INDEX

C Questions

OPERATORS

1) Which is not a bitwise operator?

1. &
2. |
3. <<
4. &&

2) Predict the output of the following program.

```
#include <stdio.h>
int main()
{
    int a=10;
    int b=2;
    int c;
    c=(a & b);
    printf("c= %d",c);
    return 0;
}
```

3) Predict the output of the following program.

```
#include <stdio.h>
#define MOBILE  0x01
#define LAPPY   0x02
int main()
{
    unsigned char  item=0x00;
    item |=MOBILE;
```

```
        item |=LAPPY;
        printf("I have purchased ...:");
        if(item & MOBILE){
                printf("Mobile, ");
        }
        if(item & LAPPY){
                printf("Lappy");
        }
        return 1;
}
```

1. I have purchased …:
2. I have purchased ...:Mobile, Lappy
3. I have purchased …:Mobile,
4. I have purchased …:Lappy

4) Predict the output of the following program.

```
#include <stdio.h>
int main()
{
        char var=0x04;
       var = var | 0x04;
       printf("%d,",var);
       var |= 0x01;
       printf("%d",var);

        return 0;
}
```

1. 8,9

2. 4,5
3. 8,8
4. 4,4

5) Predict the output of the following program.

```
#include <stdio.h>
int main()
{
    char flag=0x0f;
    flag &= ~0x02;
    printf("%d",flag);
    return 0;
}
```

1. 13
2. d
3. 22
4. 1

6) Consider the given statement:

int x = 10 ^ 2

What will be the value of x?

1. 5
2. 6
3. 7
4. 8

7) Predict the output of the following program.

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

```
int main()
{
   int x=10;
   x &= ~2;
   printf("x= %d",x);
   return 0;
}
```

1. x= 10
2. x= 8
3. x= 12
4. x= 0

8) Which Bitwise Operator can be used to check whether a number is EVEN or ODD quickly?

1. Bitwise AND (&)
2. Bitwise OR (|)
3. Bitwise XOR (^)
4. Bitwise NOT (~)

9) Which statement is suitable to check 3rd (count from 0) bit is high (set) or not?

1. (num & (1<<3))
2. (num & 0x08)
3. (num & 0x03)
4. Both (1) and (2)

10) Left shift (<<) and Right shift (>>) operators are equivalent to _____by 2.

Choose the correct words...

1. Multiplication and Division
2. Division and Multiplication

3. Multiplication and Remainder

4. Remainder and Multiplication

11) What will be the output of the following program ?

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

1. 10
2. 11
3. 0
4. ERROR

12) What will be the output of the following program ?

```
#include <stdio.h>
void main()
{
      const char var='A';
      ++var;
      printf("%c",var);
}
```

1. B
2. A
3. ERROR (because we cant increment to constant)
4. 66

13) What will be the output of the following program ?

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

1. 44
2. 45
3. 46
4. 47

14) What will be the output of the following program ?

```
#include <stdio.h>
void main()
{
    int a=10,b=2,x=0;
    x=a+b*a+10/2*a;
    printf("value is =%d",x);
}
```

1. valie is =1250
2. value is =80
3. value is =125
4. ERROR

15) What will be the output of the following program ?

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

```
void main()
{
    unsigned short var='B';
    var+=2;
    var++;
    printf("var : %c , %d ", var,var);
}
```

1. var : E, 69
2. var : E, 68
3. var : D, 68
4. var : D, 69

16) What will be the output of the following program ?

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

1. ERROR : Can not modify var.
2. ERROR : L-Value required syntax d.
3. ERROR : Expression .
4. 12

17) What will be the output of the following program ?

```
#include <stdio.h>
void main()
{
```

```
    int x=(20 || 40 ) && (10);
    printf("x= %d",x);
}
```

1. x= 60
2. x= 70
3. x= 0
4. x= 1

18) What will be the output of the following program ?

```
#include <stdio.h>
void main()
{
    int x;
    x= (printf("AA")||printf("BB"));
    printf("%d",x);
    printf("\n");
    x= (printf("AA")&& printf("BB"));
    printf("%d",x);
}
```

1. AABB1
   AABB1
2. 1
   1
3. AABB1
   AA1
4. AA1
   AABB1

19) What will be the output of the following program ?

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

1. 1,2
2. 3,2
3. 0,0
4. 2,3

20) What will be the output of the following program ?

```
#include <stdio.h>
void main(){
  int intVar=20,x;
  x= ++intVar,intVar++,++intVar;
  printf("Value of intVar=%d, x=%d",intVar,x);
}
```

1. Value of intVar=23, x=21
2. Value of intVar=23, x=23
3. Value of intVar=21, x=21
4. ERROR

21) What will be the output of the following program ?

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

```
    char val=250;
    int  ans;
    ans= val+ !val + ~val + ++val;
    printf("%d",ans);
    return 0;
}
```

1. -5
2. -6
3. 0
4. 6

22) 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

23) What will be the output of the following program?

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

```
    int x;
    x=100,30,50;
    printf("x=%d\n",x);
    x=(100,30,50);
    printf("x=%d\n",x);
    return 0;
}
```

1. x=100
   x=100
2. x=100
   x=50
3. x=50
   x=50
4. x=50
   x=100

24) What will be the output of the following program ?

```
#include <stdio.h>
int main()
{
    int i=-1,j=-1,k=0,l=2,m;
    m=i++&& j++&&k++||l++;
    printf("%d %d %d %d %d",i,j,k,l,m);
    return 0;
}
```

1. 0 0 1 2 1
2. 0 0 1 3 2
3. 0 0 1 3 1

4. 0 1 1 3 1

25) What will be the output of the following program ?

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

1. ERROR
2. value of var= -10
value of var= 10
3. value of var= 10
value of var= 10
4. value of var= 10
value of var= 11

26) What will be the value of x and y ?

```
#include <stdio.h>
int main()
{
   int x,y;

   x=(100,200);
```

```
    y=100,200;
    printf("x=%d,y=%d",x,y);
    return 0;
}
```

1. x=100,y=200
2. x=200,y=200
3. ERROR
4. x=200,y=100

27) Arrange the operators according to their precedence: +, %, ->, =

1. ->, %, +, =
2. =, +, %, ->
3. %, +, =, ->
4. %, ->, =, +

28) Toggle a given range of bits

For example, take the number 245. The equivalent binary format is 11110101and the range is 4 to 7. So, the output should be 000001010 which is 5 in decimal.

29) How to check if a particular bit is set or not in a number?

30) How to represent the above all things in MACRO for Real-time code?

31) Write MACRO to Swap the bytes in 16 bit Integer Variable.

32) Write MACRO to Swap the bytes in a 32-bit Integer Variable.

33) Write MACRO to Swap the bytes in a 64-bit Integer Variable.

34) Find whether the number is odd or even

35) Clear the last right side set a bit of a number

36) Check if the number is a power of 2

37) Count the number of set bits in a number

38) Swap two bits at a given position in an integer

39) Swap all even and odd bits

40). Write a program that enters temperature in Celsius and converts that into Fahrenheit.

41). Write a program that accepts the radius of a circle and calculates the area and perimeter of that circle.

42. Write a program to accept a number in decimal and print the number in octal and hexadecimal.

43. Write a program to accept any number and print the value of remainder after dividing it by 3.

44. Accept any two numbers, if the first number is greater than second then print the difference of these two numbers, otherwise print their sum. Write this program using a ternary operator.

45. Write a program that accepts marks in five subjects and calculates the total percentage marks.

46. Can you swap two variables without using a temporary variable? Show the code.

47. Implement a program to check if a number is even or odd using the modulus operator.

48. Write a C program that uses the bitwise operators to check if a given positive integer is divisible by both 6 and 9.

49. Develop a program to find the maximum of three numbers using the conditional operator

50. Implement a program that checks if a number is less than 50 without using the less than operator.

51. What would be the value of 'a':

a. int a = 10/45*23%45/(45%4*21)

b. float a = 10+45.0*23-45+(4*21.0)

52. Implement a C program to check if the given number is a palindrome in binary representation using bitwise operators.

53. Write a C program to rotate the bits of a given number to the left by a specified number of positions using bitwise operators

54. Implement a C program to reverse the bits of a given number using bitwise operators.

55. Write a C program to check if the given number has alternate bits set or not using bitwise operators.

56. Implement a C program to count the number of bits needed to convert integer A to integer B using bitwise operators.

57. Write a C program to check if a given number is a power of 4 using bitwise operators

58. Implement a C program to swap adjacent bits of a given number using bitwise operators.

59. Write a C program to check if the given number is a perfect square using bitwise operators.

60. Implement a C program to swap the first and last bits of a given number using bitwise operators.

61. Implement a C program to count the number of bits that need to be flipped to convert integer A to integer B using bitwise operators.

62. Write a C program to check if the given number is a power of 8 using bitwise operators.

63. Implement a C program to set all the bits at odd positions in a given number using bitwise operators.

64. Write a C program to count the number of bits set to 1 in the binary representation of the sum of two numbers using bitwise operators.

65. Implement a C program to check if the given number is a power of 16 using bitwise operators.

66 Implement a C program to swap the bits at even and odd positions of a given number using bitwise operators.

67. Implement a C program to toggle the bits at odd positions in the binary representation of a given number using bitwise operators.

CONTROL STATEMENTS

1. WRITE A C PROGRAM TO ACCEPT TWO INTEGERS AND CHECK WHETHER THEY ARE EQUAL OR NOT?

2. WRITE A C PROGRAM TO CHECK WHETHER A GIVEN NUMBER IS EVEN OR ODD?

3. WRITE A C PROGRAM TO CHECK WHETHER A GIVEN NUMBER IS POSITIVE OR NEGATIVE?

4. WRITE A C PROGRAM TO FIND WHETHER A GIVEN YEAR IS A LEAP YEAR OR NOT?

5. WRITE A C PROGRAM TO READ THE AGE OF A CANDIDATE AND DETERMINE WHETHER HE IS ELIGIBLE TO CAST HIS/HER OWN VOTE?

6. WRITE A C PROGRAM TO READ THE VALUE OF AN INTEGER M AND DISPLAY THE VALUE OF N IS 1 WHEN M IS LARGER THAN 0, 0 WHEN M IS 0 AND -1 WHEN M IS LESS THAN 0.

7. WRITE A C PROGRAM TO FIND THE LARGEST OF THREE NUMBERS?

8. WRITE A C PROGRAM TO CHECK WHETHER A CHARACTER IS A VOWEL OR CONSONANT?

9. WRITE A C PROGRAM TO CHECK WHETHER A CHARACTER IS AN ALPHABET OR NOT?

10. WRITE A C PROGRAM TO FIND MINIMUM OR MAXIMUM BETWEEN TWO NUMBERS?

11. WRITE A C PROGRAM TO ENTER WEEK NUMBER AND PRINT DAY OF WEEK?

12. WRITE A C PROGRAM TO CHECK WHETHER A CHARACTER IS UPPERCASE OR LOWERCASE?

13. WRITE A C PROGRAM TO FIND NUMBER OF DAYS IN MONTH?

14. WRITE A C PROGRAM TO FIND MAXIMUM BETWEEN TWO NUMBERS USING SWITCH CASE?

15. WRITE A C PROGRAM TO PRINT EVEN NUMBERS BETWEEN 1 TO 20 USING A FOR LOOP?

16. WRITE A C PROGRAM TO CALCULATE THE SUM OF NUMBERS FROM 1 TO 100 USING A WHILE LOOP?

17. WRITE A C PROGRAM TO FIND THE FACTORIAL OF A GIVEN NUMBER USING A FOR LOOP?

18. WRITE A C PROGRAM TO CHECK WHETHER A GIVEN NUMBER IS PRIME OR NOT USING A WHILE LOOP?

19. WRITE A C PROGRAM TO FIND THE SUM OF DIGITS OF A NUMBER USING A WHILE LOOP?

20. WRITE A C PROGRAM TO PRINT FIBONACCI SERIES UP TO N TERMS USING A FOR LOOP?

21. WRITE A C PROGRAM TO REVERSE A GIVEN NUMBER USING A WHILE LOOP?

22. WRITE A C PROGRAM TO FIND THE LARGEST ELEMENT IN AN ARRAY USING A FOR LOOP?

23. WRITE A C PROGRAM TO FIND THE SMALLEST ELEMENT IN AN ARRAY USING A WHILE LOOP?

24. WRITE A C PROGRAM TO PRINT ALL THE ELEMENTS OF AN ARRAY USING A FOR LOOP?

25. WRITE A C PROGRAM TO FIND THE SUM OF ELEMENTS IN AN ARRAY USING A WHILE LOOP?

26. WRITE A C PROGRAM TO COUNT THE NUMBER OF VOWELS IN A GIVEN STRING USING A FOR LOOP?

27. WRITE A C PROGRAM TO COUNT THE NUMBER OF WORDS IN A GIVEN STRING USING A WHILE LOOP?

28. WRITE A C PROGRAM TO CHECK WHETHER A GIVEN STRING IS A PALINDROME OR NOT USING A FOR LOOP?

29. WRITE A C PROGRAM TO CONCATENATE TWO STRINGS WITHOUT USING LIBRARY FUNCTIONS USING A WHILE LOOP?

30. WRITE A C PROGRAM TO FIND THE LENGTH OF A STRING USING A FOR LOOP?

31. WRITE A C PROGRAM TO CONVERT A STRING TO UPPERCASE USING A WHILE LOOP?

32. WRITE A C PROGRAM TO FIND THE POWER OF A NUMBER USING A FOR LOOP?

33. WRITE A C PROGRAM TO FIND THE FACTORIAL OF A NUMBER USING A WHILE LOOP?

34. WRITE A C PROGRAM TO FIND THE GCD OF TWO NUMBERS USING A WHILE LOOP?

35. WRITE A C PROGRAM TO FIND THE LCM OF TWO NUMBERS USING A FOR LOOP?

36. WRITE A C PROGRAM TO PRINT THE MULTIPLICATION TABLE OF A GIVEN NUMBER USING A FOR LOOP?

37. WRITE A PROGRAM IN C TO PRINT THE ARMSTRONG NUMBERS BETWEEN 1 AND 1000 USING A FOR LOOP?

38. WRITE A PROGRAM IN C TO IMPLEMENT A SIMPLE CALCULATOR USING SWITCH-CASE STATEMENTS?

39. WRITE A PROGRAM IN C TO CHECK WHETHER A GIVEN NUMBER IS A PALINDROME OR NOT USING WHILE LOOPS AND IF-ELSE STATEMENTS?

40. WRITE A PROGRAM IN C TO FIND THE SUM OF ELEMENTS IN THE LOWER TRIANGULAR MATRIX USING LOOPS AND IF-ELSE STATEMENTS?

41. WRITE A PROGRAM IN C TO FIND THE SUM OF ELEMENTS IN THE UPPER TRIANGULAR MATRIX USING LOOPS AND IF-ELSE STATEMENTS?

42. WRITE A PROGRAM IN C TO REMOVE DUPLICATE ELEMENTS FROM AN ARRAY USING LOOPS AND IF-ELSE STATEMENTS?

43. WRITE A PROGRAM IN C TO SORT AN ARRAY OF INTEGERS IN ASCENDING ORDER USING LOOPS AND IF-ELSE STATEMENTS?

44. WRITE A PROGRAM IN C TO SORT AN ARRAY OF INTEGERS IN DESCENDING ORDER USING LOOPS AND IF-ELSE STATEMENTS?

45. WRITE A PROGRAM IN C TO FIND THE SECOND LARGEST ELEMENT IN AN ARRAY USING LOOPS AND IF-ELSE STATEMENTS?

46. WRITE A PROGRAM IN C TO FIND THE FREQUENCY OF EACH ELEMENT IN AN ARRAY USING LOOPS AND IF-ELSE STATEMENTS?

47. WRITE A PROGRAM IN C TO CHECK WHETHER A MATRIX IS AN IDENTITY MATRIX OR NOT USING LOOPS AND IF-ELSE STATEMENTS?

48. WRITE A C PROGRAM TO PRINT ALL THE ODD NUMBERS BETWEEN 1 TO 50 USING A FOR LOOP?

49. WRITE A PROGRAM IN C TO FIND THE SUM OF ELEMENTS IN EACH COLUMN OF A MATRIX USING LOOPS AND IF-ELSE STATEMENTS?

50. WRITE A PROGRAM IN C TO FIND THE SUM OF ELEMENTS IN EACH ROW OF A MATRIX USING LOOPS AND IF-ELSE STATEMENTS?

51. WRITE A PROGRAM IN C TO ADD TWO MATRICES USING LOOPS AND IF-ELSE STATEMENTS?

52. WRITE A PROGRAM IN C TO SUBTRACT TWO MATRICES USING LOOPS AND IF-ELSE STATEMENTS?

53. WRITE A PROGRAM IN C TO MULTIPLY TWO MATRICES USING LOOPS AND IF-ELSE STATEMENTS?

54. WRITE A PROGRAM IN C TO PRINT THE ELEMENTS OF AN ARRAY IN REVERSE ORDER USING LOOPS AND IF-ELSE STATEMENTS?

55. WRITE A PROGRAM IN C TO CHECK WHETHER TWO ARRAYS ARE EQUAL OR NOT USING LOOPS AND IF-ELSE STATEMENTS?

56. WRITE A PROGRAM IN C TO FIND THE UNION OF TWO ARRAYS USING LOOPS AND IF-ELSE STATEMENTS?

57. WRITE A PROGRAM IN C TO FIND THE INTERSECTION OF TWO ARRAYS USING LOOPS AND IF-ELSE STATEMENTS?

58. WRITE A PROGRAM IN C TO FIND THE DIFFERENCE OF TWO ARRAYS USING LOOPS AND IF-ELSE STATEMENTS?

59. WRITE A PROGRAM IN C TO FIND THE MISSING NUMBER IN AN ARRAY CONTAINING NUMBERS FROM 1 TO N USING LOOPS AND IF-ELSE STATEMENTS?

60. WRITE A PROGRAM IN C TO FIND THE MAJORITY ELEMENT IN AN ARRAY USING LOOPS AND IF-ELSE STATEMENTS?

61. WRITE A C PROGRAM TO PRINT THE PATTERN OF STARS USING NESTED LOOPS?

62. WRITE A C PROGRAM TO SEARCH FOR AN ELEMENT IN AN ARRAY USING LINEAR SEARCH AND A FOR LOOP?

63. WRITE A C PROGRAM TO CHECK WHETHER A GIVEN NUMBER IS A PALINDROME IN BOTH DECIMAL AND BINARY REPRESENTATIONS USING LOOPS AND IF-ELSE STATEMENTS?

64. WRITE A C PROGRAM TO FIND THE SUM OF FIRST N NATURAL NUMBERS WHICH ARE NOT DIVISIBLE BY 3 OR 5 USING LOOPS AND IF-ELSE STATEMENTS?

65. WRITE A C PROGRAM TO FIND THE SUM OF ALL EVEN NUMBERS BETWEEN TWO GIVEN NUMBERS USING LOOPS AND IF-ELSE STATEMENTS?

66. WRITE A C PROGRAM TO FIND THE SUM OF ALL ODD NUMBERS BETWEEN TWO GIVEN NUMBERS USING LOOPS AND IF-ELSE STATEMENTS?

67. WRITE A C PROGRAM TO CHECK WHETHER A GIVEN NUMBER IS A PERFECT SQUARE OR NOT USING LOOPS AND IF-ELSE STATEMENTS.

68. WRITE A C PROGRAM TO CALCULATE THE POWER OF A NUMBER USING LOOPS AND IF-ELSE STATEMENTS?

69. WRITE A C PROGRAM TO FIND THE ASCII VALUE OF A CHARACTER USING LOOPS AND IF-ELSE STATEMENTS?

70. WRITE A C PROGRAM TO FIND THE AREA OF A CIRCLE GIVEN ITS RADIUS USING LOOPS AND IF-ELSE STATEMENTS?

71. WRITE A C PROGRAM TO FIND THE SUM OF ALL PRIME NUMBERS BETWEEN 1 AND 1000 USING LOOPS AND IF-ELSE STATEMENTS?

72. WRITE A C PROGRAM TO PRINT THE PATTERN OF STARS IN A HOLLOW SQUARE SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

73. WRITE A C PROGRAM TO PRINT THE PATTERN OF STARS IN A RIGHT TRIANGLE SHAPE USING LOOPS AND IF-ELSE STATEMENTS.

74. WRITE A C PROGRAM TO PRINT THE PATTERN OF STARS IN A MIRRORED RIGHT TRIANGLE SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

75. WRITE A C PROGRAM TO PRINT THE PATTERN OF STARS IN A PYRAMID SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

76. WRITE A C PROGRAM TO PRINT THE PATTERN OF STARS IN A MIRRORED PYRAMID SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

77. WRITE A C PROGRAM TO PRINT THE PATTERN OF STARS IN A DIAMOND SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

78. WRITE A C PROGRAM TO PRINT THE PATTERN OF STARS IN A HOLLOW DIAMOND SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

79. WRITE A C PROGRAM TO PRINT THE PATTERN OF NUMBERS IN A PYRAMID SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

80. WRITE A C PROGRAM TO PRINT THE PATTERN OF NUMBERS IN A MIRRORED PYRAMID SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

81. WRITE A C PROGRAM TO PRINT THE PATTERN OF NUMBERS IN A RIGHT TRIANGLE SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

82. WRITE A C PROGRAM TO PRINT THE PATTERN OF NUMBERS IN A MIRRORED RIGHT TRIANGLE SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

83. WRITE A C PROGRAM TO PRINT THE PATTERN OF ALPHABETS IN A PYRAMID SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

84. WRITE A C PROGRAM TO PRINT THE PATTERN OF ALPHABETS IN A MIRRORED PYRAMID SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

85. WRITE A C PROGRAM TO PRINT THE PATTERN OF ALPHABETS IN A RIGHT TRIANGLE SHAPE USING LOOPS AND IF-ELSE STATEMENTS?

86. WRITE A C PROGRAM TO FIND THE AREA OF A RECTANGLE GIVEN ITS LENGTH AND BREADTH USING LOOPS AND IF-ELSE STATEMENTS?

87. WRITE A C PROGRAM TO FIND THE SUM OF THE CUBES OF THE DIGITS OF A GIVEN NUMBER USING LOOPS AND IF-ELSE STATEMENTS?

88. WRITE A C PROGRAM TO FIND THE SUM OF THE EVEN DIGITS AND THE SUM OF THE ODD DIGITS SEPARATELY IN A GIVEN NUMBER USING LOOPS AND IF-ELSE STATEMENTS?

ARRAYS

1. Write a program in C to store elements in an array and print them.
2. Write a program in C to read n number of values in an array and display them in reverse order.
3. Write a program in C to find the sum of all elements of the array.
4. Write a program in C to copy the elements of one array into another array.
5. Write a program in C to count the total number of duplicate elements in an array.
6. Write a program in C to print all unique elements in an array.
7. Write a program in C to merge two arrays of the same size sorted in descending order.
8. Write a program in C to count the frequency of each element of an array.
9. Write a program in C to find the maximum and minimum elements in an array
10. Write a program in C to separate odd and even integers into separate arrays.
11. Write a program in C to sort elements of an array in ascending order
12. Write a program in C to sort the elements of the array in descending order
13. Write a program in C to delete an element at a desired position from an array.
14. Write a program in C to find the second largest element in an array
15. Write a program in C to find the second smallest element in an array.
16. Write a program in C for a 2D array of size 3x3 and print the matrix.
17. Write a program in C for adding two matrices of the same size.
18. Write a program in C for the subtraction of two matrices.
19. Write a program in C for the multiplication of two square matrices.
20. Write a program in C to find the transpose of a given matrix.
21. Write a program in C to find the sum of the right diagonals of a matrix.
22. Write a program in C to find the sum of the left diagonals of a matrix.
23. Write a program in C to find the sum of rows and columns of a matrix.
24. Write a program in C to print or display the lower triangular of a given matrix.
25. Write a program in C to print or display an upper triangular matrix.
26. Write a program in C to calculate the determinant of a 3 x 3 matrix

27. Write a program in C to accept two matrices and check whether they are equal.

28. Write a program in C to find the majority element of an array.

(A majority element in an array A[] of size n is an element that appears more than n/2 times (and hence there is at most one such element).

29. Write a program in C to find the missing number in a given array. There are no duplicates in the list.

30. Write a program in C to find the two repeating elements in a given array.

31. Write a program to check if a given element is present in an array.

32. Create a function to calculate the average of elements in an array

33. Write a program to count the number of even and odd elements in an array.

34. Implement a function to reverse the elements of an array.

35. Implement a function to delete an element at a specific position in an array.

36. Write a function to find the product of all elements in an array.

37. Print Square of Array Elements in C

38. Print Ascii Values using Array in C

39. C Program To Find Two Elements whose Sum is Closest to Zero

40. C Program to Find Union and Intersection of Two Arrays

41. C Program to Print all Non Repeated Elements in an Array

42. Write a program to write all the elements of 2-D Array into !-D Array in row wise.

43. Write a program to write whether a matrix is symmetric or not

44. Write a program to check if elements of an array are distinct or not.
45. Write a program to remove duplicate elements from a sorted array.
46. . Write a program to find out whether a square matrix is symmetric or not. A square matrix is symmetric if the transpose of the matrix is equal to the matrix.

47. Write a recursive function to find the sum of all even numbers in an array.
48. Write a recursive function that finds the sum of all elements of an array by repeatedly partitioning it into two almost equal parts.
49. Write a recursive function to reverse the elements of an array..
50. Write a recursive function to find whether the elements of an array are in strict ascending order or not.
51. Write a program to find the sum of rows and columns of a 2-d array and store the sums in the same array.

1	2	2	1	4	**10**
5	4	3	2	5	**19**
6	3	2	1	4	**16**
3	5	4	2	3	**17**
15	**14**	**11**	**6**	**16**	**62**

52. Write a program to print a rectangular matrix spirally.
53. Write a program to print Spiral Matrix. A spiral matrix is a n x n square matrix formed by placing the numbers 1, 2, 3, 4..... n^2 in spiral form starting from the leftmost column and topmost row. Spiral matrices can exist for both even and odd values of n. The spiral matrices for n=3, n=4, n=7 are shown below

1	2	3
8	9	4
7	6	5

n=3

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

n=4

1	2	3	4	5	6	7
24	25	26	27	28	29	8
23	40	41	42	43	30	9
22	39	48	49	44	31	10
21	38	47	46	45	32	11
20	37	36	35	34	33	12
19	18	17	16	15	14	13

n=7

54. In the previous problem, we have rotated the array to the left by one element. Now modify the previous program so that we can rotate the array by any number of elements. For example when we rotate the array by 3 elements the result would be-

1 2 3 4 5 6 7 8 9 -> 4 5 6 7 8 9 1 2 345. Write a program to partition an array such that all the negative numbers are on the left side of the array and positive numbers on the right side.

55. Write a program to create an array next_ge for an unsorted array arr containing n elements such that next_ge[i] = next greater element of arr [1] in array arr i.e. first greater element on the right of arr[i]. -1 if no such element exists.

Example-

Arr : 3 1 6 2 5 8 9 4 7

next_ge : 6 6 8 5 8 9 -1 7 -1

56. Write a program to find the kth smallest element in an array.

57. Write a program to reverse a portion of an array.

58. Write a program to modify the elements of an array such that the first element becomes the last element of the array and all other elements are shifted towards left.

1 2 3 4 5 6 7 8 9 -> 2 3 4 5 6 7 8 9 1

POINTERS

1. Write a Program to print the address of variables using the address operator.
2. Write a program to print size of pointer variables.
3. Write a program to print size of pointer variables and size of values Dereferenced by them.
4. Write a program to illustrate the dereferencing of pointers.
5. C program to illustrate pointer arithmetic
6. Write a program to declare an integer variable a, assign it a value, then declare a pointer variable, assign it the address of a, and finally print the value of a using the pointer variable.
7. Write a program to print postfix/prefix increment/decrement in a pointer variable of base type int*.
8. Write a Program to print the value and address of elements of an array using pointers notation.
9. Write a program to print the value of array elements using pointers and subscript notation.
10. Write a program to print the value and address of array elements by subscripting a pointer variable.
11. Program to understand the difference between a to an integer and a pointer to an array of integers.
12. Write a program to dereference a pointer to an array.
13. program to print the values and address of elements of 2-d array.
14. Program to print elements of a 2-D array by subscripting a pointer to an array variable.
15. Program to print elements of a 3-D array using pointer notation.
16. Write a simple program for call by value.
17. Write a simple program for call by reference.
18. Program to return more than one value from a function using call by reference.
19. Write a program to pass a 1D array to a function.
20. Create a function that swaps two numbers using pointers.
21. Implement a function that returns the length of a string using pointers
22. Write a program to find the maximum and minimum elements in an array using pointers.
23. Develop a function to reverse a string in place using pointers.

24. Write a program that calculates the sum of all elements in an integer array using pointer arithmetic.

25. Implement a function to copy one string into another using pointers, without using any standard library functions

26. Write a program to compare two strings lexicographically (like the strcmp function) using pointers.

27. Develop a function to reverse an array of integers in place using pointers.

28. Write a program to find the largest element using Dynamic Memory Allocation.

29. Write a program in C to calculate the length of a string using a pointer.

30. Write a program to swap elements using call by reference.

31. Write a program to find the factorial of a given number using pointers.

32. Write a program to count the number of vowels and consonants in a string using a pointer.

33. Write a program to sort an array using a pointer

34. Write a program to demonstrate how a function returns a pointer.

35. Write a program to compute the sum of all elements in an array using pointers

36. Write a program to print the elements of an array in reverse order.

37. Write a program to show a pointer to an array whose contents are pointers to structures.

38. Write a program to show a pointer to an array whose contents are pointers to structures.

39. Logic to search an element in an array using pointers.

40. Write a program to add two matrices using pointers.

41. Write a program to multiply two matrix using pointers

42. write a program to concatenate two strings using pointers

43. Write a program to input elements in an array and sort array using pointers.

44. Program to access dynamically allocated memory as a 1d array

45. Program to access dynamically allocate a 2-D array using a pointer to an array

46. Write a program to print array of pointer

47. Write a program to print pointer to an array

48. Program to dynamically allocate a 2-D array using array of pointers

49. Program to invoke a function using function pointer

50. Program to send a function ‘s address as an argument to other function

51. Program to pass a pointer containing functions arguments address as an argument

52. .Write the output of the following program ?

```
# include <stdio.h>
void fun(int *ptr)
{
   *ptr = 30;
}
int main()
{
  int y = 20;
  fun(&y);
  printf("%d", y);
  return 0;
}
```

53. Write the output of the following program ?

```
include <stdio.h>
void fun(int x)
{
   x = 30;
}
int main()
{
  int y = 20;
  fun(y);
  printf("%d", y);
  return 0;
```

```
}
```

54. Write the output of the following program ?

```
 # include <stdio.h>
void fun(int *ptr)
{
    *ptr = 30;
}
int main()
{
  int y = 20;
  fun(&y);
  printf("%d", y);
  return 0;
}
```

55. Write the output of the following program ?

```
 #include <stdio.h>
void changeValue(int *ptr) {
    *ptr = 30;
}
int main() {
    int val = 20;
    int *ptr = &val;
    changeValue(ptr);
    printf("%d", val);
}
```

56. Write the output of the following program ?

```
#include  <stdio.h>
#include <stdlib.h>
int main() {
   int *arr = (int *)calloc(5, sizeof(int));
   for (int i = 0; i < 5; i++) {
      *(arr + i) = i;
   }
   printf("%d", arr[3]);
   free(arr);
}
```

57. Write the output of the following program ?

```
#include<stdio.h>
void f(int *p, int *q)
{
  p = q;
  *p = 2;
}
int i = 0, j = 1;
int main()
{
  f(&i, &j);
  printf("%d %d", i, j);
  getchar();
  return 0;
}
```

58. Write the output of the following program ?

```
#include<stdio.h>
int main()
{
   int a = 12;
   void *ptr = (int *)&a;
   printf("%d", *ptr);
   getchar();
   return 0;
}
```

59. Write the output of the following program ?

```
#include<stdio.h>
int main()
{
   int a, b = 10;
   a = -b--;
   printf("a = %d, b = %d", a, b);
   return 0;
}
```

60. Write the output of the following program ?

```
int main()
{
   int array[5][5];
   printf("%d",( (array == *array) && (*array == array[0]) ));
   return 0;
}
```

61. Write the output of the following program ?

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

62. Write the output of the following program ?

```
#include <stdio.h>
int main()
{char a = 30;
 char b = 40;
 char c = 10;
 char d = (a * b) / c;
 printf ("%d ", d);
   return 0;
}
```

63. Write the output of the following program ?

```
#include <stdio.h>
int main()
{
  int arr[] = {};
```

```
 printf("%d", sizeof(arr));
 return 0;
}
```

64. Write the output of the following program ?

```
#include<stdio.h>
int main()
{
   int a[2][3] = {2,1,3,2,3,4};
   printf("Using pointer notations:\n");
   printf("%d %d %d\n", *(*(a+0)+0), *(*(a+0)+1), *(*(a+0)+2));
   printf("Using mixed notations:\n");
   printf("%d %d %d\n", *(a[1]+0), *(a[1]+1), *(a[1]+2));
   return 0;
}
```

65. Write the output of the following program ?

```
int main() {
   int a = 10, *j;
   void *k;
   j = k = &a;
   j++;
   k++;
   return 0;
}
```

66. Write the output of the following program ?

```
int main() {
```

```
    int arr[] = {10, 20, 30, 40, 50};
    int *ptr1 = arr;
    int *ptr2 = arr + 3;
    printf("%d", (ptr2 - ptr1));
    return 0;
}
```

67.
```
int main() {
    int x = 5;
    int *p = &x;
    *p = *p + 1;
    printf("%d", x);
    return 0;
}
```

68. Write the output of the following program ?

```
int main() {
    int a = 10, b = 20;
    int *p1 = &a, *p2 = &b;
    *p1 += *p2;
    *p2 = *p1 - *p2;
    *p1 = *p1 - *p2;
    printf("a=%d, b=%d", a, b);
    return 0;
}
```

69. Write the output of the following program ?

```
int main() {
```

```
    char *str = "hello";
    printf("%c", *(str+1));
    return 0;
}
```

70. Write the output of the following program ?

```
void update(int *p) {
    *p += 5;
}
int main() {
    int a = 5;
    update(&a);
    printf("%d", a);
}
```

71. Write the output of the following program ?

```
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *p = arr;
    ++*p;
    printf("%d", *p);
}
```

72. Write the output of the following program ?

```
int main() {
    char *s = "hello world";
    s[0] = 'H';
    printf("%s", s);
```

```
}
```

73. Write the output of the following program ?

```
int main() {
   int *p = (int *)malloc(sizeof(int));
   *p = 10;
   free(p);
   *p = 20;
   printf("%d", *p);
}
```

74. Write the output of the following program ?

```
int func() {
   int num = 10;
   return &num;
}
int main() {
   int *p = func();
   printf("%d", *p);
}
```

75. Write the output of the following program ?

```
int main() {
   int nums[] = {1, 2, 3, 4, 5};
   int *ptr = nums + 2;
   printf("%d", *(ptr + 1));
}
```

76. Write the output of the following program ?

```
void swap(int *x, int *y) {
   int temp = *x;
   *x = *y;
   *y = temp;
}
int main() {
   int a = 10, b = 20;
   swap(&a, &b);
   printf("a=%d, b=%d", a, b);
}
```

77. Write the output of the following program ?

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

78. Write the output of the following program ?

```
int main() {
   int x = 10;
   int *p = &x;
   int **q = &p;
```

```
    int ***r = &q;
    ***r = 20;
    printf("%d", x);
}
```

79. Write the output of the following program ?

```
int main() {
    int arr[] = {10, 20, 30, 40, 50};
    int *ptr1 = arr;
    int *ptr2 = ptr1 + 3;
    printf("%td", ptr2 - ptr1);
}
```

80.

```
void foo(int *array, int size) {
    for (int i = 0; i < size; i++) {
        *(array + i) = i * i;
    }
}
int   main()   {
    int    arr[5];
    foo(arr, 5);
    printf("%d", arr[3]);
}
```

81. Write the output of the following program ?

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

```
    char str[] = "pointer";
    char *p = str;
    p += 3;
    printf("%s", p);
}
```

82. Write the output of the following program ?

```
#include <stdio.h>
int main() {
    int a[3] = {1, 2, 3};
    int *b = a;
    *b += 2;
    *(b + 1) += 2;
    b[2] += 2;
    printf("%d %d %d", a[0], a[1], a[2]);
}
```

83. Write the output of the following program ?

```
#include <stdio.h>
void increase(int **p) {
    int q = 10;
    *p = &q;
}
int main() {
    int r = 20;
    int *ptr = &r;
    increase(&ptr);
```

```
    printf("%d", *ptr);
}
```

84. Write the output of the following program ?

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int *p = (int *)malloc(2 * sizeof(int));
    p[0] = 1; p[1] = 2;
    realloc(p, 4 * sizeof(int));
    p[2] = 3; p[3] = 4;
    printf("%d %d %d %d", p[0], p[1], p[2], p[3]);
    free(p);
}
```

85. Write the output of the following program ?

```
 #include <stdio.h>
int main() {
    char *s[] = {"knowledge", "is", "power"};
    char **ptr[] = {s+2, s+1, s};
    char ***pptr = ptr;
    printf("%s ", **++pptr);
    printf("%s ", *--*++pptr+3);
    printf("%s", **pptr+1);
}
```

86. Write the output of the following program ?

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

```
 int main() {
   int nums[] = {1, 2, 3, 4};
   int *p = nums + 3;
   printf("%d", *(p - 2));
}
```

87. Write the output of the following program ?

```
include <stdio.h>
int main()
{
   int *ptr;
   int x;
   ptr = &x;
   *ptr = 0;
   printf(" x = %dn", x);
   printf(" *ptr = %dn", *ptr);
   *ptr += 5;
   printf(" x  = %dn", x);
   printf(" *ptr = %dn", *ptr);
   (*ptr)++;
   printf(" x = %dn", x);
   printf(" *ptr = %dn", *ptr);
   return 0;
}
```

Strings

1. Write a program in C to input a string and print it.

2. Write a program in C to find the length of a string without using library functions.

3. Write a program in C to separate individual characters from a string.

4. Write a program in C to print individual characters of a string in reverse order.

5. Write a program in C to count the total number of words in a string.

6. Write a program in C to compare two strings without using string library functions.

7. Write a program in C to count the total number of alphabets, digits and special characters in a string.

8. Write a program in C to copy one string to another string.

9. Write a program in C to count the total number of vowels or consonants in a string.

10. Write a program in C to find the maximum number of characters in a string.

11. Write a C program to sort a string array in ascending order.

12. Write a program in C to read a string from the keyboard and sort it using bubble sort.

13. Write a program in C to extract a substring from a given string.

14. Write a C program to check whether a substring is present in a string.

15. Write a program in C to read a sentence and replace lowercase characters with uppercase and vice versa.

16. Write a program in C to find the number of times a given word 'the' appears in the given string.

17. Write a program in C to remove characters from a string except alphabets.

18. Write a program in C to find the frequency of characters.

19. Write a program in C to combine two strings manually.

20. Write a program in C to find the largest and smallest words in a string.

21. Write a program in C to convert a string to uppercase.

22. Write a program in C to convert a string to lowercase.

23. Write a program in C to check whether a character is a Hexadecimal Digit or not.

24. Write a program in C to check whether a letter is uppercase or not.

25. Write a program in C to replace the spaces in a string with a specific character.

26. Write a program in C to count the number of punctuation characters present in a string.

27. Write a program in C to print only the string before the new line character.

28. Write a program in C to check whether a letter is lowercase or not.

29. Write a program in C to read a file and remove the spaces between two words of its content.

30. Write a program in C to check whether a character is a digit or not.

31. Write a program in C to split strings by space into words.

32. Write a C program to find the repeated character in a string.

33. Write a C program to count each character in a given string.

34. Write a C program to convert vowels into uppercase characters in a string.

35. Write a C program to find the length of the longest substring of a given string without repeating characters.

36. A given string contains the bracket characters '(', ')', '{', '}', '<', '>', '[' and ']', Write a C program to check if the string is valid or not. The input string will be valid when open brackets and closed brackets are the same type of brackets.

37. Write a C program to multiply two positive numbers as strings. Return a string representation of the product.

38. Write a C program to reverse all the vowels present in a given string. Return the newly created string.

39. Write a C program to find the longest palindromic substring from a given string. Return the substring.

40. Write a C program to replace each lowercase letter with the same uppercase letter of a given string.

41. Write a C program to calculate the length of the longest common subsequence of two given strings.

42. Write a C program to find the length of a string.

43. Write a C program to copy one string to another string.

44. Write a C program to concatenate two strings.

45. Write a C program to compare two strings.

46. Write a C program to convert lowercase string to uppercase.

47. Write a C program to convert uppercase string to lowercase.

48. Write a C program to toggle the case of each character of a string.

49. Write a C program to find the total number of alphabets, digits or special characters in a string.

50. Write a C program to count the total number of vowels and consonants in a string.

51. Write a C program to count the total number of words in a string.

52. Write a C program to find the reverse of a string.

53. Write a C program to check whether a string is palindrome or not.

54. Write a C program to reverse order of words in a given string.

55. Write a C program to find the first occurrence of a character in a given string.

56. Write a C program to find the last occurrence of a character in a given string.

57. Write a C program to search all occurrences of a character in a given string.

58. Write a C program to count occurrences of a character in a given string.

59. Write a C program to find the highest frequency character in a string.

60. Write a C program to find the lowest frequency character in a string.

61. Write a C program to count the frequency of each character in a string.

62. Write a C program to remove the first occurrence of a character from a string.

63. Write a C program to remove the last occurrence of a character from a string.

64. Write a C program to remove all occurrences of a character from a string.

65. Write a C program to remove all repeated characters from a given string.

66. Write a C program to replace the first occurrence of a character with another in a string.

67. Write a C program to replace the last occurrence of a character with another in a string.

68. Write a C program to replace all occurrences of a character with another in a string.

69. Write a C program to find the first occurrence of a word in a given string.

70. Write a C program to find the last occurrence of a word in a given string.

71. Write a C program to search all occurrences of a word in a given string.

72. Write a C program to count occurrences of a word in a given string.

73. Write a C program to remove the first occurrence of a word from a string.

74. Write a C program to remove the last occurrence of a word in a given string.

75. Write a C program to remove all occurrences of a word in a given string.

76. Write a C program to trim leading white space characters from a given string.

77. Write a C program to trim trailing white space characters from a given string.

78. Write a C program to trim both leading and trailing white space characters from given string.

79. Write a C program to remove all extra blank spaces from given string

80. Write a function to convert a string into uppercase.

81. Write a function for performing case insensitive string comparison.

82. Write a recursive function to reverse a string using bitwise operator's.

83. Unlike gets (), the function fgets() doesn't delete the newline character entered at the end but retains it. What should we do if we don't want the newline character in the string

84. Write a function to remove all leading and trailing blanks from a string.

85. Input a string and change it so that the characters are placed in alphabetical order. For example the string "Devanshi"should be changed to "aDehinsv",

86. Write a program to abbreviate input text. For example if the input is "World Health Organization'', then the output should be WHO.

87. Write a function to extract a substring from a string. Assume that the substring starts at the i character(start counting from 0 character) and is n characters long.

88. Write a function to replace adjacent multiple spaces in a string by a single space.

89. Write your own pointer versions of the library functions atrophy() , strncmp (), strncat ().

90. Write a function stratr_r() which takes two strings as arguments and returns a pointer to the beginning of the last occurrence of the second string in the first string.

91. Write a function find_indexF () which takes two strings as arguments and returns the index of the first occurrence of the second string in the first string. Write a similar function find_indexF() that returns the index of the last occurrence of the second string in the first string.

92. Write a function str_start() which takes two strings as arguments and returns I if the second string is present at the start of the first string, otherwise returns 0. Write another function str_end () that returns 1 if the second string is present at the end of the first string otherwise returns 0.

93. Write a program to replace all occurrences of the word "Bangalore" by "Bengaluru" in a string.

94. Write a program to find the number of occurrences of a particular word in a string.

95. Write a program to input two strings and remove all the common characters from both the strings. Display the resultant strings.

97. Write a recursive function to count the number of vowels in a string.

98. Write a recursive function to replace each occurrence of a character by another character in a string.

99. Write a recursive function to reverse a string.

100. Write a recursive function to return the index of the first occurrence of a character in a string.

101. Write a recursive function to return the index of the last occurrence of a character in a string.

102. Write a recursive function to find whether a string is palindrome or not. A palindrome is a string that is read the same way forward and backward for example "radar", "hannah", "madam".

103. In the program of the previous problem, make changes so that spaces, punctuation marks, uppercase and lowercase differ- ences are ignored. The strings "A man, a plan, a canal - Panama!", "Live Evil" should be recognized as palindromes.

104. Write a recursive function to convert a string of numbers to an integer. 55. Write a recursive function to print all possible permutations of a string. For example if the string is "abc'' then all possible permutations are abc, acb, bac, bea, cba, cab,

105. Write a function to convert a string into uppercase.

106. Write a function for performing case insensitive string comparison.

107. Unlike gets (), the function fgets() doesn't delete the newline character entered at the end but retains it. What should we do if we don't want the newline character in the string 3-4. Write a function to remove all leading and trailing blanks from a string.

108. Input a string and change it so that the characters are placed in alphabetical order. For example the string "Devanshi ''should be changed to "aDehinsv '', 36. Write a program to abbreviate input text. For example if the input is "World Health Organization'', then the output should be WHO.

109. Write a function to extract a substring from a string. Assume that the substring starts at the i character(start counting from 0 character) and is n characters long.

110. Write a function to replace adjacent multiple spaces in a string by a single space.

111. Write your own pointer versions of the library functions atrophy() , strncmp (), strncat ().

112. Write a function stratr_r() which takes two strings as arguments and returns a pointer to the beginning of the last occurrence of the second string in the first string.

113. Write a function find_indexF () which takes two strings as arguments and returns the index of the first occurrence of the second string in the first string. Write a similar function find_indexF() that returns the index of the last occurrence of the second string in the first string.

114. Write a function str_start() which takes two strings as arguments and returns I if the second string is present at the start of the first string, otherwise returns 0. Write another function str_end () that returns 1 if second string is present at thebend of first string otherwise returns 0.

115. Write a program to replace all occurrences of the word "Bangalore" by "Bengaluru" in a string.

116. Write a program to find the number of occurrences of a particular word in a string.

117. Write a program to input two strings and remove all the common characters from both the strings. Display the resultant strings

118. Write a recursive function to count the number of vowels in a string.

119. Write a recursive function to replace each occurrence of a character by another character in a string.

120. Write a recursive function to reverse a string.

121. Write a recursive function to return the index of the first occurrence of a character in a string.

122. Write a recursive function to return the index of the last occurrence of a character in a string.

123. Write a recursive function to find whether a string is palindrome or not. A palindrome is a string that is read the same way forward and backward for example "radar", "hannah", "madam".

124. In the program of the previous problem, make changes so that spaces, punctuation marks, uppercase and lowercase differ- ences are ignored. The strings "A man, a plan, a canal - Panama!", "Live Evil" should be recognized as palindromes.

125. Write a recursive function to convert a string of numbers to an integer.

126. Write a recursive function to print all possible permutations of a string. For example if the string is "abc'' then all possible permutations are abc, acb, bac, bea, cba, cab.

STRUCTURES & UNIONS

1. What keyword is used to define a structure in C?

a) struct

b) union

c) typedef

d) define

2. Which of the following statements is true about accessing members of a structure?

a) Members are accessed using the dot (.) operator.

b) Members are accessed using the arrow (->) operator.

c) All members must have the same data type.

d) Structures cannot hold members of different data types.

3. What is the size of a structure in memory?

a) The size is always equal to the size of its largest member.

b) The size is the sum of the sizes of all its members.

c) The size depends on the compiler implementation.

d) Structures do not occupy any memory space

4. What is the difference between a structure and a union?

a) Structures cannot hold different data types, while unions can.

b) Unions can only hold one value at a time, while structures can hold multiple values.

c) Structures are used for grouping related data, while unions are used for memory optimization.

d) All of the above.

5. Which of the following statements is NOT valid for initializing a structure?

a) Using a comma-separated list of values inside curly braces.

b) Assigning another structure variable of the same type.

c) Using individual assignment statements for each member.

d) Initializing only some members, leaving others undefined.

6. What is the purpose of the -> operator when accessing members of a structure?

a) It is used to access members of a structure pointer variable.

b) It is used to modify members of a structure by reference.

c) It is an alternative way to access members of any structure.

d) It is not a valid operator for accessing structure members.

7. How can you pass a structure by value to a function in C?

a) By passing the structure name directly.

b) By passing the structure variable using the & (address-of) operator.

c) By creating a pointer to the structure and passing the pointer.

d) Structures cannot be passed by value to functions.

8. What happens when you declare an array of structures?

a) It creates a single structure variable that can hold multiple values.

b) It creates multiple copies of the same structure in memory.

c) It creates a contiguous block of memory to store multiple structures of the same type.

d) It creates a linked list of structures.

9. What is padding in the context of structures?

a) It is the process of adding extra bytes to the size of a structure to improve memory alignment.

b) It is a technique to optimize memory usage by storing smaller data types within larger ones. C) It is a way to dynamically allocate memory for structures at runtime.

d) Padding is not relevant to structures in C.

10. What is the most common use case for unions in C programming?

a) To group related data of different types under a single name.

b) To store multiple values of the same type and access them interchangeably.

c) To improve code readability and maintainability.

d) To dynamically change the data type of a variable during program execution.

11. What is the difference between shallow copy and deep copy when copying structures?

a) Shallow copy copies only the structure itself, while deep copy copies the members recursively.

b) Shallow copy copies the structure by value, while deep copy copies by reference.

c) Shallow copy uses the = operator, while deep copy requires a custom function.

d) There is no difference; both terms refer to the same process.

12. What is the use of self-referential structures in C?

a) To represent hierarchical relationships between data, like a tree structure.

b) To create circular references that can lead to memory leaks.

c) To improve code efficiency by avoiding redundant member declarations.

d) Self-referential structures are not allowed in C.

13. What are bit fields in C?

a) A way to pack multiple small data types (like flags) into a single byte of memory.

b) A mechanism to define custom operators for user-defined data types.

c) A technique to dynamically allocate memory for structures during runtime.

d) Bit fields are not relevant to structures in C.

14. How can you compare two structures for equality in C?

a) By comparing the addresses of the structure variables.

b) By using a built-in compare function for structures.

c) By writing a custom function that compares each member individually.

d) Structures cannot be compared for equality.

15. What is the potential consequence of accessing a union member that hasn't been initialized?

a) The program will crash due to undefined behavior.

b) The value accessed will be the same as the previously accessed member.

c) The value accessed will be garbage or random data.

d) There is no consequence; any member can be accessed regardless of initialization.

16. What is the use of anonymous structures in C?

a) To declare a structure without a name, used only once within a function.

b) To define a structure template that can be used to create multiple instances with different names.

c) To create a structure that cannot be accessed directly and must be used through a pointer.

d) Anonymous structures are not supported in C.

17. How can you define a structure within another structure in C?

a) By using nested structures, where one structure is a member of another.

b) By creating a pointer to the inner structure within the outer structure.

c) By defining both structures separately and using them independently.

d) Structures cannot be members of other structures in C.

18. Create a structure to represent a student with the following members: name (string), roll number (int), and marks (float). Write a function to display the details of a student.

19. Define a union to store either an integer or a floating point number. Write a function to accept the type of data (integer or float) and then read the corresponding value from the user. Store the value in the union and print it.

20. Define a structure to represent a point in 2D space with x and y coordinates (both integers). Write a function to check if two points are equal (have the same x and y coordinates).

21. Create a structure to represent a book with the following members: title (string), author (string), ISBN (long int), and number of pages (int). Write a function to accept details of a book from the user and store them in a structure variable.

22. Define a union to represent the size of a product, which can be specified in centimeters, inches, or feet. Write a function to convert the size from one unit to another (e.g., centimeters to inches).

23. Define a structure to represent a date with day, month, and year (all integers). Write a function to check if a given year is a leap year.

24. Define a structure to represent a complex number with real and imaginary parts (both floats). Write a function to add two complex numbers represented by structures.

25. Implement a linked list using structures. Each node in the list should hold an integer value and a pointer to the next node.

26. Define a self-referential structure to represent a binary tree node. Each node should have data (integer) and pointers to left and right child nodes.

27. Write a program that reads a text file containing lines of student data (name, roll number, marks) and stores the information in an array of structures.

28. Implement a function that takes a structure representing a date (day, month, year) and checks if the date is valid (e.g., not exceeding the number of days in a month).

29. Define a union to represent a shape. The union can hold the dimensions of different shapes like circle (radius), rectangle (length, breadth), or triangle (base, height). Write functions to calculate the area of each shape based on the type stored in the union.

30. Define a structure to represent a playing card with suit (enum) and rank (integer). Write a function to generate a random playing card and another function to print the card's details.

31. Implement a function that takes a structure representing a time (hours, minutes, seconds) and performs basic time arithmetic (e.g., adding two time durations).

32. Create a structure to represent a student enrollment record with student ID (string), course name (string), and grade (character). Write functions to add a new record to an array of structures and to search for a record based on student ID.

33. Analyze the following code snippet and explain the potential issue:

```
struct person {
  char name[50];
  int age;
};
int main() {
  struct person p1;
  strcpy(p1.name, "John Doe");
  p1.age = 30;
  printf("Name: %s, Age: %d\n", p1.name, p1.age);
  return 0;
}
```

34. Consider the following code:

```
union data {
  int i;
  float f;
};
```

```
int main() {
  union data value;
  value.i = 10;
  printf("Float value: %f\n", value.f);
  return 0;
}
```

Explain what happens when the code prints the float value. Why might this be unexpected?

35. Analyze the following code snippet and explain the potential issue:

```
struct student {
  int *marks; // Pointer to an integer
  int num_subjects;
};

int main() {
  struct student s1;
  s1.num_subjects = 5;
  s1.marks = malloc(s1.num_subjects * sizeof(int)); // Allocate memory for marks
  // ... (use s1.marks to store marks)
  free(s1.marks); // Deallocate memory
  return 0;
}
```

Explain what happens if the code doesn't call free(s1.marks) before exiting the function.

36. Analyze the following code snippet and explain the potential issue:

```
struct data {
  int i;
  char str[20];
};

void print_data(struct data *d) {
  printf("Integer: %d\n", d->i);
  printf("String: %s\n", d->str);
```

```
}
int main() {
  struct data d1;
  d1.i = 10;
  strcpy(d1.str, "Hello");
  print_data(&d1);
  return 0;
}
```

Explain what happens if the strcpy function in print_data writes beyond the allocated memory for the str member.

37. Analyze the following code snippet and explain the potential issue:

```
union data {
  int i;
  float f;
};
int main() {
  union data value;
  value.i = 10;
  printf("Float value (before modification): %f\n", value.f);
  value.f = 3.14;
  printf("Float value (after modification): %f\n", value.f);
  return 0;
}
```

Explain what happens to the value of i after modifying the union member f.

38 .what is the size of the structure

```
#include <stdio.h>
typedef struct
{
    char A;
    int B;
    char C;
```

```
} InfoData;
int main(int argc, char *argv[])
{
   //Calculate size of structure
   printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
   return 0;
}
```

what is the size of the structure

```
39.#include <stdio.h>
typedef struct
{
   char A;
   int B;
   char C;
} InfoData;
int main(int argc, char *argv[])
{
   //Calculate size of structure
   printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
   return 0;
}
```

what is the size of the structure

```
40.#include <stdio.h>
typedef struct
{
   double A;
   char B;
   char C;
} InfoData;
int main(int argc, char *argv[])
```

```
{
   //Calculate size of structure
   printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
   return 0;
}
```

41. what is the size of the structure

```
#include <stdio.h>
typedef struct
{
   int A;
   int B;
   char C;
   char D;
   float E;
} InfoData;
int main(int argc, char *argv[])
{
   //Calculate size of structure
   printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
   return 0;
}
```

42. what is the size of the structure

```
#include <stdio.h>
typedef struct
{
   char A;
   char B;
} InfoData;
int main(int argc, char *argv[])
{
   //Calculate size of structure
```

```
    printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
    return 0;
}
```

43. what is the size of the structure.

```
#include <stdio.h>
typedef struct
{
    char A;
    short B;
    int C;
    char D;
} InfoData;
int main(int argc, char *argv[])
{
    //Calculate size of structure
    printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
    return 0;
}
```

44. what is the size of the structure.

```
#include <stdio.h>
typedef struct
{
    char A;
    double B;
    char C;
} InfoData;
int main(int argc, char *argv[])
{
    //Calculate size of structure
    printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
    return 0;
}
```

45. what is the size of the structure

```
#include <stdio.h>
typedef struct
{
   char A;
   char B;
   short C;
   int D;
} InfoData;
int main(int argc, char *argv[])
{
   //Calculate size of structure
   printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
   return 0;
}
```

46. what is the size of the structure

```
#include <stdio.h>
#pragma pack(push, 1)
typedef struct
{
   double A;
   char B;
} InfoData;
#pragma pack(pop)
/* main function */
int main(int argc, char *argv[])
{
   printf("\n Size of Structure = %d\n\n\n\n",sizeof(InfoData));
   return 0;
}
```

```
47. #include <stdio.h>
#pragma pack(push,4)
typedef struct
{
   double A; // 8-byte
   char B; // 1-byte
} InfoData;
#pragma pack(pop)
/* main function */
int main(int argc, char *argv[])
{
   printf("\n Size of Structure = %d\n\n\n\n",sizeof(InfoData));
   return 0;
}
```

DATA STRUCTURES

1. Write a function to count the number of occurrences of an element in a single linked list.

2. Write a function to find the smallest and largest element of a single linked list.

3 . Write a function to create a copy of a single linked list.

4. Write a function to move the largest element to the end of a single linked list.

5. Write a function to move the smallest element to the beginning of a single linked list.

6. Write a program to remove first node of the list and insert it at the end, without changing info part of any node.

7. Write a program to remove the last node of the list and insert it in the beginning, without changing info part of any node.

8. Write recursive functions for the following operations on a linked list.

(i) Find length of the list

(ii) Find sum of all elements in the list

(iii) Display the linked list

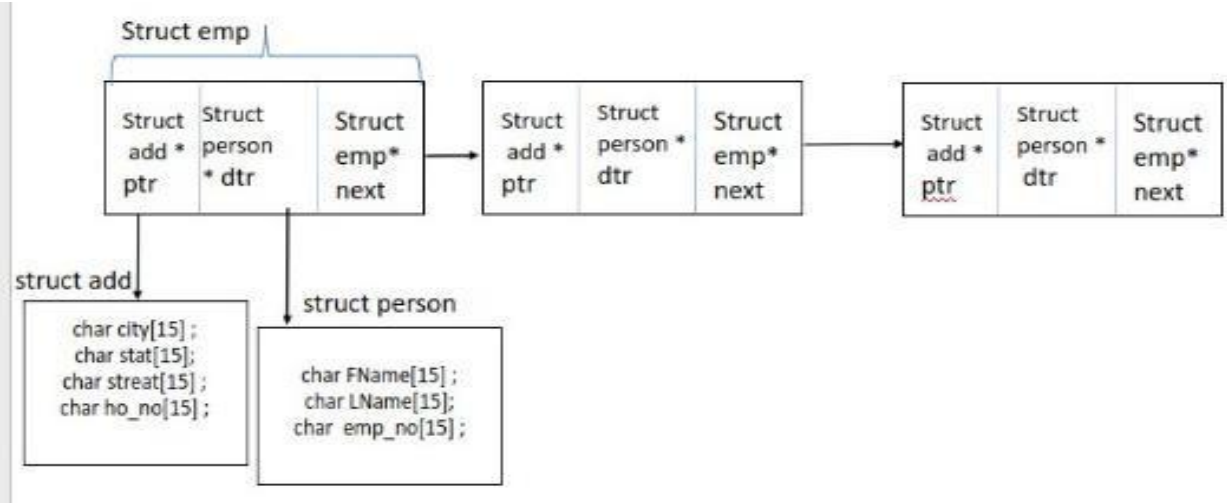
(iv) Display the list in reverse order

(v) Reverse the linked list.

(vi) Insert a node at the end of the list

(vii) Delete the last node from the list

(viii) Search for an element in the list



9. Write a C program to store employee details which contains three structures first structure contains personal details of employee (i.e. first name ,last name ,employee number) second structure contains address (i.e. city ,state , street number and house number) and third structure contains 2 pointer to structure of first and second structure and one self-reference pointer. take reference from above figure

10. Create a linked list to store employee based on above details.

11. Adding an employee node at the beginning of the linked list

12. Adding an employee node at the end of the linked list

13. Adding an employee node at the position of the linked list

14. Delete an employee node at the beginning of the Linked list

15. Delete an employee node at the end of the Linked list

16. Delete an employee node at the position of the Linked list

17. Search an employee node in the linked list

18. Sort an employee node by first name

19. Sort an employee node by last name done pending

20. Sort an employee node by state

21. Sort an employee node by employee number

22. Sort an employee node by city done pending

23. Delete an employee node by first name

24. Delete an employee node by last name

25. Delete an employee node by state

26. Delete an employee node by employee number

27. Delete an employee node by city

28. Create a linked list which contains even employee number

29. Create a linked list which contains odd employee number

30. Create a Single linked list program by performing below operations with a detailed flowchart Insert node at top Insert node at end Insert node at position Delete node at number Display full list. Find node number with value Delete node number with value

31. Create a Single Linked List program and perform below operations with a detailed flowchart Create nodes. Input node data from the user.

-> Print a given node. Determine if the list has a loop.

->Find the total number of nodes.

->Print the full list using the 'print_node' function.

->Swap nodes Sort nodes by data in ascending order

32. Write a program to create Linked list with user nodes with user input data value by performing below operations with a detailed flowchart The head of list should have count of list Create function to split the list to 2 list(one for even data and other for odd data values). Create function to print list by passing head node Create function to print function should also display number of nodes stored in list head

FILES

1. Write a C program to read content from one file and write it to another file.

2. Create a program that reads integers from a file and calculates their sum, then writes the sum to another file.

3. Develop a program to read a text file and count the number of words in it

4. Implement a file management system that allows users to create, read, update, and delete files using C functions.

5. Write a program that checks whether a file exists or not.

6. Create a C program to rename a file.

7. Develop a C program that handles errors while opening files and prints appropriate error messages.

8. Write a program that reads a file line by line and prints each line along with its line number.

9. Create a program to delete a specific line from a text file.

10. Write a C program to store and retrieve student records using binary file handling.

11. Implement a program that copies the contents of one binary file to another.

12. Develop a program that reads a binary file containing integer data and finds the maximum and minimum values.

13. Create a program to open a file and move the file pointer to the end of the file and determine the file size.

14. Write a C program to read the last n lines of a text file.

15. Implement a program to search for a specific string in a text file and print its line number.

16. Develop a C program that appends data to an existing text file.

17. Write a program that appends the current date and time to a log file each time it is run.

18. Create a program that allows users to input text and appends it to a file until they enter "exit".

19. Write a C program to create a random access file that stores records containing information about employees (e.g., name, ID, salary) and allows users to update specific records.

20. Implement a program that reads records from a random access file and displays them based on user input (e.g., display employee details by ID).

21. Develop a C program that demonstrates error handling during file input/output operations, such as handling file not found errors, permission issues, etc.

22. Write a program to read integers from a file and compute their average. Handle cases where the file might contain non-integer data or errors during file reading.

23. Create a program that demonstrates file locking mechanisms in C to prevent concurrent access to a file by multiple processes.

24. Write a program that simulates a simple file-based database system where file locking is used to ensure data consistency during read and write operations.

25. Develop a program that compresses a text file using simple compression techniques (e.g., run-length encoding) and saves the compressed data to another file.

26. Write a program that decompresses a compressed file and restores the original text.

27. Implement a C program that reads data from a CSV (comma-separated values) file, parses it, and performs specific operations (e.g., calculate average, find maximum/minimum).

28. Write a program to extract specific information (e.g., email addresses, phone numbers) from a text file containing unstructured data using regular expressions.

29. Develop a program that retrieves and displays file permissions and ownership details (e.g., owner ID, group ID) for a given file.

30. Implement a program that changes file permissions and ownership based on user input.

31. Write a program in C to create and store information in a text file.

32. Write a program in C to read an existing file

33. Write a program in C to write multiple lines to a text file.

34. Write a program in C to read the file and store the lines in an array.

35. Write a program in C to find the number of lines in a text file.

36. Write a program in C to count the number of words and characters in a file.

37. Write a program in C to delete a specific line from a file.

38. Write a program in C to replace a specific line with another text in a file.

39. Write a program in C to append multiple lines to the end of a text file.

40. Write a program in C to copy a file to another name.

41. Write a program in C to merge two files and write them to another file.

42. Write a program in C to encrypt a text file.

43. Write a program in C to decrypt a previously encrypted file.

44. Write a program in C to remove a file from the disk.

45. A set of strings represent directory paths and a single character directory separator (/). Write a program in C language to get a part of the directory tree that is common to all directories.

46. Write a program in C to read from a file into a variable and write a variable's contents into a file.

47. Write a program in C to display the last modification time of a file

48. Write a program in C language to find the size of a given file.

49. C program to read the name and marks of a number of students and store them in a file.

50. C program to read the name and marks of a number of students from and store them in a file. If the file previously exists, add the information to the file.

51. C program to write all the members of an array of structures to a file using fwrite(). Read the array from the file and display it on the screen.

52. C Program to write to file using fprintf() function

53. C Program to copy data from One file to Another file

54. C Program to read text file using File Handling

55. C Program to create file and store data using file handling

56. C Program to convert File to upper case using file handling

57. C Program to Count characters of file

58. C Program to append data to the file

59. C program to write student record to a file

60. C program read integers and appends sum to the end

61. C program to merge two files using file handling

62. C Program to read student details and store it in file

63. C Program To read text file and print it on screen

64. C program to compare data of two files in File Handling

65. C program to read last n characters of file

66. C Program Delete Line from text File

67. C Program to Replace specific Line in Text File

68. C Program to Find Size of File in File Handling

69. C Program to Create Employee and Update it
70. C Program to Replace First Letter with Capital Letter
71. C Program to Count Lines,Blank Lines,Comments in File
72. C Program to Convert text of File to LowerCase
73. Write a C program that compares two text files and identifies differences between them.
74. Develop a program that merges multiple text files into a single file, preserving the order of lines from each input file.

STORAGE CLASSES

1. Which storage class has the highest scope among all storage classes in C?
   A) auto
   B) register
   C) static
   D) extern

2. What does the "auto" storage class signify in C?
   A) It specifies automatic duration and local scope.
   B) It specifies static duration and file scope.
   C) It specifies external linkage.
   D) It specifies automatic duration and file scope.

3. Variables declared inside a function without any storage class specifier default to which storage class?
   A) auto
   B) static
   C) extern
   D) register

4. Which storage class is preferred for variables that are frequently accessed within a function?
   A) auto
   B) static
   C) extern
   D) register

5. What is the lifetime of a variable declared with the "auto" storage class?
   A) Program runtime
   B) Function execution
   C) Until the next function call
   D) Until the end of the file

6. The "register" storage class is used to suggest to the compiler to store the variable in which memory location?
   A) Disk
   B) RAM
   C) Cache
   D) ROM

7. What is the significance of the "static" storage class for local variables?
   A) It retains the value between function calls.
   B) It makes the variable accessible from other files.
   C) It allocates memory in the heap.
   D) It allows variables to be stored in CPU registers.

8. In C programming, how are static variables initialized by default?
   A) They are initialized to 0.
   B) They are initialized to the value assigned during declaration.
   C) They are initialized to garbage values.
   D) They are not initialized by default.

9. What does the "extern" storage class signify in C?
   A) It specifies automatic duration and local scope.
   B) It specifies static duration and file scope.
   C) It specifies external linkage.
   D) It specifies automatic duration and file scope.

10. Which storage class allows variables to be accessed across different files in C programming?
   A) auto
   B) register
   C) static
   D) extern

11. What is the default initial value of an external variable in C if not explicitly initialized?
   A) 0
   B) 1
   C) Undefined
   D) Null

12. How does the "extern" storage class affect the memory allocation of variables in C?
   A) It allocates memory in the stack.
   B) It allocates memory in the heap.
   C) It does not allocate memory; it only declares the existence of variables.
   D) It allocates memory in the data segment.

13. Which storage class is not applicable to function parameters in C programming?
   A) auto
   B) static
   C) extern
   D) register

14. What is the scope of a variable declared with the "static" storage class inside a function?
   A) Function
   B) File
   C) Program
   D) Block

15. What happens if a variable is declared both "static" and "extern" in C?
   A) It leads to a compilation error.
   B) It depends on the compiler; behavior is undefined.
   C) It becomes inaccessible from other files.
   D) It becomes inaccessible within the same file.

16. Which storage class in C is used to declare constants?
A) auto
B) register
C) static
D) const

17. In C programming, which storage class is recommended for variables that need to be accessed quickly?
A) auto
B) register
C) static
D) extern

18. What is the storage class of a variable declared outside all functions in C if no storage class is specified?
A) auto
B) register
C) static
D) extern

19. What is the default storage class for global variables in C if no storage class is specified?
A) auto
B) register
C) static
D) extern

20. What is the primary use of the "extern" storage class in C?
A) To declare variables without defining them.
B) To allocate memory dynamically.
C) To specify local scope for variables.
D) To register variables for faster access.

21. Which storage class is typically used for variables that need to retain their values across multiple function calls?
A) auto
B) register
C) static
D) extern

22. In C programming, where are variables with the "extern" storage class stored?
A) Stack
B) Heap
C) Data segment
D) Code segment

23. Which storage class in C is used to specify that a variable should be stored in CPU registers if possible?
A) auto
B) register
C) static
D) extern

24. In C programming, what happens if a variable with the "static" storage class is declared within a loop?
A) It retains its value between loop iterations.
B) It loses its value after each loop iteration.
C) It leads to a compilation error.
D) It becomes inaccessible outside the loop.

25. What is the difference between the "auto" and "register" storage classes in C?
A) "auto" variables have automatic duration, while "register" variables have static duration.
B) "auto" variables are stored in CPU registers, while "register" variables are stored in memory.
C) "auto" variables can't be used with pointers, while "register" variables can.
D) "auto" variables can only be used within functions, while "register" variables can be used globally.

26. Which storage class in C is used to specify that a variable should be stored in RAM?
A) auto
B) register
C) static
D) extern

27. What is the default storage class for global variables in C programming?
A) auto
B) register
C) static
D) extern

28. What is the purpose of the "static" storage class for functions in C?
A) It specifies that the function cannot be accessed from other files.
B) It specifies that the function cannot be used with pointers.
C) It specifies that the function should retain its value between function calls.
D) It specifies that the function should have file scope.

29. In C programming, where are variables with the "auto" storage class stored?
A) Stack
B) Heap
C) Data segment
D) Code segment

30. Which storage class in C specifies that a variable should be stored in a register for faster access?
A) auto
B) register
C) static
D) extern

31. What is the scope of a variable declared with the "register" storage class?
A) Local to the function
B) Global to the file
C) Block in which it is declared
D) Throughout the

program

32. In C programming, what is the primary difference between "static" and "extern" storage classes?
A) "static" variables have external linkage, while "extern" variables have internal linkage.
B) "static" variables have file scope, while "extern" variables have function scope.
C) "static" variables have internal linkage, while "extern" variables have external linkage.
D) "static" variables have automatic duration, while "extern" variables have static duration.

33. Which storage class in C is used to specify that a variable should retain its value between function calls?
A) auto
B) register
C) static
D) extern

34. What is the significance of the "auto" storage class for function parameters in C programming?
A) It specifies that the parameters are automatically initialized.
B) It specifies that the parameters are stored in CPU registers.
C) It is redundant; all function parameters have automatic storage by default.
D) It specifies that the parameters have automatic duration and local scope.

35. What happens if a variable with the "register" storage class is declared but the compiler cannot allocate it to a register?
A) It leads to a compilation error.
B) It is automatically promoted to the "static" storage class.
C) It is automatically promoted to the "auto" storage class.
D) It is automatically promoted to the "extern" storage class.

36. Which storage class in C programming is used to specify that a variable should be stored in the CPU cache for faster access?
A) auto
B) register
C) static
D) extern

37. In C programming, how are variables declared with the "static" storage class initialized by default?
A) They are initialized to 0.
B) They are initialized to the value assigned during declaration.
C) They are initialized to garbage values.
D) They are not initialized by default.

38. What is the default storage class for local variables in C programming?
A) auto
B) register
C) static
D) extern

39. Which storage class in C programming is used to specify that a variable should be stored in the data segment?
A) auto
B) register
C) static
D) extern

40. In C programming, which storage class is preferred for variables that need to retain their value between function calls but should not be accessible from other files?
A) auto
B) register
C) static
D) extern

41. What is the default storage class for function parameters in C programming?
A) auto
B) register
C) static
D) extern

42. What happens if a variable with the "static" storage class is declared inside a function?
A) It becomes accessible from other files.
B) It retains its value between function calls.
C) It loses its value after each function call.
D) It leads to a compilation error.

43. Which storage class in C programming is used to specify that a variable should be stored in ROM?
A) auto
B) register
C) static
D) extern

44. In C programming, what is the significance of the "auto" storage class for global variables?
A) It specifies that the variables are stored in CPU registers.
B) It specifies that the variables have automatic duration and local scope.
C) It specifies that the variables are stored in the data segment.
D) It specifies that the variables are accessible only within the file where they are declared.

45. What is the storage class of a variable declared within a block in C programming if no storage class is specified?
A) auto
B) register
C) static
D) extern

46. Which storage class in C programming is used to specify that a variable should be stored in the stack?
A) auto
B) register
C) static
D) extern

47. What is the scope of a variable declared with the "static" storage class outside of all functions in C programming?
A) Function
B) File
C) Program
D) Block

48. What is the significance of the "extern" storage class for function parameters in C programming?
A) It specifies that the parameters are automatically initialized.
B) It specifies that the parameters are stored in CPU registers.
C) It is redundant; all function parameters have automatic storage by default.
D) It specifies that the parameters have external linkage.

49. In C programming, what is the default storage class for variables declared inside a function?
A) auto
B) register
C) static
D) extern

50. Which storage class in C programming is used to specify that a variable should be stored in the code segment?
A) auto
B) register
C) static
D) extern

51. What is the default storage class for function declarations in C programming?
A) auto
B) register
C) static
D) extern

52. What is the significance of the "static" storage class for function parameters in C programming?
A) It specifies that the parameters are automatically initialized.
B) It specifies that the parameters are stored in CPU registers.
C) It is redundant; all function parameters have automatic storage by default.
D) It specifies that the parameters retain their value between function calls.

53. In C programming, where are variables with the "static" storage class stored?
A) Stack
B) Heap
C) Data segment
D) Code segment

54. Which storage class in C programming is used to specify that a variable should be stored in the data segment?
A) auto
B) register
C) static
D) extern

55. What is the scope of a variable declared with the "register" storage class?
A) Local to the function
B) Global to the file
C) Block in which it is declared
D) Throughout the program

56. In C programming, what is the primary difference between "static" and "extern" storage classes?
A) "static" variables have external linkage, while "extern" variables have internal linkage.
B) "static" variables have file scope, while "extern" variables have function scope.
C) "static" variables have internal linkage, while "extern" variables have external linkage.
D) "static" variables have automatic duration, while "extern" variables have static duration.

57. Which storage class in C programming is used to specify that a variable should be stored in the CPU cache for faster access?
A) auto
B) register
C) static
D) extern

58. In C programming, how are variables declared with the "static" storage class initialized by default?
A) They are initialized to 0.
B) They are initialized to the value assigned during declaration.
C) They are initialized to garbage values.
D) They are not initialized by default.

59. What is the default storage class for local variables in C programming?
A) auto
B) register
C) static
D) extern

60. Which storage class in C programming is used to specify that a variable should be stored in the data segment?
A) auto
B) register
C) static
D) extern

61. In C programming, which storage class is preferred for variables that need to retain their value between function calls but should not be accessible from other files?
A) auto
B) register
C) static
D) extern

62. What is the default storage class for function parameters in C programming?
A) auto
B) register
C) static
D) extern

63. What happens if a variable with the "static" storage class is declared inside a function?
A) It becomes accessible from other files.
B) It retains its value between function calls.
C) It loses its value after each function call.
D) It leads to a compilation error.

64. Which storage class in C programming is used to specify that a variable should be stored in ROM?
A) auto
B) register
C) static
D) extern

65. In C programming, what is the significance of the "auto" storage class for global variables?
A) It specifies that the variables are stored in CPU registers.
B) It specifies that the variables have automatic duration and local scope.
C) It specifies that the variables are stored in the data segment.
D) It specifies that the variables are accessible only within the file where they are declared.

66. What is the storage class of a variable declared within a block in C programming if no storage class is specified?
A) auto
B) register
C) static
D) extern

67. Which storage class in C programming is used to specify that a variable should be stored in the stack?
A) auto
B) register
C) static
D) extern

68. What is the scope of a variable declared with the "static" storage class outside of all functions in C programming?
A) Function
B) File
C) Program
D) Block

69. What is the significance of the "extern" storage class for function parameters in C programming?
A) It specifies that the parameters are automatically initialized.
B) It specifies that the parameters are stored in CPU registers.
C) It is redundant; all function parameters have automatic storage by default.
D) It specifies that the parameters have external linkage.

70. In C programming, what is the default storage class for variables declared inside a function?
A) auto
B) register
C) static
D) extern

71. Which storage class in C programming is used to specify that a variable should be stored in the code segment?
A) auto
B) register
C) static
D) extern

72. What is the default storage class for function declarations in C programming?
A) auto
B) register
C) static
D) extern

73. What is the significance of the "static" storage class for function parameters in C programming?
A) It specifies that the parameters are automatically initialized.
B) It specifies that the parameters are stored in CPU registers.
C) It is redundant; all function parameters have automatic storage by default.
D) It specifies that the parameters retain their value between function calls.

74. In C programming, where are variables with the "static" storage class stored?
A) Stack
B) Heap
C) Data segment
D) Code segment

75. Which storage class in C programming is used to specify that a variable should be stored in the data segment?
A) auto
B) register
C) static
D) extern

76. What is the scope of a variable declared with the "register" storage class?
A) Local to the function
B) Global to the file
C) Block in which it is declared
D) Throughout the program

77. In C programming, what is the primary difference between "static" and "extern" storage classes?
A) "static" variables have external linkage, while "extern" variables have internal linkage.
B) "static" variables have file scope, while "extern" variables have function scope.
C) "static" variables have internal linkage, while "extern" variables have external linkage.
D) "static" variables have automatic duration, while "extern" variables have static duration.

78. Which storage class in C programming is used to specify that a variable should be stored in the CPU cache for faster access?
A) auto
B) register
C) static
D) extern

79. In C programming, how are variables declared with the "static" storage class initialized by default?

A) They are initialized to 0.
B) They are initialized to the value assigned during declaration.
C) They are initialized to garbage values.
D) They are not initialized by default.

80. What is the default storage class for local variables in C programming?
A) auto
B) register
C) static
D) extern

81. Which storage class in C programming is used to specify that a variable should be stored in the data segment?
A) auto
B) register
C) static
D) extern

82. In C programming, which storage class is preferred for variables that need to retain their value between function calls but should not be accessible from other files?
A) auto
B) register
C) static
D) extern

83. What is the default storage class for function parameters in C programming?
A) auto
B) register
C) static
D) extern

84. What happens if a variable with the "static" storage class is declared inside a function?
A) It becomes accessible from other files.
B) It retains its value between function calls.
C) It loses its value after each function call.
D) It leads to a compilation error.

85. Which storage class in C programming is used to specify that a variable should be stored in ROM?
A) auto
B) register
C) static
D) extern

86. In C programming, what is the significance of the "auto" storage class for global variables?

A) It specifies that the variables are stored in CPU registers.
B) It specifies that the variables have automatic duration and local scope.
C) It specifies that the variables are stored in the data segment.
D) It specifies that the variables are accessible only within the file where they are declared.

87. What is the storage class of a variable declared within a block in C programming if no storage class is specified?
A) auto
B) register
C) static
D) extern

88. Which storage class in C programming is used to specify that a variable should be stored in the stack?
A) auto
B) register
C) static
D) extern

89. What is the scope of a variable declared with the "static" storage class outside of all functions in C programming?

A) Function
B) File
C) Program
D) Block

90. What is the significance of the "extern" storage class for function parameters in C programming?
A) It specifies that the parameters are automatically initialized.
B) It specifies that the parameters are stored in CPU registers.
C) It is redundant; all function parameters have automatic storage by default.
D) It specifies that the parameters have external linkage.

91. In C programming, what is the default storage class for variables declared inside a function?
A) auto
B) register
C) static
D) extern

92. Which storage class in C programming is used to specify that a variable should be stored in the code segment?
A) auto
B) register
C) static
D) extern

93. What is the default storage class for function declarations in C programming?
A) auto
B) register
C) static
D) extern

94. What is the significance of the "static" storage class for function parameters in C programming?
A) It specifies that the parameters are automatically initialized.
B) It specifies that the parameters are stored in CPU registers.
C) It is redundant; all function parameters have automatic storage by default.
D) It specifies that the parameters retain their value between function calls.

95. In C programming, where are variables with the "static" storage class stored?
A) Stack
B) Heap
C) Data segment
D) Code segment

96. Which storage class in C programming is used to specify that a variable should be stored in the data segment?
A) auto
B) register
C) static
D) extern

97. What is the scope of a variable declared with the "register" storage class?
A) Local to the function
B) Global to the file
C) Block in which it is declared
D) Throughout the program

98. In C programming, what is the primary difference between "static" and "extern" storage classes?
A) "static" variables have external linkage, while "extern" variables have internal linkage.
B) "static" variables have file scope, while "extern" variables have function scope.
C) "static" variables have internal linkage, while "extern" variables have external linkage.
D) "static" variables have automatic duration, while "extern" variables have static duration.

99. Which storage class in C programming is used to specify that a variable should be stored in the CPU cache for faster access?
A) auto
B) register
C) static
D) extern

100. In C programming, how are variables declared with the "static" storage class initialized by default?

A) They are initialized to 0.
B) They are initialized to the value assigned during declaration.
C) They are initialized to garbage values.
D) They are not initialized by default.

101. What is the default storage class for local variables in C programming?
A) auto
B) register
C) static
D) extern

102. Which storage class in C programming is used to specify that a variable should be stored in the data segment?
A) auto
B) register
C) static
D) extern

103. In C programming, which storage class is preferred for variables that need to retain their value between function calls but should not be accessible from other files?
A) auto
B) register
C) static
D) extern

104. What is the default storage class for function parameters in C programming?
A) auto
B) register
C) static
D) extern

105. What happens if a variable with the "static" storage class is declared inside a function?
A) It becomes accessible from other files.
B) It retains its value between function calls.
C) It loses its value after each function call.
D) It leads to a compilation error.

106. Which storage class in C programming is used to specify that a variable should be stored in ROM?
A) auto
B) register
C) static
D) extern

107. In C programming, what is the significance of the "auto" storage class for global variables?

A) It specifies that the variables are stored in CPU registers.
B) It specifies that the variables have automatic duration and local scope.
C) It specifies that the variables are stored in the data segment.
D) It specifies that the variables are accessible only within the file where they are declared.

108. What is the storage class of a variable declared within a block in C programming if no storage class is specified?
A) auto
B) register
C) static
D) extern

109. Which storage class in C programming is used to specify that a variable should be stored in the stack?
A) auto
B) register
C) static
D) extern

110. What is the scope of a variable declared with the "static" storage class outside of all functions in C programming?
A) Function
B) File
C) Program
D) Block

111. What is the significance of the "extern" storage class for function parameters in C programming?
A) It specifies that the parameters are automatically initialized.
B) It specifies that the parameters are stored in CPU registers.
C) It is redundant; all function parameters have automatic storage by default.
D) It specifies that the parameters have external linkage.

112. In C programming, what is the default storage class for variables declared inside a function?
A) auto
B) register
C) static
D) extern

113. Which storage class in C programming is used to specify that a variable should be stored in the code segment?
A) auto
B) register
C) static
D) extern

114. What is the default storage class for function declarations in C programming?

A) auto
B) register
C) static
D) extern

115. What is the significance of the "static" storage class for function parameters in C programming?
A) It specifies that the parameters are automatically initialized.
B) It specifies that the parameters are stored in CPU registers.
C) It is redundant; all function parameters have automatic storage by default.
D) It specifies that the parameters retain their value between function calls.

116. In C programming, where are variables with the "static" storage class stored?
A) Stack
B) Heap
C) Data segment
D) Code segment

117. Which storage class in C programming is used to specify that a variable should be stored in the data segment?
A) auto
B) register
C) static
D) extern

118. What is the scope of a variable declared with the "register" storage class?
A) Local to the function
B) Global to the file
C) Block in which it is declared
D) Throughout the program

BITWISE OPERARTOR

1.) Check if the ith bit is set or not
2.) Set the ith bit of a number.
3.) clear the ith bit of a number.
4.) Remove the last set bit of a number.
5.) Find whether a number is even or odd
6.) Check if the number is a power of 2?
7.) Check if a number is a power of 4?
8.) Check if a number is a power of 8?
9.) Check if a number is a power of 16?
10.) Toggle ith Bit of a number?
11.) Count the number of set bits in a number
12.) Find the two non-repeating elements in an array of repeating elements/ Unique Numbers 2
13.) Convert Uppercase to LowerCase:
14.). Convert Lowercase to Uppercase
15.). Invert Alphabet's Case
16.). Find Letter Position in alphabet
17.) Given a set of numbers where all elements occur an even number of times except one number, find the odd occurring number.
18.) Swap two numbers using Bit manipulation:
19.) Calculate XOR from 1 to n
20.) Find XOR of numbers from the range [L,R]
21.) Check whether the number is even or not
22.) Find the XOR of the XOR of all subsets of an array:
23.) Count Number of bits to be flipped to convert A to B:
24.) Find missing number in an array:
25.) Print the binary representation of decimal number:
26.) Reverse the bits of a number:
27.) Swap the ith and Jth bit.
28.) Swap all even and odd bits
29.) Copy set bits in a range, toggle set bits in a range:
30.) Divide two integers without using Multiplication, Division and mod operator:
31.) One unique rest thrice
32.) Reduce a Number to 1
33.) Detect if two integers have opposite sign
34.) Add 1 to an integer
35.) Find Xor of a number without using XOR operator
36.) Determine if two integers are equal without using comparison and arithmetic operators
37.) Find minimum or maximum of two integers without using branching
38.) Find missing and repeating number / Set mismatch:
39.) Maximum Product of Word Lengths
40.) Check if a String contains all binary codes of size k
41.) Find the Duplicate Number