int a_3;

c) int 3_a;

d) int _3a

View Answer

Answer: c

Explanation: Variable name cannot start with a digit.

5. Why do variable names beginning with the underscore is not encouraged?

a) It is not standardized

b) To avoid conflicts since assemblers and loaders use such names

c) To avoid conflicts since library routines use such names

d) To avoid conflicts with environment variables of an operating system

View Answer

Answer: c

Explanation: None.

6. All keywords in C are in ____________

a) LowerCase letters

b) UpperCase letters

c) CamelCase letters

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

7. Variable name resolution (number of significant characters for the uniqueness of variable) depends on ___________

a) Compiler and linker implementations

b) Assemblers and loaders implementations

c) C language

d) None of the mentioned

View Answer

Answer: a

Explanation: It depends on the standard to which compiler and linkers are adhering.

8. Which of the following is not a valid C variable name?

a) int number;

b) float rate;

c) int variable_count;

d) int $main;

View Answer

Answer: d

Explanation: Since only underscore and no other special character is allowed in a variable name, it results in an error.

9. Which of the following is true for variable names in C?

a) They can contain alphanumeric characters as well as special characters

b) It is not an error to declare a variable to be one of the keywords(like goto, static)

c) Variable names cannot start with a digit

d) Variable can be of any length

View Answer

Answer: c

Explanation: According to the syntax for C variable name, it cannot start with a digit.

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

```
#include <stdio.h>
void main()
{
    int x = 5;
    if (x < 1)
        printf("hello");
    if (x == 5)
        printf("hi");
    else
        printf("no");
}
```

a) hi

b) hello

c) no

d) error

View Answer

Answer: a

Explanation: None.

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

```
#include <stdio.h>
int x;
void main()
{
    if (x)
        printf("hi");
    else
        printf("how are u");
}
```

a) hi

b) how are you

c) compile time error

d) error

View Answer

Answer: b

Explanation: None.

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

```
#include <stdio.h>
void main()
{
    int x = 5;
    if (true);
        printf("hello");
}
```

a) It will display hello

b) It will throw an error

c) Nothing will be displayed

d) Compiler dependent

View Answer

Answer: b

Explanation: None.

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

```
#include <stdio.h>
void main()
{
    int x = 0;
    if (x == 0)
        printf("hi");
    else
        printf("how are u");
        printf("hello");
}
```

a) hi

b) how are you

c) hello

d) hihello

View Answer

Answer: d

Explanation: None.

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

```
#include <stdio.h>
void main()
{
    int x = 5;
```

```
        if (x < 1);

            printf("Hello");


    }
```

a) Nothing

b) Run time error

c) Hello

d) Varies

View Answer

Answer: c

Explanation: None.

6. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
    #include <stdio.h>

    void main()

    {

        double ch;

        printf("enter a value between 1 to 2:");

        scanf("%lf", &ch);

        switch (ch)

        {

           case 1:

             printf("1");

             break;

           case 2:

             printf("2");

             break;

        }
    }
```

a) Compile time error

b) 1

c) 2

d) Varies

View Answer

Answer: a

Explanation: None.

7. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
    #include <stdio.h>
    void main()
    {
        char *ch;
        printf("enter a value between 1 to 3:");
        scanf("%s", ch);
        switch (ch)
        {
           case "1":
              printf("1");
              break;
           case "2":
              printf("2");
              break;
        }
    }
```

a) 1

b) 2

c) Compile time error

d) No Compile time error

View Answer

Answer: c

Explanation: None.

8. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
#include <stdio.h>
void main()
{
    int ch;
    printf("enter a value between 1 to 2:");
    scanf("%d", &ch);
    switch (ch)
    {
        case 1:
            printf("1\n");
        default:
            printf("2\n");
    }
}
```

a) 1

b) 2

c) 1 2

d) Run time error

View Answer

Answer: c

Explanation: None.

9. What will be the output of the following C code? (Assuming that we have entered the value 2 in the standard input)

```
#include <stdio.h>
void main()
{
    int ch;
    printf("enter a value between 1 to 2:");
    scanf("%d", &ch);
    switch (ch)
    {
       case 1:
          printf("1\n");
          break;
          printf("Hi");
       default:
          printf("2\n");
    }
}
```

a) 1

b) Hi 2

c) Run time error

d) 2

View Answer

Answer: d

Explanation: None.

10. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
    #include <stdio.h>
    void main()
    {
        int ch;
        printf("enter a value between 1 to 2:");
        scanf("%d", &ch);
        switch (ch, ch + 1)
        {
           case 1:
              printf("1\n");
              break;
           case 2:
              printf("2");
              break;
        }
    }
```

a) 1

b) 2

c) 3

d) Run time error

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        m();
        void m()
        {
            printf("hi");
        }
    }
```

a) hi

b) Compile time error

c) Nothing

d) Varies

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        m();
    }
    void m()
    {
        printf("hi");
        m();
    }
```

a) Compile time error

b) hi

c) Infinite hi

d) Nothing

View Answer

Answer: c

Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        static int x = 3;
        x++;
        if (x <= 5)
        {
            printf("hi");
            main();
        }
    }
```

a) Run time error

b) hi

c) Infinite hi

d) hi hi

View Answer

Answer: d

Explanation: None.

4. Which of the following is a correct format for declaration of function?

a) return-type function-name(argument type);

b) return-type function-name(argument type){}

c) return-type (argument type)function-name;

d) all of the mentioned

View Answer

Answer: a

Explanation: None.

5. Which of the following function declaration is illegal?

a) int 1bhk(int);

b) int 1bhk(int a);

c) int 2bhk(int*, int []);

d) all of the mentioned

View Answer

Answer: d

Explanation: None.

6. Which function definition will run correctly?

a)

```
   int sum(int a, int b)
   return (a + b);
```

b)

```
   int sum(int a, int b)
   {return (a + b);}
```

c)

```
   int sum(a, b)
   return (a + b);
```

d) none of the mentioned

View Answer

Answer: b

Explanation: None.

7. Can we use a function as a parameter of another function? [Eg: void wow(int func())].

a) Yes, and we can use the function value conveniently

b) Yes, but we call the function again to get the value, not as convenient as in using variable

c) No, C does not support it

d) This case is compiler dependent

View Answer

Answer: c

Explanation: None.

8. The value obtained in the function is given back to main by using ________ keyword.

a) return

b) static

c) new

d) volatile

View Answer

Answer: a

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

```
#include <stdio.h>

int *m();

void main()

{

    int k = m();

    printf("%d", k);

}

int *m()

{

    int a[2] = {5, 8};
```

```
        return a;
    }
```

a) 5

b) 8

c) Nothing

d) Varies

View Answer

Answer: d

Explanation: None.

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

```
    #include <stdio.h>
    void m(int k)
    {
        printf("hi");
    }
    void m(double k)
    {
        printf("hello");
    }
    void main()
    {
        m(3);
    }
```

a) hi

b) hello

c) Compile time error

d) Nothing

View Answer

Answer: c

Explanation: None.

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

a) void

b) int

c) double

d) short int

View Answer

Answer: b

Explanation: None.

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

```
#include <stdio.h>
int foo();
int main()
{
    int i = foo();
}
foo()
{
    printf("2 ");
    return 2;
}
```

a) 2

b) Compile time error

c) Depends on the compiler

d) Depends on the standard

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    double foo();
    int main()
    {
        foo();
        return 0;
    }
    foo()
    {
        printf("2 ");
        return 2;
    }
```

a) 2

b) Compile time error

c) Depends on the compiler

d) Depends on the standard

View Answer

Answer: b

Explanation: None.

6. Functions can return structure in C?

a) True

b) False

c) Depends on the compiler

d) Depends on the standard

View Answer

Answer: a

Explanation: None.

7. Functions can return enumeration constants in C?

a) true

b) false

c) depends on the compiler

d) depends on the standard

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    enum m{JAN, FEB, MAR};
    enum m foo();
    int main()
    {
        enum m i = foo();
        printf("%d\n", i);
    }
    int  foo()
    {
        return JAN;
    }
```

a) Compile time error

b) 0

c) Depends on the compiler

d) Depends on the standard

View Answer

Answer: a

Explanation: None.

1. If the file name is enclosed in double quotation marks, then _________

a) The preprocessor treats it as a user-defined file

b) The preprocessor treats it as a system-defined file

c) The preprocessor treats it as a user-defined file & system-defined file

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

2. If the file name is enclosed in angle brackets, then ___________

a) The preprocessor treats it as a user-defined file

b) The preprocessor treats it as a system-defined file

c) The preprocessor treats it as a user-defined file & system-defined file

d) None of the mentioned

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code snippet?

```
  #include (stdio.h)
  void main()
  {
      printf("hello");
  }
```

a) hello

b) Nothing

c) Compile time error

d) Depends on compiler

View Answer

Answer: c

Explanation: File to be included must be specified either in “” or <>.

Output:

```
$ cc pgm1.c
pgm1.c:1: error: #include expects “FILENAME” or
pgm1.c: In function ‘main’:
pgm1.c:4: warning: incompatible implicit declaration of built-in function ‘printf’
```

4. The below two lines are equivalent to ________

```
    #define C_IO_HEADER <stdio.h>
    #include C_IO_HEADER
```

a) #include<stdlib.h>

b) #include”printf”

c) #include”C_IO_HEADER”

d) #include<stdio.h>

View Answer

Answer: d

Explanation: Since C_IO_HEADER is defined to be <stdio.h>, the second line becomes #include<stdio.h>, since C_IO_HEADER is replaced with <stdio.h>

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

```
    #include <stdio.h>
    #include "printf"
    void main()
    {
        printf("hello");
    }
```

a) hello

b) Error

c) Depends on compiler

d) Varies

View Answer

Answer: a

Explanation: None.

6. Which of the following file extensions are accepted with #include?

a) .h

b) .in

c) .com

d) All of the mentioned

View Answer

Answer: d

Explanation: The preprocessor will include whatever file extension you specify in your #include statement. However, it is not a good practice as another person debugging it will find it difficult in finding files you have included.

7. Which of the following names for files not accepted?

a) header.h.h

b) 123header.h

c) _head_er.h

d) None of the mentioned

View Answer

Answer: d

Explanation: All file names are accepted as for the execution to occur. There are no constraints on giving file names for inclusion.

1. What is the return-type of the function sqrt()?

a) int

b) float

c) double

d) depends on the data type of the parameter

View Answer

Answer: c

Explanation: None.

2. Which of the following function declaration is illegal?

a)

```
double func();
int main(){}
double func(){}
```

b)

```
double func(){};
int main(){}
```

c)

```
int main()
{
    double func();
}
double func(){//statements}
```

d) None of the mentioned

View Answer

Answer: d

Explanation: None.

3. What will be the output of the following C code having void return-type function?

```
#include <stdio.h>
void foo()
{
    return 1;
}
void main()
{
    int x = 0;
    x = foo();
    printf("%d", x);
}
```

a) 1

b) 0

c) Runtime error

d) Compile time error

View Answer

Answer: d

Explanation: None.

4. What will be the data type returned for the following C function?

```
#include <stdio.h>
int func()
{
    return (double)(char)5.0;
    }
```

a) char

b) int

c) double

d) multiple type-casting in return is illegal

View Answer

Answer: b

Explanation: None.

5. What is the problem in the following C declarations?

```
   int func(int);
   double func(int);
   int func(float);
```

a) A function with same name cannot have different signatures

b) A function with same name cannot have different return types

c) A function with same name cannot have different number of parameters

d) All of the mentioned

View Answer

Answer: d

Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int k = m();
        printf("%d", k);
    }
    void m()
    {
        printf("hello");
    }
```

a) hello 5

b) Error

c) Nothing

d) Junk value

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int *m()
    {
        int *p = 5;
        return p;
    }
    void main()
    {
        int *k = m();
        printf("%d", k);
    }
```

a) 5

b) Junk value

c) 0

d) Error

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int *m();
    void main()
    {
        int *k = m();
        printf("hello ");
        printf("%d", k[0]);
    }
    int *m()
    {
        int a[2] = {5, 8};
        return a;
    }
```

a) hello 5 8

b) hello 5

c) hello followed by garbage value

d) Compilation error

View Answer

Answer: c

Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        m();
        printf("%d", x);
    }
    int x;
    void m()
    {
        x = 4;
    }
```

a) 4

b) Compile time error

c) 0

d) Undefined

View Answer

Answer: b

Explanation: None.

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

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

a) Junk value

b) Run time error

c) 0

d) Undefined

View Answer

Answer: c

Explanation: None.

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

```
#include <stdio.h>
int x = 5;
void main()
{
    int x = 3;
    printf("%d", x);
    {
        x = 4;
    }
    printf("%d", x);
}
```

a) Run time error

b) 3 3

c) 3 5

d) 3 4

View Answer

Answer: d

Explanation: None.

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

```
#include <stdio.h>
int x = 5;
void main()
{
        int x = 3;
        printf("%d", x);
        {
            int x = 4;
        }
        printf("%d", x);
    }
```

a) 3 3

b) 3 4

c) 3 5

d) Run time error

View Answer

Answer: a

Explanation: None.

5. Functions in C are always _________

a) Internal

b) External

c) Both Internal and External

d) External and Internal are not valid terms for functions

View Answer

Answer: b

Explanation: None.

6. Global variables are ____________

a) Internal

b) External

c) Both Internal and External

d) None of the mentioned

View Answer

Answer: b

Explanation: None.

7. Which of the following is an external variable in the following C code?

```
#include <stdio.h>
int func (int a)
{
    int b;
    return b;
}
int main()
{
    int c;
    func (c);
}
int d;
```

a) a

b) b

c) c

d) d

View Answer

Answer: d

Explanation: None.

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

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

a) 9

b) 10

c) 11

d) Compile time error

View Answer

Answer: d

Explanation: None.

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

```
    #include <stdio.h>
    double var = 8;
    int main()
    {
        int var = 5;
        printf("%d", var);
    }
```

a) 5

b) 8

c) Compile time error due to wrong format identifier for double

d) Compile time error due to redeclaration of variable with same name

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    double i;
    int main()
    {
       printf("%g\n",i);
       return 0;
    }
```
a) 0

b) 0.000000

c) Garbage value

d) Depends on the compiler

View Answer

Answer: a

Explanation: None.

2. Which part of the program address space is p stored in the following C code?

```
    #include <stdio.h>
    int *p = NULL;
    int main()
    {
        int i = 0;
        p = &i;
        return 0;
    }
```
a) Code/text segment

b) Data segment

c) Bss segment

d) Stack

View Answer

Answer: b

Explanation: None.

3. Which part of the program address space is p stored in the following C code?

```
    #include <stdio.h>
    int *p;
    int main()
    {
        int i = 0;
        p = &i;
        return 0;
    }
```

a) Code/text segment

b) Data segment

c) Bss segment

d) Stack

View Answer

Answer: c

Explanation: None.

4. Can variable i be accessed by functions in another source file?

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

a) Yes

b) No

c) Only if static keyword is used

d) Depends on the type of the variable

View Answer

Answer: a

Explanation: None.

5. Property of the external variable to be accessed by any source file is called by the C90 standard as __________

a) external linkage

b) external scope

c) global scope

d) global linkage

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int *i;
    int main()
    {
        if (i == NULL)
            printf("true\n");
        return 0;
    }
```

a) true

b) true only if NULL value is 0

c) Compile time error

d) Nothing

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int *i;
    int main()
    {
        if (i == 0)
            printf("true\n");
        return 0;
    }
```

a) true

b) true only if NULL value is 0

c) Compile time error

d) Nothing

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    static int x = 5;
    void main()
    {
        x = 9;
        {
            int x = 4;
        }
        printf("%d", x);
    }
```

a) 9

b) 4

c) 5

d) 0

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int i;
    int main()
    {
        extern int i;
        if (i == 0)
            printf("scope rules\n");
    }
```

a) scope rules

b) Compile time error due to multiple declaration

c) Compile time error due to not defining type in statement extern i

d) Nothing will be printed as value of i is not zero because i is an automatic variable

View Answer

Answer: a

Explanation: None.

2. What will be the output of the following C code (without linking the source file in which ary1 is defined)?

```
    #include <stdio.h>
    int main()
    {
        extern ary1[];
        printf("scope rules\n");
    }
```

a) scope rules

b) Linking error due to undefined reference

c) Compile time error because size of array is not provided

d) Compile time error because datatype of array is not provided

View Answer

Answer: a

Explanation: None.

3. What will be the output of the following C code (after linking to source file having definition of ary1)?

```
    #include <stdio.h>
    int main()
    {
        extern ary1[];
        printf("%d\n", ary1[0]);
    }
```

a) Value of ary1[0];

b) Compile time error due to multiple definition

c) Compile time error because size of array is not provided

d) Compile time error because datatype of array is not provided

View Answer

Answer: d

Explanation: None.

4. What is the scope of an external variable?

a) Whole source file in which it is defined

b) From the point of declaration to the end of the file in which it is defined

c) Any source file in a program

d) From the point of declaration to the end of the file being compiled

View Answer

Answer: d

Explanation: None.

5. What is the scope of a function?

a) Whole source file in which it is defined

b) From the point of declaration to the end of the file in which it is defined

c) Any source file in a program

d) From the point of declaration to the end of the file being compiled

View Answer

Answer: d

Explanation: None.

6. Comment on the output of the following C code.

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 0;i < 5; i++)
    int a = i;
```

```
        printf("%d", a);
    }
```

a) a is out of scope when printf is called

b) Redeclaration of a in same scope throws error

c) Syntax error in declaration of a

d) No errors, program will show the output 5

View Answer

Answer: c

Explanation: None.

7. Which variable has the longest scope in the following C code?

```
    #include <stdio.h>
    int b;
    int main()
    {
        int c;
        return 0;
    }
    int a;
```

a) a

b) b

c) c

d) Both a and b

View Answer

Answer: b

Explanation: None.

8. Comment on the following 2 C programs.

```
#include <stdio.h> //Program 1
int main()
{
    int a;
    int b;
    int c;
}

#include <stdio.h> //Program 2
int main()
{
    int a;
    {
        int b;
    }
    {
        int c;
    }
}
```

a) Both are same

b) Scope of c is till the end of the main function in Program 2

c) In Program 1, variables a, b and c can be used anywhere in the main function whereas in Program 2, variables b and c can be used only inside their respective blocks

d) None of the mentioned

View Answer

Answer: c

Explanation: None.

1. What will be the sequence of allocation and deletion of variables in the following C code?

```
#include <stdio.h>
int main()
{
    int a;
    {
        int b;
    }
}
```

a) a->b, a->b

b) a->b, b->a

c) b->a, a->b

d) b->a, b->a

View Answer

Answer: b

Explanation: None.

2. Array sizes are optional during array declaration by using ______ keyword.

a) auto

b) static

c) extern

d) register

View Answer

Answer: c

Explanation: None.

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

```
#include <stdio.h>
void main()
{
        int x = 3;
        {
            x = 4;
            printf("%d", x);
        }
    }
```

a) 4

b) 3

c) 0

d) Undefined

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int x = 5;
    void main()
    {
        int x = 3;
        m();
        printf("%d", x);
    }
    void m()
    {
        x = 8;
        n();
    }
    void n()
    {
        printf("%d", x);
    }
```

a) 8 3

b) 3 8

c) 8 5

d) 5 3

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int x;
    void main()
    {
        m();
        printf("%d", x);
    }
    void m()
    {
        x = 4;
    }
```

a) 0

b) 4

c) Compile time error

d) Undefined

View Answer

Answer: b

Explanation: None.

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

```
#include <stdio.h>
static int x = 5;
void main()
{
    int x = 9;
    {
        x = 4;
    }
    printf("%d", x);
}
```

a) 9

b) 5

c) 4

d) 0

View Answer

Answer: c

Explanation: None.

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

```
#include <stdio.h>
void main()
{
    {
        int x = 8;
    }
```

```
        printf("%d", x);
    }
```

a) 8

b) 0

c) Undefined

d) Compile time error

View Answer

Answer: d

Explanation: None.

1. What will be the sequence of allocation and deletion of variables in the following C code?

```
    #include <stdio.h>
    int main()
    {
        int a;
        {
            int b;
        }
    }
```

a) a->b, a->b

b) a->b, b->a

c) b->a, a->b

d) b->a, b->a

View Answer

Answer: b

Explanation: None.

2. Array sizes are optional during array declaration by using ______ keyword.

a) auto

b) static

c) extern

d) register

View Answer

Answer: c

Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        int x = 3;
        {
            x = 4;
            printf("%d", x);
        }
    }
```

a) 4

b) 3

c) 0

d) Undefined

View Answer

Answer: a

Explanation: None.

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

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

```
int x = 5;
void main()
{
    int x = 3;
    m();
    printf("%d", x);
}
void m()
{
    x = 8;
    n();
}
void n()
{
    printf("%d", x);
}
```

a) 8 3

b) 3 8

c) 8 5

d) 5 3

View Answer

Answer: a

Explanation: None.

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

```
#include <stdio.h>
int x;
void main()
{
    m();
    printf("%d", x);
}
void m()
{
    x = 4;
}
```

a) 0

b) 4

c) Compile time error

d) Undefined

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    static int x = 5;
    void main()
    {
        int x = 9;
        {
            x = 4;
        }
        printf("%d", x);
    }
```

a) 9

b) 5

c) 4

d) 0

View Answer

Answer: c

Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        {
            int x = 8;
        }
        printf("%d", x);
    }
```

a) 8

b) 0

c) Undefined

d) Compile time error

View Answer

Answer: d

Explanation: None.

1. What will be the output of the following C code if these two files namely test.c and test1.c are linked and run?

```
      -------file test.c-------
    #include <stdio.h>
    #include ""test.h""
    int main()
    {
        i = 10;
        printf(""%d "", i);
        foo();
    }

    -----file test1.c------
    #include <stdio.h>
    #include ""test.h""
    int foo()
    {
        printf(""%d\n"", i);
    }

    -----file test.h-----
    #include <stdio.h>
    #include <stdlib.h>
    static int i;
```

a) 10 0

b) 0 0

c) 10 10

d) Compilation Error

View Answer

Answer: a

Explanation: None.

2. Functions have static qualifier for its declaration by default.

a) True

b) False

c) Depends on the compiler

d) Depends on the standard

View Answer

Answer: b

Explanation: None.

3. Is initialisation mandatory for local static variables?

a) Yes

b) No

c) Depends on the compiler

d) Depends on the standard

View Answer

Answer: b

Explanation: None.

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

```
#include <stdio.h>
int main()
{
    foo();
    foo();
}
void foo()
{
    int i = 11;
    printf("%d ", i);
    static int j = 12;
    j = j + 1;
    printf("%d\n", j);
}
```

a) 11 12 11 12

b) 11 13 11 14

c) 11 12 11 13

d) Compile time error

View Answer

Answer: b

Explanation: None.

5. Assignment statements assigning value to local static variables are executed only once.

a) True

b) False

c) Depends on the code

d) None of the mentioned

View Answer

Answer: b

Explanation: None.

6. What is the format identifier for “static a = 20.5;”?

a) %s

b) %d

c) %f

d) Illegal declaration due to absence of data type

View Answer

Answer: b

Explanation: None.

7. Which of the following is true for the static variable?

a) It can be called from another function

b) It exists even after the function ends

c) It can be modified in another function by sending it as a parameter

d) All of the mentioned

View Answer

Answer: b

Explanation: None.

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

```
#include <stdio.h>
void func();
int main()
{
    static int b = 20;
    func();
}
void func()
{
    static int b;
    printf("%d", b);
}
```

a) Output will be 0

b) Output will be 20

c) Output will be a garbage value

d) Compile time error due to redeclaration of static variable

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        register int i = 10;
        int *p = &i;
        *p = 11;
        printf("%d %d\n", i, *p);
    }
```

a) Depends on whether i is actually stored in machine register

b) 10 10

c) 11 11

d) Compile time error

View Answer

Answer: d

Explanation: None.

2. register keyword mandates compiler to place it in machine register.

a) True

b) False

c) Depends on the standard

d) None of the mentioned

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        register static int i = 10;
        i = 11;
        printf("%d\n", i);
    }
```

a) 10

b) Compile time error

c) Undefined behaviour

d) 11

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        register auto int i = 10;
        i = 11;
        printf("%d\n", i);
    }
```

a) 10

b) Compile time error

c) Undefined behaviour

d) 11

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        register const int i = 10;
        i = 11;
        printf("%d\n", i);
    }
```

a) 10

b) Compile time error

c) Undefined behaviour

d) 11

View Answer

Answer: b

Explanation: None.

6. Register storage class can be specified to global variables.

a) True

b) False

c) Depends on the compiler

d) Depends on the standard

View Answer

Answer: b

Explanation: None.

7. Which among the following is wrong for “register int a;”?

a) Compiler generally ignores the request

b) You cannot take the address of this variable

c) Access time to a is critical

d) None of the mentioned

View Answer

Answer: d

Explanation: None.

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

```
#include <stdio.h>
void main()
{
    register int x = 5;
    m();
    printf("x is %d", x);
}
void m()
{
    x++;
}
```

a) 6

b) 5

c) Junk value

d) Compile time error

View Answer

Answer: d

Explanation: None.

1. When compiler accepts the request to use the variable as a register?

a) It is stored in CPU

b) It is stored in cache memory

c) It is stored in main memory

d) It is stored in secondary memory

View Answer

Answer: a

Explanation: None.

2. Which data type can be stored in register?

a) int

b) long

c) float

d) all of the mentioned

View Answer

Answer: d

Explanation: None.

3. Which of the following operation is not possible in a register variable?

a) Reading the value into a register variable

b) Copy the value from a memory variable

c) Global declaration of register variable

d) All of the mentioned

View Answer

Answer: d

Explanation: None.

4. Which among the following is the correct syntax to declare a static variable register?

a) static register a;

b) register static a;

c) Both static register a; and register static a;

d) We cannot use static and register together

View Answer

Answer: d

Explanation: None.

5. Register variables reside in ________

a) stack

b) registers

c) heap

d) main memory

View Answer

Answer: b

Explanation: None.

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

```
#include <stdio.h>
void main()
{
    register int x = 0;
    if (x < 2)
    {
        x++;
        main();
    }
}
```

a) Segmentation fault

b) main is called twice

c) main is called once

d) main is called thrice

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        register int x;
        printf("%d", x);
    }
```

a) 0

b) Junk value

c) Compile time error

d) Nothing

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    register int x;
    void main()
    {
        printf("%d", x);
    }
```

a) Varies

b) 0

c) Junk value

d) Compile time error

View Answer

Answer: d

Explanation: None.

1. What is the scope of an automatic variable?

a) Within the block it appears

b) Within the blocks of the block it appears

c) Until the end of program

d) Within the block it appears & Within the blocks of the block it appears

View Answer

Answer: d

Explanation: None.

2. Automatic variables are allocated space in the form of a __________

a) stack

b) queue

c) priority queue

d) random

View Answer

Answer: a

Explanation: None.

3. Which of the following is a storage specifier?

a) enum

b) union

c) auto

d) volatile

View Answer

Answer: c

Explanation: None.

4. If storage class is not specified for a local variable, then the default class will be auto.

a) True

b) False

c) Depends on the standard

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

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

```
#include <stdio.h>
void foo(auto int i);
int main()
{
    foo(10);
}
void foo(auto int i)
{
    printf("%d\n", i );
}
```

a) 10

b) Compile time error

c) Depends on the standard

d) None of the mentioned

View Answer

Answer: b

Explanation: None.

6. Automatic variables are stored in ________

a) stack

b) data segment

c) register

d) heap

View Answer

Answer: a

Explanation: None.

7. What linkage does automatic variables have?

a) Internal linkage

b) External linkage

c) No linkage

d) None of the mentioned

View Answer

Answer: c

Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        auto i = 10;
        const auto int *p = &i;
        printf("%d\n", i);
    }
```

a) 10

b) Compile time error

c) Depends on the standard

d) Depends on the compiler

View Answer

Answer: a

1. Automatic variables are _________

a) Declared within the scope of a block, usually a function

b) Declared outside all functions

c) Declared with the auto keyword

d) Declared within the keyword extern

View Answer

Answer: a

Explanation: None.

2. What is the scope of an automatic variable?

a) Exist only within that scope in which it is declared

b) Cease to exist after the block is exited

c) Exist only within that scope in which it is declared & exist after the block is exited

d) All of the mentioned

View Answer

Answer: c

Explanation: None.

3. Automatic variables are allocated memory in ___________

a) heap

b) Data segment

c) Code segment

d) stack

View Answer

Answer: d

Explanation: None.

4. What will be the x in the following C code?

```
#include <stdio.h>
void main()
{
    int x;
}
```

a) automatic variable

b) static variable

c) register variable

d) global variable

View Answer

Answer: a

Explanation: None.

5. Automatic variables are initialized to ___________

a) Zero

b) Junk value

c) Nothing

d) Both Zero & Junk value

View Answer

Answer: b

Explanation: None.

6. Which of the following storage class supports char data type?

a) register

b) static

c) auto

d) all of the mentioned

View Answer

Answer: d

Explanation: None.

7. A local variable declaration with no storage class specified is by default _________

a) auto

b) extern

c) static

d) register

View Answer

Answer: a

Explanation: None.

1. Property which allows to produce different executable for different platforms in C is called?

a) File inclusion

b) Selective inclusion

c) Conditional compilation

d) Recursive macros

View Answer

Answer: c

Explanation: Conditional compilation is the preprocessor facility to produce a different executable.

2. What is #include <stdio.h>?

a) Preprocessor directive

b) Inclusion directive

c) File inclusion directive

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

3. C preprocessors can have compiler specific features.

a) True

b) False

c) Depends on the standard

d) Depends on the platform

View Answer

Answer: a

Explanation: #pragma is compiler specific feature.

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

```
#include <stdio.h>
#define foo(m, n) m * n = 10
int main()
{
    printf("in main\n");
}
```

a) In main

b) Compilation error as lvalue is required for the expression m*n=10

c) Preprocessor error as lvalue is required for the expression m*n=10

d) None of the mentioned

View Answer

Answer: a

Explanation: Preprocessor just replaces whatever is given compiler then checks for error at the replaced part of the code. Here it is not replaced anywhere.

Output:

```
$ cc pgm1.c
$ a.out
in main
```

5. C preprocessor is conceptually the first step during compilation.

a) True

b) False

c) Depends on the compiler

d) Depends on the standard

View Answer

Answer: a

Explanation: None.

6. Preprocessor feature that supply line numbers and filenames to compiler is called?

a) Selective inclusion

b) macro substitution

c) Concatenation

d) Line control

View Answer

Answer: d

Explanation: None.

7. #include <somefile.h> are ________ files and #include “somefile.h” _________ files.

a) Library, Library

b) Library, user-created header

c) User-created header, library

d) They can include all types of file

View Answer

Answer: d

Explanation: Both of these statement can be used to select any file.

8. What is a preprocessor?

a) That processes its input data to produce output that is used as input to another program

b) That is nothing but a loader

c) That links various source files

d) All of the mentioned

View Answer

Answer: a

Explanation: A preprocessor is a program that processes its input data to produce output that is used as input to another program.

1. Which of the following are C preprocessors?

a) #ifdef

b) #define

c) #endif

d) all of the mentioned

View Answer

Answer: d

Explanation: None.

2. #include statement must be written __________

a) Before main()

b) Before any scanf/printf

c) After main()

d) It can be written anywhere

View Answer

Answer: a

Explanation: Using these directives before main() improves readability.

3. #pragma exit is primarily used for?

a) Checking memory leaks after exiting the program

b) Informing Operating System that program has terminated

c) Running a function at exiting the program

d) No such preprocessor exist

View Answer

Answer: c

Explanation: It is primarily used for running a function upon exiting the program.

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

```
#include <stdio.h>
int main()
{
    int one = 1, two = 2;
    #ifdef next
    one = 2;
    two = 1;
    #endif
    printf("%d, %d", one, two);
}
```

a) 1, 1

b) 1, 2

c) 2, 1

d) 2, 2

View Answer

Answer: b

Explanation: None.

5. The C-preprocessors are specified with _________symbol.

a) #

b) $

c) ” ”

d) &

View Answer

Answer: a

Explanation: The C-preprocessors are specified with # symbol.

6. What is #include directive?

a) Tells the preprocessor to grab the text of a file and place it directly into the current file

b) Statements are not typically placed at the top of a program

c) All of the mentioned

d) None of the mentioned

View Answer

Answer: a

Explanation: The #include directive tells the preprocessor to grab the text of a file and place it directly into the current file and are statements are typically placed at the top of a program.

7. The preprocessor provides the ability for _______________

a) The inclusion of header files

b) The inclusion of macro expansions

c) Conditional compilation and line control

d) All of the mentioned

View Answer

Answer: d

Explanation: The preprocessor provides the ability for the inclusion of header files, macro expansions, conditional compilation, and line control.

8. If #include is used with file name in angular brackets.

a) The file is searched for in the standard compiler include paths

b) The search path is expanded to include the current source directory

c) The search path will expand

d) None of the mentioned

View Answer

Answer: a

Explanation: With the #include, if the filename is enclosed within angle brackets, the file is searched for in the standard compiler include paths.

1. What is the sequence for preprocessor to look for the file within <>?

a) The predefined location then the current directory

b) The current directory then the predefined location

c) The predefined location only

d) The current directory location

View Answer

Answer: a

Explanation: <> first searches the predefined location for the specified file and then the current directory.

2. Which directory the compiler first looks for the file when using #include?

a) Current directory where program is saved

b) C:COMPILERINCLUDE

c) S:SOURCEHEADERS

d) Both C:COMPILERINCLUDE and S:SOURCEHEADERS simultaneously

View Answer

Answer: b

Explanation: None.

3. What would happen if you create a file stdio.h and use #include “stdio.h”?

a) The predefined library file will be selected

b) The user-defined library file will be selected

c) Both the files will be included

d) The compiler won’t accept the program

View Answer

Answer: b

Explanation: None.

4. How is search done in #include and #include “somelibrary.h” according to C standard?

a) When former is used, current directory is searched and when latter is used, standard directory is searched

b) When former is used, standard directory is searched and when latter is used, current directory is searched

c) When former is used, search is done in implementation defined manner and when latter is used, current directory is searched

d) For both, search for ‘somelibrary’ is done in implementation-defined places

View Answer

Answer: d

Explanation: None.

5. How is search done in #include and #include”somelibrary.h” normally or conventionally?

a) When former is used, current directory is searched and when latter is used, standard directory is searched

b) When former is used, predefined directory is searched and when latter is used, current directory is searched and then predefined directories are searched

c) When former is used, search is done in implementation defined manner and latter is used to search current directory

d) For both, search for somelibrary is done in implementation-defined manner

View Answer

Answer: b

Explanation: None.

6. Can function definition be present in header files?

a) Yes

b) No

c) Depends on the compiler

d) Depends on the standard

View Answer

Answer: a

Explanation: None.

7. Comment on the output of the following C code.

```
    #include <stdio.h>
    #include "test.h"
    #include "test.h"
    int main()
    {
        //some code
    }
```

a) True

b) Compile time error

c) False

d) Depends on the compiler

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    #define foo(m, n) m ## n
    void myfunc();
    int main()
    {
        myfunc();
    }
    void myfunc()
    {
        printf("%d\n", foo(2, 3));
    }
```

a) 23

b) 2 3

c) Compile time error

d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

1. If the file name is enclosed in double quotation marks, then _________

a) The preprocessor treats it as a user-defined file

b) The preprocessor treats it as a system-defined file

c) The preprocessor treats it as a user-defined file & system-defined file

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

2. If the file name is enclosed in angle brackets, then ___________

a) The preprocessor treats it as a user-defined file

b) The preprocessor treats it as a system-defined file

c) The preprocessor treats it as a user-defined file & system-defined file

d) None of the mentioned

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code snippet?

```
    #include (stdio.h)
    void main()
    {
        printf("hello");
    }
```

a) hello

b) Nothing

c) Compile time error

d) Depends on compiler

View Answer

Answer: c

Explanation: File to be included must be specified either in “” or <>.

Output:

```
$ cc pgm1.c
pgm1.c:1: error: #include expects “FILENAME” or
pgm1.c: In function ‘main’:
pgm1.c:4: warning: incompatible implicit declaration of built-in function ‘printf’
```

4. The below two lines are equivalent to ________

```
    #define C_IO_HEADER <stdio.h>
    #include C_IO_HEADER
```

a) #include<stdlib.h>

b) #include”printf”

c) #include”C_IO_HEADER”

d) #include<stdio.h>

View Answer

Answer: d

Explanation: Since C_IO_HEADER is defined to be <stdio.h>, the second line becomes #include<stdio.h>, since C_IO_HEADER is replaced with <stdio.h>

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

```
#include <stdio.h>

#include "printf"

void main()

{

    printf("hello");

}
```

a) hello

b) Error

c) Depends on compiler

d) Varies

View Answer

Answer: a

Explanation: None.

6. Which of the following file extensions are accepted with #include?

a) .h

b) .in

c) .com

d) All of the mentioned

View Answer

Answer: d

Explanation: The preprocessor will include whatever file extension you specify in your #include statement. However, it is not a good practice as another person debugging it will find it difficult in finding files you have included.

7. Which of the following names for files not accepted?

a) header.h.h

b) 123header.h

c) _head_er.h

d) None of the mentioned

View Answer

Answer: d

Explanation: All file names are accepted as for the execution to occur. There are no constraints on giving file names for inclusion.

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

```
    #include <stdio.h>
    #define foo(m, n) m ## n
    int main()
    {
        printf("%s\n", foo(k, l));
    }
```

a) k l

b) kl

c) Compile time error

d) Undefined behaviour

View Answer

Answer: c

Explanation: None.

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

```
    #include <stdio.h>
    #define foo(m, n) " m ## n "
    int main()
    {
        printf("%s\n", foo(k, l));
    }
```

a) k l

b) kl

c) Compile time error

d) m ## n

View Answer

Answer: d

Explanation: None.

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

```
    #include <stdio.h>
    #define foo(x, y) #x #y
    int main()
    {
        printf("%s\n", foo(k, l));
        return 0;
    }
```

a) kl

b) k l

c) xy

d) Compile time error

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    #define foo(x, y) x / y + x
    int main()
    {
        int i = -6, j = 3;
        printf("%d\n",foo(i + j, 3));
        return 0;
    }
```

a) Divided by zero exception

b) Compile time error

c) -8

d) -4

View Answer

Answer: c

Explanation: None.

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

```
    #include <stdio.h>
    void f();
    int main()
    {
        #define foo(x, y) x / y + x
        f();
    }
    void f()
    {
        printf("%d\n", foo(-3, 3));
    }
```

a) -8

b) -4

c) Compile time error

d) Undefined behaviour

View Answer

Answer: b

Explanation: None.

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

```
#include <stdio.h>
void f();
int main()
{
    #define max 10
    f();
    return 0;
}
void f()
{
    printf("%d\n", max * 10);
}
```

a) 100

b) Compile time error since #define cannot be inside functions

c) Compile time error since max is not visible in f()

d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    #define foo(x, y) x / y + x
    int main()
    {
        int i = -6, j = 3;
        printf("%d ", foo(i + j, 3));
        printf("%d\n", foo(-3, 3));
        return 0;
    }
```

a) -8 -4

b) -4 divided by zero exception

c) -4 -4

d) Divided by zero exception

View Answer

Answer: a

Explanation: None.

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

```
 #include <stdio.h>
    int foo(int, int);
    #define foo(x, y) x / y + x
    int main()
    {
        int i = -6, j = 3;
        printf("%d ",foo(i + j, 3));
        #undef foo
        printf("%d\n",foo(i + j, 3));
    }
    int foo(int x, int y)
    {
        return x / y + x;
    }
```

a) -8 -4

b) Compile time error

c) -8 -8

d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

9. What is the advantage of #define over const?

a) Data type is flexible

b) Can have a pointer

c) Reduction in the size of the program

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        #define max 37;
        printf("%d", max);
    }
```

a) 37

b) Compile time error

c) Varies

d) Depends on compiler

View Answer

Answer: b

Explanation: None.

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

```
#include <stdio.h>
void main()
{
    #define max 37
    printf("%d", max);
}
```

a) 37

b) Run time error

c) Varies

d) Depends on compiler

View Answer

Answer: a

Explanation: None.

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

```
#include <stdio.h>
void main()
{
    #define const int
    const max = 32;
    printf("%d", max);
    }
```

a) Run time error

b) 32

c) int

d) const

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    void main()
    {
        #define max 45
        max = 32;
        printf("%d", max);
    }
```

a) 32

b) 45

c) Compile time error

d) Varies

View Answer

Answer: c

Explanation: None.

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

```
    #include <stdio.h>
    # define max
```

```
void m()
{
    printf("hi");
}
void main()
{
    max;
    m();
}
```

a) Run time error

b) hi hi

c) Nothing

d) hi

View Answer

Answer: d

Explanation: None.

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

```
#include <stdio.h>
#define A 1 + 2
#define B 3 + 4
int main()
{
    int var = A * B;
    printf("%d\n", var);
}
```

a) 9

b) 11

c) 12

d) 21

View Answer

Answer: b

Explanation: None.

7. Which of the following Macro substitution are accepted in C?

a)

```
#define A #define
A VAR 20
```

b)

```
#define A define
#A VAR 20
```

c)

```
#define #A #define
#A VAR 20
```

d) None of the mentioned

View Answer

8. Comment on the output of the following C code.

```
#include <stdio.h>
#define var 20);
int main()
{
    printf("%d\n", var
}
```

a) No errors, it will show the output 20

b) Compile time error, the printf braces aren’t closed

c) Compile time error, there are no open braces in #define

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

9. Which of the following properties of #define is not true?

a) You can use a pointer to #define

b) #define can be made externally available

c) They obey scope rules

d) All of the mentioned

View Answer

Answer: d

Explanation: None.

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

```
#include <stdio.h>
#define SYSTEM 20
int main()
{
    int a = 20;
    #if SYSTEM == a
    printf("HELLO ");
    #endif
    #if SYSTEM == 20
    printf("WORLD\n");
    #endif
}
```

a) HELLO

b) WORLD

c) HELLO WORLD

d) No Output

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    #define Cprog
    int main()
    {
        int a = 2;
        #ifdef Cprog
        a = 1;
        printf("%d", Cprog);
    }
```

a) No output on execution

b) Output as 1

c) Output as 2

d) Compile time error

View Answer

Answer: d

Explanation: None.

3. The “else if” in conditional inclusion is written by?

a) #else if

b) #elseif

c) #elsif

d) #elif

View Answer

Answer: d

Explanation: None.

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

```
#include <stdio.h>
#define COLD
int main()
{
    #ifdef COLD
    printf("COLD\t");
    #undef COLD
    #endif
    #ifdef COLD
    printf("HOT\t");
    #endif
}
```

a) HOT

b) COLD

c) COLD HOT

d) No Output

View Answer

Answer: b

Explanation: None.

5. Which of the following sequences are unaccepted in C language?

a)

```
  #if
  #else
  #endif
```

b)

```
  #if
  #elif
  #endif
```

c)

```
  #if
  #if
  #endif
```

d)

```
  #if
  #undef
  #endif
```

View Answer

Answer: c

Explanation: None.

6. In a conditional inclusion, if the condition that comes after the if is true, then what will happen during compilation?

a) Then the code up to the following #else or #elif or #endif is compiled

b) Then the code up to the following #endif is compiled even if #else or #elif is present

c) Then the code up to the following #eliif is compiled

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

7. Conditional inclusion can be used for ___________

a) Preventing multiple declarations of a variable

b) Check for existence of a variable and doing something if it exists

c) Preventing multiple declarations of same function

d) All of the mentioned

View Answer

Answer: d

Explanation: None.

8. The #elif directive cannot appear after the preprocessor #else directive.

a) True

b) False

View Answer

Answer: a

Explanation: None.

1. For each #if, #ifdef, and #ifndef directive.

a) There are zero or more #elif directives

b) Zero or one #else directive

c) One matching #endif directive

d) All of the mentioned

View Answer

Answer: d

Explanation: None.

2. The #else directive is used for _________

a) Conditionally include source text if the previous #if, #ifdef, #ifndef, or #elif test fails

b) Conditionally include source text if a macro name is not defined

c) Conditionally include source text if a macro name is defined

d) Ending conditional text

View Answer

Answer: a

Explanation: None.

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

```
#include <stdio.h>
#define MIN 0
#if MIN
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN);
    return 0;
}
```

a) 10 0

b) Compile time error

c) Undefined behaviour

d) None of the mentioned

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    #define MIN 0
    #ifdef MIN
    #define MAX 10
    #endif
    int main()
    {
        printf("%d %d\n", MAX, MIN);
        return 0;
    }
```

a) 10 0

b) Compile time error

c) Undefined behaviour

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    #define MIN 0
    #if defined(MIN) + defined(MAX)
    #define MAX 10
    #endif
    int main()
    {
        printf("%d %d\n", MAX, MIN);
        return 0;
    }
```

a) 10 0

b) Compile time error

c) Undefined behaviour

d) Somegarbagevalue 0

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    #define MIN 0
    #if defined(MIN) - (!defined(MAX))
    #define MAX 10
    #endif
    int main()
    {
        printf("%d %d\n", MAX, MIN);
        return 0;
    }
```

a) 10 0

b) Compile time error

c) Undefined behaviour

d) Somegarbagevalue 0

View Answer

Answer: b

Explanation: None.

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

```
    #include <stdio.h>
    #define MIN 0
    #ifdef(MIN)
    #define MAX 10
    #endif
    int main()
    {
        printf("%d %d\n", MAX, MIN);
        return 0;
    }
```

a) 10 0

b) Compile time error

c) Run time error

d) Preprocessor error

View Answer

Answer: d

Explanation: None.

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

```
    #include <stdio.h>
    #define MIN 0);
    #ifdef MIN
    #define MAX 10
    #endif
    int main()
    {
        printf("%d %d\n", MAX, MIN
        return 0;
    }
```

a) 10 0

b) Compile time error due to illegal syntax for printf

c) Undefined behaviour

d) Compile time error due to illegal MIN value

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        char *p = NULL;
        char *q = 0;
        if (p)
            printf(" p ");
        else
            printf("nullp");
        if (q)
            printf("q\n");
        else
            printf(" nullq\n");
    }
```

a) nullp nullq

b) Depends on the compiler

c) x nullq where x can be p or nullp depending on the value of NULL

d) p q

View Answer

Answer: a

Explanation: None.

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

```
#include <stdio.h>
int main()
{
    int i = 10;
    void *p = &i;
    printf("%d\n", (int)*p);
    return 0;
}
```

a) Compile time error

b) Segmentation fault/runtime crash

c) 10

d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

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

```
#include <stdio.h>
int main()
{
    int i = 10;
    void *p = &i;
    printf("%f\n", *(float*)p);
    return 0;
}
```

a) Compile time error

b) Undefined behaviour

c) 10

d) 0.000000

View Answer

Answer: d

Explanation: None.

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

```
#include <stdio.h>
int *f();
int main()
{
    int *p = f();
    printf("%d\n", *p);
}
int *f()
{
    int *j = (int*)malloc(sizeof(int));
    *j = 10;
    return j;
}
```

a) 10

b) Compile time error

c) Segmentation fault/runtime crash since pointer to local variable is returned

d) Undefined behaviour

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int *f();
    int main()
    {
        int *p = f();
        printf("%d\n", *p);
    }
    int *f()
    {
        int j = 10;
        return &j;
    }
```

a) 10

b) Compile time error

c) Segmentation fault/runtime crash

d) Undefined behaviour

View Answer

Answer: a

Explanation: We are returning address of a local variable which should not be done. In this specific instance, we are able to see the value of 10, which may not be the case if we call other functions before calling printf() in main().

6. Comment on the following pointer declaration.

```
int *ptr, p;
```

a) ptr is a pointer to integer, p is not

b) ptr and p, both are pointers to integer

c) ptr is a pointer to integer, p may or may not be

d) ptr and p both are not pointers to integer

View Answer

Answer: a

Explanation: None.

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

```
    #include <stdio.h>
    int main()
    {
        int *ptr, a = 10;
        ptr = &a;
        *ptr += 1;
        printf("%d,%d/n", *ptr, a);
    }
```

a) 10,10

b) 10,11

c) 11,10

d) 11,11

View Answer

Answer: d

Explanation: None.

1. Local variables are stored in an area called ___________
a) Heap
b) Permanent storage area
c) Free memory
d) Stack
View Answer
Answer: d
Explanation: Local variables are stored in an area called stack. Global variables, static variables and program instructions are stored in the permanent storage area. The memory space between these two regions is known a heap.

2. The size of both stack and heap remains the same during run time.
a) True
b) False
View Answer
Answer: b
Explanation: Memory can be allocated or de-allocated during the run time in the heap region. Memory bindings (allocation and de-allocation) are established and destroyed during the execution of the program. Hence we can see that the size of heap does not remain same during run time. However, the size of stack remains the same.

3. Choose the statement which is incorrect with respect to dynamic memory allocation.
a) Memory is allocated in a less structured area of memory, known as heap
b) Used for unpredictable memory requirements
c) Execution of the program is faster than that of static memory allocation
d) Allocated memory can be changed during the run time of the program based on the requirement of the program
View Answer
Answer: c
Explanation: Execution of the program using dynamic memory allocation is slower than that using static memory allocation. This is because in dynamic memory allocation, the memory has to be allocated during run time. This slows down the execution of the program.

4. Which of the following header files must necessarily be included to use dynamic memory allocation functions?
a) stdlib.h
b) stdio.h
c) memory.h
d) dos.h
View Answer
Answer: a
Explanation: stdlib.h is a header file which stands for the standard library. It consists of the declaration for dynamic memory allocation functions such as malloc(), calloc(), realloc() and free.

5. The type of linked list in which the node does not contain any pointer or reference to the previous node is _____________
a) Circularly singly linked list
b) Singly linked list
c) Circular doubly linked list
d) Doubly linked list
View Answer
Answer: b
Explanation: A singly linked list is one in which each node has two fields, namely data field and pointer field. Data field stores the data and the pointer field points to the address of the next node.

6. Which of the following is an example for non linear data type?
a) Tree
b) Array
c) Linked list
d) Queue
View Answer

Answer: a
Explanation: A data structure is said to be linear if its elements form a sequence or a linear list. For example array, linked list, queue, stack etc. Elements in a non linear data structure do not form a sequence. For example Trees, graphs etc.

7. Queue data structure works on the principle of ____________
a) Last In First Out (LIF0)
b) First In Last Out (FILO)
c) First In First Out (FIFO)
d) Last In Last Out (LILO)
View Answer
Answer: c
Explanation: Queue is a linear data structure which works on the principle of first in first out. This means that the first element to be inserted in a queue will be the first one to be removed.

8. Which of the following is an example of static memory allocation?
a) Linked list
b) Stack
c) Queue
d) Array
View Answer
Answer: d
Explanation: Array is an example of static memory allocation whereas linked list, queue and stack are examples for dynamic memory allocation.

9. Array is preferred over linked list for the implementation of ________
a) Radix sort
b) Insertion sort
c) Binary search
d) Polynomial evaluation
View Answer
Answer: c
Explanation: When we try to implement binary search using linked list, the traversal steps per element increases in order to find the middle element. Thus, this process is slow and inefficient. This process is much faster using an array, hence it is preferable to use an array for the implementation of binary search.

10. The advantage of using linked lists over arrays is that ________
a) Linked list is an example of linear data structure
b) Insertion and deletion of an element can be done at any position in a linked list
c) Linked list can be used to store a collection of homogenous and heterogeneous data types
d) The size of a linked list is fixed
View Answer
Answer: b
Explanation: Insertion and deletion in a linked list can be done at any position. On the other hand, in an array, to insert an element at a specific position, the rest of the elements have to be moved one position to the left and to delete an element, all the elements after the deleted element have to be moved one position to the rig

1. What will be the output of the following C code if the input entered as first and second number is 5 and 6 respectively?

```
#include<stdio.h>
#include<stdlib.h>
main()
{
    int *p;
    p=(int*)calloc(3*sizeof(int));
    printf("Enter first number\n");
    scanf("%d",p);
    printf("Enter second number\n");
    scanf("%d",p+2);
    printf("%d%d",*p,*(p+2));
    free(p);
}
```

a) 56
b) Address of the locations where the two numbers are stored
c) 57
d) Error
View Answer
Answer: d
Explanation: The above code results in an error. This is because the syntax of the function calloc() is incorrect. In order to rectify this error, we must write: calloc(3,sizeof(int));

2. In the function malloc(), each byte of allocated space is initialized to zero.
a) True
b) False
View Answer
Answer: b
Explanation: In the function malloc(), allocated space is initialized to junk values. In calloc(), each byte of allocated space is initialized to zero.

3. Which of the following functions allocates multiple blocks of memory, each block of the same size?
a) malloc()
b) realloc()
c) calloc()
d) free()
View Answer
Answer: c
Explanation: malloc() allocates a single block of memory whereas calloc() allocates multiple blocks of memory, each block with the same size.

4. A condition where in memory is reserved dynamically but not accessible to any of the programs is called _______________
a) Memory leak
b) Dangling pointer
c) Frozen memory
d) Pointer leak
View Answer
Answer: a
Explanation: If we allocate memory dynamically in a function (malloc, calloc, realloc), the allocated memory will not be de-allocated automatically when the control comes out of the function. This allocated memory cannot be accessed and hence cannot be used. This unused inaccessible memory results in a memory leak.

5. What will happens if the statement free(a) is removed in the following C code?

```
#include<stdio.h>
#include<stdlib.h>
main()
{
    int *a;
    a=(int*)malloc(sizeof(int));
    *a=100;
    printf("*a%d",*a);
    free(a);
    a=(int*)malloc(sizeof(int));
    *a=200;
    printf("a%p",a);
    *a=200;
    printf("a%d",*a);
}
```

a) Error
b) Memory leak
c) Dangling pointer arises
d) 200 is printed as output
View Answer
Answer: b
Explanation: The pointer 'a' points to the recent value 200, making the memory allocated earlier inaccessible. Hence, the memory where the value 100 is inaccessible and cannot be freed. Therefore, memory leak occurs.

6. The incorrect statement with respect to dangling pointers is ___________
a) Pointer pointing to non-existent memory location is called dangling pointer
b) When a dynamically allocated pointer references the original memory after it has been freed, a dangling pointer arises
c) If memory leak occurs, it is mandatory that a dangling pointer arises
d) Dangling pointer may result in segmentation faults and potential security risks
View Answer
Answer: c
Explanation: Memory leak and dangling pointers are not inter dependent. Hence, when memory leak occurs, it is not mandatory that a dangling pointer arises

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

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    char *p = calloc(100, 1);
    p = "welcome";
    printf("%s\n", p);
}
```

a) error
b) welcome
c) memory location stored by the pointer
d) junk value
View Answer

Answer: b
Explanation: There is no error in the above code. The format specifier being %s, address is not returned. Hence, welcome is the output.

8. In the function realloc(), if the new size of the memory block is larger than the old size, then the added memory ____________
a) is initialized to junk values
b) is initialized to zero
c) results in an error
d) is not initialized
View Answer
Answer: d
Explanation: The function realloc() changes the size of a particular memory block. If the new size is larger than the old size, the added memory is not initialized.

9. The free() function frees the memory state pointed to by a pointer and returns ____________
a) the same pointer
b) the memory address
c) no value
d) an integer value
View Answer
Answer: c
Explanation: The free() function frees the memory state pointed by a pointer and returns no value.

10. The following C code is an example of __________

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
main()
{
    char *p,*q;
    p=(char*)malloc(3*sizeof(char));
    q=p;
    strcpy(p,"hello");
    printf("p=%s",p);
    printf("q=%s",q);
    free(q);
    q=NULL;
    gets(p);
    gets(q);
    printf("%s",p);
    printf(“%s”,q);
}
```

a) Memory leak
b) Dangling pointer
c) Static memory allocation
d) Linked list
View Answer
Answer: b
Explanation: In the above code, the pointer p, points to a memory location which has been freed. Hece the above code is an example of dangling pointer.

1. What will be the output of the following C code if it is executed on a 32 bit processor?

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int *p;
    p = (int *)malloc(20);
    printf("%d\n", sizeof(p));
    free(p);
    return 0;
}
```

a) 2
b) 4
c) 8
d) Junk value
View Answer
Answer: b
Explanation: The size of a pointer is 2 bytes on a 16 bit platform, 4 bytes on a 32 bit platform and 8 bytes on a 64 bit platform.

2. The number of arguments taken as input which allocating memory dynamically using malloc() is ___________
a) 0
b) 1
c) 2
d) 3
View Answer
Answer: b
Explanation: An example of memory allocated using malloc():
(int*)malloc(3*sizeof(int)
It is clear from the above example that malloc() takes only one argument as input, that is the number of bytes to be allocated.

3. Suppose we have a one dimensional array, named 'x', which contains 10 integers. Which of the following is the correct way to allocate memory dynamically to the array 'x' using malloc()?
a) x=(int*)malloc(10);
b) x=(int*)malloc(10,sizeof(int));
c) x=malloc(int 10,sizeof(int));
d) x=(int*)malloc(10*sizeof(int));
View Answer
Answer: d
Explanation: According to the syntax of malloc, the correct way to do the specified operation is: x=(int*)malloc(10*sizeof(int)); This operation can also be performed using calloc(). In that case, the method will be: x=(int*)calloc(10,sizeof(int));

4. What will be the error (if any) in the following C code?

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
    char *p;
    *p = (char)calloc(10);
    strcpy(p, "HELLO");
    printf("%s", p);
    free(p);
    return 0;
}
```

a) No error
b) Error in the statement: strcpy(p,"HELLO");
c) Error in the statement: *p=(char)calloc(10);
d) Error in the statement: free(p);
View Answer
Answer: c
Explanation: The syntax for dynamically allocating memory using calloc() is incorrect. Hence, this code results in an error. The correct syntax for calloc is: void*calloc(size_t n, size_t size);

5. If malloc() and calloc() are not type casted, the default return type is ___________
a) void*
b) void**
c) int*
d) char*
View Answer
Answer: a
Explanation: If malloc() and calloc() are not type casted, they return a pointer of the type void.

6. Pick out the correct statement with respect to the heap.
a) Local variables are stored on the heap
b) Static variables are stored on the heap
c) Heap is the data structure which is used to implement recursive function calls
d) Everything on the heap is anonymous
View Answer
Answer: a
Explanation: Local variables are stored on the stack. Static variables are stored in the permanent storage area. Stack is the data structure used to implement recursive function calls. Hence, it is true that everything on heap s anonymous.

7. What will be the output of the following C code? (Given that the size of array is 4 and new size of array is 5)

```
#include<stdio.h>
#include<stdlib.h>
main()
{
    int *p,i,a,b;
    printf("Enter size of array");
    scanf("%d",&a);
    p=(int*)malloc(a*sizeof(int));
    for(i=0;i<a;i++)
    printf("%d\n",i);
    printf("Enter new size of array");
    scanf("%d",&b);
    realloc(p,b);
    for(i=0;i<b;i++)
    printf("%d\n",i);
    free(p);
}
```

a)

```
1234
12345
```

b) Error
c)

```
0123
01234
```

d)

```
0123
12345
```

View Answer
Answer: c
Explanation: In the above code, we are reallocating memory. When the size of the array is 4, the output is 0123. When the size of the array is changed to 5, the output is 01234.
Hence, the output is:
0123
01234

8. When the pointer is NULL, then the function realloc is equivalent to the function ___________
a) malloc
b) calloc
c) free
d) alloc
View Answer
Answer: a
Explanation: If pointer is NULL, the call to the function realloc is equal to malloc(size), for any value of size. If size is equal to zero, then the pointer is not NULL and the call is equivalent to free(pointer).

9. Garbage collector frees the programmer from worrying about ___________
a) Dangling pointers
b) Creating new objects
c) Memory leak

d) Segmentation errors
View Answer
Answer: c
Explanation: A garbage collector is a program that automatically removes unwanted data held temporarily in the memory during processing. Hence it frees the programmer from worrying about memory leaks.

10. If the space in memory allocated by malloc is not sufficient, then an allocation fails and returns ___________
a) NULL pointer
b) Zero
c) Garbage value
d) The number of bytes available
View Answer
Answer: a
Explanation: A NULL pointer is returned when the memory allocated by malloc dynamically is insufficient.

1. The preprocessor directive used to give additional information to the compiler, beyond which is conveyed in the language _____________
a) #include
b) #define
c) #pragma
d) #elif
View Answer
Answer: c
Explanation: The preprocessor directive #pragma is used to give additional information to the compiler, other than what is conveyed in the language itself.

2. What will be the output of the following C code, if it is run on a 32 bit platform?

```
#include<stdio.h>
#pragma(1)
struct test
{
    int i;
    char j;
};
main()
{
    printf("%d",sizeof(struct test));
}
```

a) Error
b) 1
c) 4
d) 8
View Answer
Answer: d
Explanation: #pragma pack(n), where n is the number of alignment in bytes. #pragma pack(1) is the directive for the compiler to pack the structure.

3. In the directive, #pragma pack(n), which of the following is not a valid value of n?
a) 1

b) 2
c) 3
d) 4
View Answer
Answer: c
Explanation: Valid arguments are 1,2,4 and 8. 3 is not a valid value for n.

4. Which of the following attributes is used to specify that the minimum required memory to be used to represent the types?
a) packed
b) aligned
c) unused
d) deprecated
View Answer
Answer: a
Explanation: The keyword __attribute__ allows you to specify special attributes of struct type. 6 attributes are currently defined for types: aligned, packed, transparent_union, unused, deprecated and may_alias. The attribute “packed” is used to specify that the minimum required memory to be used to represent the types.

5. In the directive #pragma pack(n), if the value of ‘n’ is given to be 5, then what happens?
a) Error
b) Warning but no error
c) Executes the pragma statement
d) Ignores the pragma statement and executes the program
View Answer
Answer: d
Explanation: Valid values for n are 1,2,4 and 8. If the value of n is one that the compiler does not recognize, then it simply ignores the pragma statement without throwing any error or warning.

6. The correct syntax of the attribute packed is __________
a) __attribute__((packed));
b) _attribute(packed);
c) _attribute_((packed));
d) __attribute__(packed);
View Answer
Answer: a
Explanation: The correct syntax of the attribute packed is: __attribute__((packed));

7. The pragma ____________________ is used to remove an identifier completely from a program.
a) GNU piston
b) GCC poison
c) GNU poison
d) GCC piston
View Answer
Answer: b
Explanation: There are several pragmas defines. One such pragma is GCC poison which is used to remove an identifier.

8. The function of __attribute__((packed)); can also be performed using _________
a) #pragma pack(1);
b) #pragma pack(2);
c) #pragma pack(4);
d) #pragma pack(8);
View Answer
Answer: a
Explanation: __attribute((packed)); and #pragma(1) are used to perform the same function, that is, to direct the compiler to pack the structure.

9. #pragma GCC poison should be followed by a list of identifiers that are _________
a) even in number
b) odd in number
c) valid
d) invalid
View Answer
Answer: d
Explanation: #pragma poison GCC is used to remove an identifier from a program. It should be followed by a list of identifiers which are not valid in the program, for example: scanf, printf etc.

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

```
#include<stdio.h>
#pragma GCC poison printf
main()
{
    printf("sanfoundry");
    return 0;
}
```

a) error is thrown
b) sanfoundry is printed
c) warning but no error
d) yrdnoufnas is printed
View Answer
Answer: a
Explanation: The code shown above results in an error: attempt to use poisoned printf
When the above program is compiled, it results in an error since #pragma was used to specify that the identifier printf should not be used in the program.

1. Which of the following is a stringizing operator?
a) < >
b) #
c) %
d) ##
View Answer
Answer: b
Explanation: # is the stringizing operator. It allows formal arguments within a macro definition to be converted to a string.

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

```
#define sanfoundry(s,n) #s #n
```

```
main()
{
    printf(sanfoundry(hello,world));
}
```

a) sanfoundry(hello,world)
b) sanfoundry
c) hello,world
d) helloworld
View Answer
Answer: d
Explanation: The output to this code will be helloworld because when we use the stringizing operator, the resulting string will automatically be concatenated (combined) with any adjacent strings.

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

```
#define display(text) printf(#text "@")
main()
{
    display(hello.);
    display(good morning!);
}
```

a) hello.@good morning!
b) error
c) hello.good morning!@
d) hello.@good morning!@
View Answer
Answer: d
Explanation: Each actual argument is converted into string within the printf function. Each argument is concatenated with '@', which is written as a separate string within the macro definition.

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

```
#define display(a) #a
main()
{
    printf(display("56#7"));
}
```

a) Error
b) "56#7"
c) 56#7
d) 567
View Answer
Answer: b
Explanation: In this case, it is not necessary for the argument in the printf function to be enclosed in double quotes. However, if the argument is enclosed in double quotes, no error is thrown. The output of the code shown will be "56#7".

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

```
#define HELLO(a) #a
```

```
main()
{
    printf(HELLO(good        morning));
}
```

a) good morning
b) goodmorning
c) good morning
d) error
View Answer
Answer: c
Explanation: The output of the code shown above will be: good morning
In the resulting string, the consecutive blank spaces are replaced by a single blank space when we use the stringizing operator.

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

```
#include <stdio.h>
#define sanfoundry(x)   #x
int main()
{
    int marks=100;
    printf("value of %s is = %d\n",sanfoundry(marks),marks);
    return 0;
}
```

a) error
b) value of marks=100
c) value of=100
d) 100
View Answer
Answer: b
Explanation: In the code shown above, the variable name(marks) is passed as an argument. By using the # operator, we can print the name of the variable as a string.

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

```
#define hello(c) #c
main()
{
    printf(hello(i,am));
}
```

a) i,am
b) iam
c) i am
d) error
View Answer
Answer: d
Explanation: The above code will result in an error. This is because we have passed to arguments to the macro hello, but it should be talking only one.

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

```
#define hello(c,d) #c #d
main()
```

```
{
    printf(hello(i,"am"));
}
```

a) iam
b) i“am”
c) am
d) “am”
View Answer
Answer: b
Explanation: The output for the following code will be i”am”. Since 2 arguments are passed and the macro hello takes two arguments, there is no error.

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

```
#define F abc
#define B def
#define FB(arg) #arg
#define FB1(arg) FB(arg)
main()
{
    printf(FB(F B));
    FB1(F B);
}
```

a) F B
b) Error
c) FB
d) “FB”
View Answer
Answer: a
Explanation: The argument F B(only one space between F and B) is passed to the macro FB. This argument is converted to a string with the by the stringizing operator. Thus F B is printed.

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

```
#define display(text) "$" #text
main()
{
    printf(display(hello         world));
}
```

a) hello world
b) $helloworld
c) $hello world
d) error
View Answer
Answer: c
Explanation: The output of the code shown above is $hello world
The argument “hello   world” is passed to the macro text. The symbol “$” is present from before. In the resulting string, all the blank spaces are replaced by a single blank space. In addition to this, “$” is concatenated to the beginning of the resultant string.