

Multiple choice questions:

- | | |
|---|---|
| <p>1. A whole number with a decimal point is known as:</p> <ol style="list-style-type: none"> floating point number character integer none <p>2. The declaration <code>unsigned u</code> indicates:</p> <ol style="list-style-type: none"> <code>u</code> is a character <code>u</code> is an unsigned integer <code>u</code> is unsigned character <code>u</code> is unsigned long integer <p>3. Which statement must not end with semicolon:</p> <ol style="list-style-type: none"> <code>#define</code> variable declaration assignment none <p>4. Point out the valid variable names:</p> <ol style="list-style-type: none"> gross salary gross-salary AVG AVG. <p>5. If <code>a</code> is an integer variable, <code>a = 5/2</code> will return a value.</p> <ol style="list-style-type: none"> 2.5 3 2 0 <p>6. The expression, <code>a=7/22*(3.14+2)*3/5</code> is evaluated to</p> <ol style="list-style-type: none"> 8.28 6.28 3.14 0 | <p>7. The expression <code>a=30*1000+2768</code> evaluates to</p> <ol style="list-style-type: none"> 32768 -32768 113040 0 <p>8. The expression <code>a=4+2%-8</code> evaluates to</p> <ol style="list-style-type: none"> -6 6 4 none of the above <p>9. Hierarchy decides which operator</p> <ol style="list-style-type: none"> is most important is used first is fastest operates on largest numbers <p>10. In C a variable can not contain</p> <ol style="list-style-type: none"> blank spaces hyphen (-) decimal point all the above <p>11. Which of the following is odd one out</p> <ol style="list-style-type: none"> + - / ** <p>12. What will be the value of <code>d</code> assuming it to be float after the operation <code>d=2/7.0</code></p> <ol style="list-style-type: none"> 0 0.2857 can not be determined none of the above |
|---|---|

Matrix Computers

Answers:-

1	(a)	2	(b)	3	(a)	4	(c)	5	(c)	6	(d)
7	(b)	8	(b)	9	(b)	10	(d)	11	(d)	12	(b)

Give the output of the following:-

1. `printf("%d\n%d\n%d\n%d", 72,072,0x72,0X72);`

2. `printf("%d\n%o\n%x", 72,72,72);`

3. `printf("%#d\n%#o\n %#x", 72,72,72);`

4. `char ch = 291;
printf ("%d\n%d\n%c", 32770,ch,ch);`

5. `printf ("%d\n%c\n");
printf ("%d\n%c\n");`

6. `int a =33000;
float b= 3.4e100;
printf ("a= %d\nb=%f\n", a,b);`

7. `int a,b;
a=-3-3;
b=-3-(-3);
printf("a=%d\nb=%d", a,b);`

8. `int x;
x=3+4-7*8/5%10;
printf("x=%d",x);`

9. `int x;
x=4%5+6%5;
printf("x=%d",x);`

10. `int x;
x=-3*4%-6/-5;
printf("x=%d",x);`

11. `float a = 5, b = 2 ;
int c ;
c = a % b ;
printf ("%d", c) ;`

12.

`int x ;
x = 3**4-7^8 ;
printf ("x = %d", x) ;`

13.

`int g = 300*300/300 ;
printf ("g = %d", g) ;`

14.

`float a = 1.5 ;
int b = 3 ;
a = b/2 + b*8/b-b + a/3 ;
printf ("a = %f", a) ;`

15.

`int i = 3, a = 4, n ;
float t = 4.2 ;
n = a*a/i + i/2*t + 2 + t ;
printf ("n = %d", n) ;`

16.

`int q = 2, d = 3, st ;
st = q*d/4-12/12 + 12/3*16/d ;
printf ("st = %d", st) ;`

17.

`int a, b ;
a = 5.999999 ;
b = 5.000001 ;
printf ("a = %d b = %d", a, b) ;`

18.

`float a ;
a = 4/2 ;
printf ("%f %f", a, 4/2) ;`

19.

`printf ("%d %f\n", 4, 4) ;
printf ("%d %f\n", 4.0, 4.0) ;`

20.

`float a = 4 ;
int i = 2 ;
printf ("%f %d", i/a, i/a) ;
printf ("%d %f", i/a, i/a) ;`

M@trix Computers

21.

```
printf ("%d", sizeof(4)/sizeof(2.0));
printf ("%d", sizeof(2.0)/sizeof(4));
```

22.

```
printf ( "nn\n\n nn\n");
printf ( "nn/n/n nn/n" );
```

23.

```
/*this program attempts to find what happens
when integer range /*-32768 to +32767*/ is
exceeded*/
int a = 330000 ;
float b = 3.4e 100 ;
printf ("a = %d b = %f\n", a, b) ;
```

24.

printf ("menu is a list of options you have at a particular point in a program. it is just like a restaurant menu - everything has a misleading name and what you want is never available.");

25.

```
int i = 4, j = -1, k = 0, w, x, y, z;
w = i||j||k;
x = i&&j&&k;
y = i|| j&&k;
z = i&&j||k;
printf("\nw=%d x=%d y=%d z=%d", w,x,y,z);
```

26.

```
int i = 4, j = -1, k = 0, y, z;
y = j+5&& j+1|| k+2;
z = i+5 || j+1 && k+2;
printf("\n y=%d z=%d",y,z);
```

27.

We want to round off x, a float, to an int value. The correct way to do so would be:

- a. y=(int) (x+0.5);
- b. y=int (x+0.5);
- c. y=(int) x+0.5;
- d. y=(int) ((int) x+0.5);

28.

What should we do to treat the constant 3.14 as a long double?

29.

What will be the output of the following statement:

```
printf("%d %d %d", sizeof(3.14f), sizeof(3.14),
sizeof(3.14l));
a. 4 4 4
```

b. 4 garbage value garbage value

c. 4 8 10

d. Error

30.

What should we do to treat the constant 3.14 as a float?

31.

```
int i = -1, j = 1, k, l;
k = i&&j;
l = i||j;
printf("%d %d", i, j);
```

32.

```
int j = 4, k;
k = !5 && j;
printf("k = %d", k);
```

33.

```
int i=3;
i=i++;
printf("%d", i);
```

34.

```
int i=2;
printf("%d %d", ++i, ++i);
```

- a. 3 4
- b. 4 3
- c. 4 4
- d. Output may vary from compiler to compiler.

35.

```
int a = 3, b = 4 ;
b % = 3 + 4 ;
a * = a + 5 ;
printf ("b = %d a = %d", b, a) ;
```

36.

```
int x = 3 ;
x *= x + 4 ;
printf ("x = %d", x) ;
```

37.

```
int x = 3, y, z ;
z = y = x ;
z * = y = x*x ;
printf ("x = %d y = %d z = %d", x, y, z) ;
```

38.

```
int x = 3, y, z ;
z = y = x ;
```

M@trix Computers

$z^* = y/x$;
`printf("x = %d y = %d z = %d", x, y, z);`

39.
`int x = 1, y = 5;`
 $y^* = x$;
`printf("x = %d y = %d", x, y);`

40.
`int x = 3, y, z;`
 $z = y = x$;
 $z = y + x = -z$;
`printf("x = %d y = %d z = %d", x, y, z);`

41.
`int x = 5;`
 $x++;$
`printf("x=%d\n", x);`
 $++x;$
`printf("x=%d\n", x);`

42.
`int x=3,z;`
 $x=x++ + 10;$
`printf("x=%d z=%d",x,z);`

43.
`int x=3,z;`
 $x=++x + 10;$
`printf("x=%d z=%d",x,z);`

44.
`int x=3,z;`
 $z=x--111;$
`printf("x=%d z=%d",x,z);`

45.
`int x=3,z;`
 $z=x--1;$
`printf("x=%d z=%d",x,z);`

46.
`int x=3,z;`
 $z=x++ + x++;$
`printf("x=%d z=%d",x,z);`

47.
`int x=3,z;`
 $z=x++ + ++x;$
`printf("x=%d z=%d",x,z);`

48.
`int x=3,z;`

$z=x--1$
`printf("x=%d z=%d",x,z);`

49.
`int x=3,z;`
 $z=x/++x;$
`printf("x=%d z=%d",x,z);`

50.
`int i=3,j;`
 $j=++i * ++i * ++i;$
`printf("%d",j);`

51.
`int x=3,y=3,z=3;`
 $z= -x-- -y;$
`printf("x=%d y=%d z=%d",x,y,z);`

52.
`int x=10,y=x,z=x;`
 $y= x;$
 $z = -x;$
 $x= -x$
`printf("y=%d z=%d x=%d",y,z,x);`

53.
`int x=10,y,z;`
 $z=y=x;$
 $y =x--;$
 $z =--x;$
 $x = --x - x--;$
`printf("y=%d z=%d x=%d",y,z,x);`

54.
`int x=4,y=3,z;`
 $z = x--y;$
`printf("x=%d y=%d z=%d",x,y,z);`

55.
`int x,y,z;`
 $x=y=z=1;$
 $z= ++x || ++y \&& ++z;$
`printf("x=%d y=%d z=%d\n",x,y,z);`

56.
`int x,y,z;`
 $x=y=z=1;$
 $z= ++x \&& ++y || ++z;$
`printf("x=%d y=%d z=%d\n",x,y,z);`

57.
`int x,y,z;`
 $x=y=z=1;$
 $z= ++x \&& ++y \&& ++z;$

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

58.

```
int x,y,z;
x=y=z=-1;
z= ++x && ++y || ++z;
printf("x=%d y=%d z=%d\n",x,y,z);
```

59.

```
int x,y,z;
x=y=z=-1;
z= ++x || ++y && ++z;
printf("x=%d y=%d z=%d\n",x,y,z);
```

60.

```
int x,y,z;
x=y=z=-1;
z= ++x && ++y && ++z;
printf("x=%d y=%d z=%d\n",x,y,z);
```

61.

```
int x=10,y;
y = -x--;
```

Output:

```
1.    72      58      114     114
2.    72      110      48
3.    72      0110     0x48
4.   -32766      35      #
5.   -22      A (Garbage)
       -22      A (Garbage)
6.   a = -32536    b = +inf
7.   a = 0    b = -6
8.   x = 6
9.   x = 5
10.  x = 0
11:  error message: illegal use of floating
     point in function main
12:  error message: invalid indirection
13:  g = 81
14:  6.500000
15:  n = 15
16:  st = 21
17:  a = 5    b = 5
18:  2.000000   -1.4746701e+308
19:  4      512.000001
     0      0.000000
20:  0.500000      0
     0      0.000000
21:  0      4
22:  nn
     nn
     nn/n/n nn/n
```

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

62.

```
int i=2;
printf("i-- = %d",i--);
```

63. What would be the output of the following program:
`printf("%%%%");`

64. Would the following code work, if yes, what would be the output:
`int n=5;
printf("n=%*d",n,n);`

65. Point out the error if any in the following program:

```
char ch;
int i;
scanf("%c", &i);
scanf("%d", &ch);
printf("%c %d",ch, i);
```

error message: expression syntax in function main (nested comments are not allowed.)

23:

menu is a list of options you have at a particular point in a program. it is just like a restaurant menu – everything has a misleading name and what you want is never available.

24:

`w=1 x=0 y=1 z=1`

25:

`y=1`

26:

`z=1`

27:

`a`

28:

`Use 3.14!`

29:

`c`

30:

`Use 3.14f`

31:

`-1 1`

32:

`0`

33:

`4`

34:

`b`

35:

`b = 4 a = 24`

36:

`x = 21`

37:

`x = 3 y = 9 z = 27`

38:

`x = 3 y = 1 z = 3`

39:

error message: expression syntax in function main

40:

`x = -3 y = 0 z = 0`

41:

`x = 6`

`x = 7`

42:

`x = 4 z = 13`

```

43: x = 4 z = 14
44: x = 2 z = -108
45: x=2 z=4
46: x=5 z=6
47: x=5 z=8
48: Error Message:Lvalue required in
    function main
49: x=4 z=1
50: 216
51: x=2 y=2 z=8
52: y=0 z=-10 x=-10
53: y=0 z=2 x=6
54: x=3 y=3 z=1
55: x=2 y=1 z=1
56: x=2 y=2 z=1
57: x=2 y=2 z=1
58: x=0 y=-1 z=0
59: x=0 y=0 z=0
60: x=0 y=-1 z=0

```

- 61: Lvalue required in function main
 62: i-- = 2
 63: %
 64: Yes. n=_____ 5
 * indicates that an *int* value from the argument list will be used for field width. In the argument list the width precedes the value to be printed. In this case the format specifier becomes *%5d*
 65. You will not get a chance to input a character for the second *scanf()* statement. Solution to this problem is to precede the second *scanf()* with the following statement.
fflush(stdin);
 This would flush the enter hit for the previous *scanf()* to be flushed out from the input stream, i.e. keyboard.

Theory Questions:-

- Explain what are data types? Name all the data types in c language with their size in bytes (in dos operating system). List all the format specifiers. Also write the formula by which we can calculate the range of a given data type. Calculate the range of int by this.
- How many keywords are present in C. Give the name of all the keywords?
- Explain all the naming rules of a variable in C.
- Characteristics of a program.
- Name all the escape sequences.
- What do you mean by precedence of operators? List all the available operators in c in the order of their precedence. (higher to lower)
- What do you mean by typecasting? Explain implicit and explicit typecasting

Some Solved Programs:

- 2.1 If a four digit number is input through the keyboard, write a program to obtain the sum of the first and last digit of the number.

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     int n,a,b,sum;
6.     clrscr();
7.     printf("Enter a four digit number: ");
8.     scanf("%d",&n);
9.     a=n%10;
10.    b=n/1000;
11.    sum = a + b;
12.    printf("Sum of the first and last digit is %d", sum);
13.    getch();
14. }

```

M@trix Computers

- 2.2 WAP to print the range of a number. E.g. number 78 is between 70 and 79, 102 is between 100 and 109.**

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     int n,lower_r,upper_r;
6.     clrscr();
7.     printf("Enter a number: ");
8.     scanf("%d",&n);
9.     lower_r = n % 10 * 10; /*the arithmetic operators work from left to right*/
10.    upper_r = lower_r + 10;
11.    printf("Range is %d - %d", lower_r,upper_r);
12.    getch();
13. }
```

- 2.3 WAP to print the various denominations of a given rupee. E.g. if a person has 1779 in his pocket the program should print the following.**

$500 \times 3 = 1500, 100 \times 2 = 200, 50 \times 1 = 50, 20 \times 1 = 20, 10 \times 0 = 0, 5 \times 1 = 5, 2 \times 2 = 4$

$1 \times 0 = 0$

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     int Rs,a,b,c,d,e,f,g,h;
6.     clrscr();
7.     printf("Enter the amount: ");
8.     scanf("%d",&Rs);
9.     a= Rs/500;
10.    Rs = Rs%500;
11.    b = Rs/100;
12.    Rs=Rs%100;
13.    c=Rs/50;
14.    Rs=Rs%50;
15.    d=Rs/20;
16.    Rs=Rs%20;
17.    e=Rs/10;
18.    Rs=Rs%10;
19.    f=Rs/5;
20.    Rs=Rs%5;
21.    g=Rs/2;
22.    Rs=Rs%2;
23.    h=Rs/1;
24.    printf("The various denominations of the given rupees are\n");
25.    printf("500X%d\n100X%d\n50X%d\n20X%d\n10X%d\n5X%d\n2X%d\n1X%d",a,b,c,d,e,f,g,h);
26.    getch();
27. }
```

}

Some Solved Problems:-

To calculate the real roots of the quadratic equation

3.1

```
1. #include<stdio.h>
2. #include<conio.h>
3. #include<math.h>      /*to use function sqrt*/
4. main()
5. {
6.     float a,b,c,d,r1,r2;
7.     clrscr();
8.     printf("enter three numbers: ");
9.     scanf("%f %f %f",&a,&b,&c);
10.    d=b*b-(4*a*c);
11.    if(d<0)
12.        printf("roots are imaginary");
13.    else
```

M(a)trix Computers

```

14.     {
15.         r1=(-b+sqrt(d))/(2*a); /*to calculate square root */
16.         r2=(-b-sqrt(d))/(2*a);
17.         printf("The roots are %f %f",r1,r2);
18.     }
19.     getch();
20. }
```

- 3.2 If cost price and selling price of an item is input through the keyboard, WAP to determine whether the seller has made profit or incurred loss. Also determine how much profit he made or loss be incurred.

```

1 #include<stdio.h>
2 #include<conio.h>
3 main()
4 {
5     float cp,sp;
6     clrscr();
7     printf("Enter the cost price and the selling price of the item: ");
8     scanf("%f %f",&cp,&sp);
9     if(sp>cp)
10        printf("profit is Rs. %.2f",sp-cp); /*%.2f is used to print number up to 2
11 decimal places only*/
12    else if(cp > sp)
13        printf("loss is Rs %.2f",cp-sp);
14    else
15        printf("No Profit No Loss");
16    getch();
17 }
```

- 3.3 WAP to create a calculator which can do addition, subtraction, multiplication, division, & modulus (remainder).

```

1 #include<stdio.h>
2 #include<conio.h>
3 main()
4 {
5     int a,b,ch;
6     clrscr();
7     printf("Enter two nos. ");
8     scanf("%d%d",&a,&b);
9     printf("1:\tAdd\n2:\tSubtract\n3:\tMultiply\n4:\tDivide\n5:\tModulus\nEnter your choice: ");
10    scanf("%d",&ch);
11    switch(ch)
12    {
13        case 1:
14            printf("%d\n",a+b);
15            break;
16        case 2:
17            printf("%d\n",a-b);
18            break;
19        case 3:
20            printf("%d\n",a*b);
21            break;
22        case 4:
```

```

23.         printf("%f\n", (float)a/b);
24.         break;
25.     case 5:
26.         printf("%d\n", a%b);
27.         break;
28.     default:
29.         printf("Invalid Input");
30.     }
31.     getch();
32. }
```

What will be the output of the following program:-

1.

```

int i = 4, z = 12;
if( i == 5 || z > 50)
    printf("\nwelcome to matrix");
else
    printf("\n you may go now");
```

2.

```

float a = 0.7;
if(a < 0.7)
    printf("Hello");
else
    printf("Bye");
```

3.

```

int i = -3, j = 3;
if (!i + !j * 1)
    printf("Hello");
else
    printf("Bye");
```

4.

```

int a = 40;
if(a > 40 && a < 45)
    printf("a is between 40 and 45");
else
    printf("%d", a);
```

5.

Rewrite the following set of statements using conditional operators:

```

int a=1,b;
if(a>10)
    b=20;
```

6.

```

int i = 4, j, num;
j = (num < 0 ? 0 : num * num);
printf("%d", j);
```

7.

```

int a, n = 30;
a = (n > 5 ? (n <= 10 ? 100 : 200) : 500);
printf("%d", n);
```

8.

```

int k = 4;
(!k != 1 ? printf("\nHello") : printf("Bye"));
```

9.

```

int p = 8, q = 20;
if(p == 5 && q < 5)
    printf("Hello Matrix");
else
    printf("Bye Matrix");
```

10.

```

int i = 4;
switch (i)
{
    default:
        printf("Matrix");
    case 1:
        printf("Computer");
        break;
    case 2:
        printf("Education");
        break;
    case 3:
        printf("Hello");
}
```

11.

```

int i = 4, j = 4;
switch (i)
{
    case 1:
        printf("Hello");
        break;
    case j:
```

```

        printf("Bye");
        break;
    }
}

```

12.

```

int i =1;
switch (i)
{
    case 1:
        printf("Hello");
        break;
    case 1 * 2 + 4:
        printf("Bye");
        break;
}

```

13.

```

int i =4;
switch (i)
{
}
printf("Hello World");

```

14.

```

int i =1;
switch (i)
{
    printf("Matrix");/*common for both
                      cases*/
    case 1:
        printf("Hello");
        break;
    case 2:
        printf("Bye");
        break;
}

```

15.
Can we use switch statement to switch on string?

16.
We want to test whether a value lies in the range 2 to 4 or 5 to 7. Can we do this using switch?

17.

```

switch(a)
{
    case 2:
    case 3:
    case 4:

```

```

/*some statements*/
break;
case 5
case 6:
case 7:
/*some other statements*/
break;
}

```

The way *break* is used to take the control of switch can *continue* be used to take the control to the beginning of the switch?

18.

```

char card = 3;
switch(card)
{
    case 1:
        printf("\n King");
    case 2:
        printf("Queen");
    default:
        printf("Jokar");
}

```

printf("\n You have losing the game");

19.

```

Int k;
float j = 2.0;
switch(k = j+1)
{
    case 3:
        printf("you have passed:");
        break;
    default:
        printf("Leave It");
}

```

20.

```

int a=300,b,c;
if(a>=400)
    b=300;
    c=200;
    printf("\n%d %d",b,c);

```

21.

```

int a=500,b,c;
if(a>=400)
    b=300;
    c=200;
    printf("\n%d %d",b,c);

```

Matrix Computers

22.
 int x = 10, y = 20;
 if(x == y);
 printf("\n%d %d", x, y);

23.
 int a = 3;
 float b = 3.0;
 if(a == b)
 printf("\n a and b are equal");
 else
 printf("\n a and b are not equal");

24.
 int x=3,y=5;
 if (x == 3)
 printf("\n %d",x);
 else;
 printf("\n%d",y);

25.
 int x = 3,y,z;
 y = x =10;
 z = x<10;
 printf("\n x=%d y=%d z = %d");

26.
 int k=35;
 printf("%d %d %d",k = 35,k = 50, k = 40);

27.
 int a=97;
 char b ='a';
 if(a == b)
 printf("hello matrix");
 else
 printf("bye matrix");

28.
 int x = 15;
 printf("%d %d %d ",x!=15,x = 20,x<30);

29.
 int ch = 'a'+ 'b' ;
 switch(ch)
{
 case 'a':
 case 'b':
 printf("you have secured a");
 case 'A':
 printf("you are confused:");

case 'b' + 'a'
 printf("you have secured
 both a and b ");
}

30.
 int i = 1;
 switch(i-2)
{
 case -1:
 printf("\nAnsI C");
 case 0:
 printf("\nTurbo C");
 case 1:
 printf("\nBorland C");
 default:
 printf("\nExploring C");
}

31.
 int a = 30, b = 40, x ;
 x = (a != 10) && (b = 50) ;
 printf(x = %d , x);

32.
 int a = 300, b = 10, c = 20 ;
 if (!(a > = 400))
 b = 300 ;
 c = 200 ;
 printf ("b = %d c = %d", b, c) ;

33.
 int x = 10, y = 100 % 90 ;
 if (x != y) ;
 printf ("x = %d y = %d", x, y) ;

34.
 int x = 3, y = 4, z = 4 ;
 printf ("ans=%d", (z > = y && y > = x ? 1:0)) ;

35.
 int x = 3, y = 4, z = 4 ;
 printf("ans=%d", (z > = y > = x ? 100:200)) ;

36.
 float a = 12.25, b = 13.65 ;
 if (a = b)
 printf ("a and b are equal") ;
 else
 printf ("a and b are not equal") ;

37.

```
int x = 10 ;
if x >= 2
    printf ("%d\n", x) ;
```

38.

```
int i = 10, j = 40 ;
if ((j - i) % 10)
    printf ("man sees your actions..") ;
else
    printf ("god sees your motives..") ;
```

39.

```
int i = -4, j, num = 10 ;
j = i % -3 ;
j = (j ? 0 : num * num) ;
printf ("j = %d", j) ;
```

40.

```
float a = 0.7 ;
if (a < 0.7)
    printf ("matrix") ;
else
    printf ("computers") ;
```

41.

```
float a = 0.7f ;
if (a < 0.7f)
    printf ("matrix") ;
else
    printf ("computers") ;
```

42.

```
int k = 12, n = 30 ;
k = (k > 5 && n = 4 ? 100 : 200) ;
printf ("k = %d", k) ;
```

43.

```
int c = 0, d = 5, e = 10, a ;
a = c > 1 ? d > 1 | e > 1 ? 100 : 200 : 300 ;
printf ("a = %d", a) ;
```

44.

```
int x = 10, y = 20 ;
if (!(x) && x)
    printf ("x = %d", x) ;
else
```

```
printf ("y = %d", y) ;
```

45.

```
int i = 2;
switch (i)
{
    case 1 :
        printf ("do") ;
    case 2 :
        printf ("re") ;
    case 3 :
        printf ("me") ;
    case default :
        printf ("fa so la ti do") ;
}
```

46.

```
int k = -2, j = 4 ;
switch (k /= j / k)
{
    default :
        printf ("all aresame!\n") ;
    case 0 :
        printf ("happy birthday\n") ;
    case 1 :
        printf ("a punch on the mouth\n") ;
    case 2 :
        printf ("a kick in the back\n") ;
}
```

47.

```
char ch = 'E' ;
switch (ch)
{
    case (ch >= 65 && ch <= 90) :
        printf ("capital letter") ;
        break ;
    case (ch >= 97 && ch <= 122) :
        printf ("small case letter") ;
        break ;
    case (ch >= 48 && ch <= 57) :
        printf ("digit") ;
        break ;
    default :
        printf ("any other character") ;
}
```

Output:-

M@trix Computers

- | | |
|--|---|
| 1. welcome to matrix | 22. 10 20 |
| 2. Hello | 23. a and b are equal |
| 3. Bye | 24. 3 |
| 4. 40 | 25. 5 |
| 5. int a=1,b,dummy;
a>10?b=20:(dummy=1);
Note that the following will not work:
a>10?b=20:: | 26. x = 10 y = 10 z = 0 |
| 6. Garbage | 27. hello matrix |
| 7. 30 | 28. 1 20 1 |
| 8. Hello | 29. you have secured both a and b. |
| 9. Bye Matrix | 30. Ansi C |
| 10. MatrixComputer | Turbo C |
| 11. Error constant expressions are required in switch, we cannot use j. | Borland C |
| 12. Hello | Exploring C |
| 13. Hello World. A switch can occur that does not have any statement. | 31. x = 1 |
| 14. Hello | 32. b = 300 c = 200 |
| Though there is no error but the first printf can never get executed. In other words all the statements in a switch must belong to some case or the other. | 33. x = 10 y = 10 |
| 15. No switch can work only on integer constants or constant expressions | 34. ans = 1 |
| 16. yes but it is not practical if the ranges are bigger. | 35. ans = 200 |
| 17. No, continue can work only with loops and not switch. | 36. a and b are equal |
| 18. Joker | 37. error message: if statement missing |
| You have lost the game: | 38. god sees your motives.. |
| 19. you have passed: | 39. j = 0 |
| 20. b contains garbage and c=200. | 40. matrix |
| 21. 300 200. | computers |
| | 41. error message: Lvalue required in function main |
| | 42. a = 300 |
| | 43. x = 10 |
| | 44. error message: expression syntax in function main |
| | 45. a punch on the mouth |
| | 46. a kick in the back |
| | 47. error message: constant expression required in main |

Programs to be solved in class:-

- WAP to check whether a number input through the keyboard is even or odd.
- WAP to print the maximum out of two numbers.
- WAP to print the maximum out of three numbers.
- WAP to print the maximum out of four numbers.
- WAP to print the maximum out of four numbers. (Short logic)
- WAP to print the second maximum out of three numbers.
- WAP to check whether a year is leap year or not.
- WAP to calculate the grade of a student after the input of marks of that student.
 Percentage ≥ 90 grade is 'A'
 Percentage ≥ 70 grade is 'B'
 Percentage ≥ 50 grade is 'C'
 Percentage < 50 grade is 'F'
- WAP to compute the tel. bill of a customer. Monthly Rental Rs.100.Rates are as follows.
 No of calls 1to 100 rate = 0
 No of calls 101 to 200 rate = 0.80

M@trix Computers

- No of calls 201to 500 rate = 1.00
 No of calls 501to -- rate = 1.20
11. Any year is entered through the keyboard, WAP to determine the year is leap or not. Use the logical operators && and ||
12. Any character is entered through the keyboard, WAP to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol.
13. WAP using conditional operators to determine whether a year entered through the keyboard is a leap year or not.
14. WAP to calculate real roots of a quadratic equation.
15. WAP to compute the simple interest and compound interest.

Lab Exercise:-

1. WAP to compute the pension of an employee.

If the person is male.

Age >= 90	pension is 4000
Age >= 60	pension is 6000,
Age < 60	pension is 0

If the person is female.

Age >= 90	pension is 3000
Age >= 60	pension is 5000
Age < 60	pension is 0

2. WAP to check whether a 3 digit number is a magic number or not. (Palindrome) A number is a magic number if its reverse is same as the original number.

Matrix

Diploma In Hardware

- Computer Assembling
- Software Installation
- Internet Installation
- Printer Installation
- Viruses and Antivirus

- DOS & Windows Operating Systems
- Trouble Shooting
- Repairing

Duration: 1 ½ months

```
.....  
.....  
}
```

}

The break statement would cause the exit from loop 2 but not from loop 1.

What would be the output of the following programs.

1.
 char j = 1 ;
 while (j <= 255)
 {
 printf ("%d ", j) ;
 j = j + 1 ;
 }

 2.
 int j = 1 ;
 while (j <= 255) ;
 {
 printf ("%c %d\n", j, j) ;
 j++ ;
 }

 3.
 int j = 1 ;
 while (j <= 255)
 printf ("%d\n", ++j) ;

 4.
 int a ;
 for (a = 1 ; a <= 32767; a++)
 printf ("%d", a) ;

 5.
 int i ;
 for (i = 1 ; i++ <= 5 ; printf ("%d", i)) ;

 6.
 int i = 1, j = 1 ;
 for (; j ; printf ("%d %d\n", i, j))
 j = i++ <= 5 ;

 7.
 int i = 1;
 for (; i++ ;)
 printf ("%d", i) ;

 8.
 int a = 5 ;
 do
 {
 printf ("%d\n", a) ;
 a = -1 ;

.....
 } while (a > 0) ;

 9.
 int i;
 for(i=-1;i<=10;i++)
 {
 if(i<5)
 continue;
 else
 break;
 printf("Gets printed only once!!");
 }

 10.
 for(i=1;i<=5;i--)
 printf ("%d",i);

 11.
 for(i=1;i<=5;++i)
 printf ("%d",i);

 12.
 for(i=1;++i<=5;)
 printf ("%d",i);

 13.
 for(i=1;i++<=5;)
 printf("%d",i);

 14.
 for(i=1;i<=5;printf("%d",i++));

 15.
 for(i=1;i<=5;printf("%d",++i));

 16.
 for(i=1;i=5;i++)
 printf ("%d",i);

 17.
 for(i=1;i=0;i++)
 printf("%d",i);

 18.
 for(; ;)

19.
for(i=1,i++<=5,i++)
 printf("%d",-i);

20.
for(i=1,i==5,i++)
 printf("%d",i);

21.
for(i=1;i<=5;)
 printf("%d",i++);

22.
for(i=1;5;i++)
 printf("%d",i);

23.
for(i=1;i<=5;i++)
 printf("%d",i);

24.
int i=1;
for(;i<=5;i++)
 printf("%d",i);

25.
int i=1;
while()
{
 printf("%d",i);
 i++;
}

26.
int i=1;
while(i<=5)
 printf("%d",i++);

27.
for(i=1;i<4;i++)
 printf("%d", (i%2)?i:2*i);

28.
int i=1;
while(1)
{
 printf("%d",i);
 i++;
}

29.

int i =1;
while(i<= 10);
{
 printf(" \n%d",i);
 i++;
}

30.
char ch;
while(ch =0; ch<=255;ch++)
 printf("\n %d-%c",ch,ch);

31.
int a =4;
while(a ==1)
{
 a = a-1;
 printf("%d",a);
 --a;
}

32.
~~int a =4,b,c;~~
~~b = -a;~~
~~c = a--;~~
printf("\n%d %d %d",a,b,c);

33.
int a =4,b =3,c;
c = a-- -b;
printf("\n%d%d%d",a,b,c);

34.
while('a'<'b')
 printf("\n malyalam is a
palindrome");

35.
int i;
while(i =15)
{
 printf("\n%d",i);
 i++;
}

36.
float a = 1.1;
while(a ==1.1)
{
 printf("\n%d",x);
 x=x-0.1;
}

}

37.

```
int i = 0;
for(;i)
    printf("\n Hello Matrix");
```

38.

```
int a =4, b =0, c;
while(a>=0)
{
    a--;
    b++;
    if( a ==b)
        continue;
    else
        printf(" \n %d %d",a,b);
}
```

39.

```
int a = 4, b = 0,c;
while( a>=0)
{
    if(a ==b)
        break;
    else
        printf(" \n %d %d",a,b);
    a--;
    b++;
}
```

40.

```
int i;
for(i = 1;i<=5;printf("\n%d",i));
    i++;
```

41.

```
int i = 1,j = 1;
for(; ;)
{
    if(i>5)
        break;
    else
        j+=i;
    printf("%d\n",j);
    i+=j;
}
```

42.

The three parts of the for loop are:

- a. The i _____ expression

- b. The c _____ expression
c. The i _____ expression

43.

The break statement is used to exit from:

- a. An if statement.
b. A for loop
c. A program
d. The main() function

44.

A do-while loop is useful when we want that the statements within the loop must be executed:

- a. Only once
b. Atleast once
c. More than once
d. None of the above.

45.

In what sequence the initialization, condition, execution is done in the do-while loop

- a. Initialization, testing, execution.
b. Initialization, execution, testing
c. Testing, execution, Initialization

46.

Which of the following statements is used to take the control to the beginning of the loop

- a. exit
b. break
c. continue
d. none of the above.

47.

```
int i=1;
for( ;i++ ;)
    printf("%d", i);
```

48.

```
int a=5;
do
{
    printf("%d\n",a);
    a = -1;
}while(a >0);
```

M@trix Computers

49.

What will be the value of sum after the execution of the following program.

```
int sum, index;
sum=1;
```

Output:-

1. Infinite loop 1 2 3 ... 127 -128 -127
-126 ... 0 1 ...
2. Infinite loop but no output
3. 2 3 ... 256
4. 2 3 ... 32766 32767 -32768 -
32767..0 1 2..32767.. infinite loop
5. 2 3 4 5 6
6. 2 1
3 1
4 1
5 1
6 1
7 0
7. 2 3...32767...-32768 -32767...-1 0
8. 5
9. No Output
10. 1,0,-1,-2.....-32768
11. 1,2,3,4,5
12. 2,3,4,5
13. 2,3,4,5,6
14. 1,2,3,4,5
15. 2,3,4,5,6
16. 5,5,5,5.....
17. loop will not run as the condition
will be considered false because at
the place of condition there is an
assignment of 0.
18. infinite loop but no output.
19. 1,2,3,4,5
20. No output will be generated
because the condition is checking
if the value of i is 5 which is false
in the beginning.
21. 1,2,3,4,5
22. infinte loop 1,2,3,4,5.....

```
index=9;
do
{
    index=index-1;
    sum=sum*2;
}while(index>9);
```

23. 6
24. 1,2,3,4,5
25. Error as condition is missing and
this is not allowed in while loop.
26. 1,2,3,4,5
27. 1,4,3,8,5
28. infinite loop
29. No output because a infinite loop.
30. Error.
31. No output
32. 2 3 3
33. 3 3 1
34. "malyalam is a palindrome" printed
indefinitely.
35. print 15 indefinitely.
36. No output.
37. No output.
38. 3 1
1 3
0 4
-1 5
39. 4 0
3 1
40. 1 will printed indefinitely no. of
times.
41. 2 5
42. initialization,condition,increment
43. b
44. b
45. b
46. c
47. 1,2,3..32767, -32768,..infinite loop
48. 5
49. 2

Programs to be solved in class:-

1. 1,2,3,4,5,.....n terms
2. n,.....4,3,2,1
3. 1, 3, 5, 7, 9,n terms
4. 2, 4, 6, 8, 10,n terms
5. 1, 4, 9, 16, 25,n terms
6. 1, 8, 27, 64, 125,n terms

M@trix Computers

7. 1, -1, 1, -1, 1,n terms
 8. 1, 0, 1, 0, 1,n terms
 9. 1, -3, 5, -7, 9,n terms
 10. 9,99,999,9999,.....n terms
 11. 1,1,2,3,5,8,13,.....n terms (Fibonacci series)
 12. 1, 1, 2, 4, 7, 13, 24, n terms (Lucas series)
 13. WAP to print the table of n.
 14. WAP to calculate the factorial of n.
 15. WAP to calculate the power p of a number n.
 16. WAP to print the sum of all the digits of a given number.
 17. WAP to count the number of digits in a given number.
 18. WAP to reverse a number.
 19. WAP to check whether a given number is palindrome or not.
 20. WAP to print all ascii characters along with their ascii values.
 21. WAP to fill the entire screen with smiling face. (ASCII is 1).
 22. WAP to print all the combinations of 1,2 & 3.
 23. 1
 12
 123
 1234
 12345
 24. 1
 22
 333
 4444
 55555
 25. *
 **

 26. 1
 21
 321
 4321
 54321
 27. 12345
 1234
 123
 12
 1
 28. 54321
 4321
 321
 21
 1

try

29. 1
 23
 456
 7890
 12345

try

30. 1
 01
 101
 0101
 10101

try

31. a
 ab
 abc
 abcd
 abcde

try

32. 1
 123
 12345
 1234567
 123456789

try

33. 1
 12
 123
 1234
 12345

try

34. 1
 123
 12345
 1234567
 123456789

35.

```

      *
     ***
    * * * *
   * * * * *
  * * * * * *

```

36.

```

    1
   121
  12321
 1234321
123454321

```

39. WAP to print pascal triangle.

```

    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
1 5 10 10 1

```

40. WAP to check whether a given number is armstrong or not.
 41. WAP to check whether a given number is prime or not.
 42. WAP to print all the prime numbers between any two given numbers.
 43. WAP to find the sum of first n prime numbers.
 44. WAP to count all the vowels, consonants, digits, spaces, special symbols from a given text typed by the user, terminated by the enter key.
 45. $1 + x^2/2! + x^3/3! + x^4/4! + x^5/5! \dots \dots \dots (e^x)$
 46. $x - x^3/3! + x^5/5! - x^7/7! + x^9/9! \dots \dots \dots (\sin x)$
 47. $1 - x^2/2! + x^4/4! - x^6/6! + x^8/8! \dots \dots \dots (\cos x)$
 48. WAP to find the maximum of n numbers.
 49. WAP to find the average of the values read from the input. The sequence of values in the input is terminated by -1.
 50. WAP to find the number of times the minimum value has appeared in the given sequence of n numbers.

37

```

    1
   232
  34543
 4567654
567898765

```

38

```

    4
   434
  43234
4321234
 43234
 434
    4

```

Some Solved Programs:-

- 4.1 WAP to check if the given sequence of numbers is in ascending order or not. The sentinel value for the sequence is -1.
1. #include<stdio.h>
 2. #include<conio.h>
 3. enum {false,true};
 4. main()
 5. {
 6. int prev,n,flag=true,i;
 7. clrscr();
 8. printf("Enter number 1 ");
 9. scanf("%d",&n);

```

10.     prev = n;
11.     i=2;
12.     while(n != -1)
13.     {
14.         printf("Enter number %d ",i++);
15.         scanf("%d",&n);
16.         if(n < prev && n != -1)
17.             flag=false;
18.         prev=n;
19.     }
20.     if(flag == false)
21.         printf("Not in ascending order");
22.     else
23.         printf("In ascending order");
24.     getch();
25. }
```

4.2 Program to find the square root of positive no.

```

1. #include<stdio.h>
2. #include<conio.h>
3. #include<process.h>
4. #include<math.h>
5. main()
6. {
7.     float num,low=2,high;
8.     long a,b;
9.     clrscr();
10.    printf("Enter a number ");
11.    scanf("%f",&num);
12.    if(num<0)
13.    {
14.        printf("Square root of negative no. is not allowed");
15.        exit(0);
16.    }
17.    while(1)
18.    {
19.        high=num/low;
20.        a=high*10000;
21.        b=low*10000;
22.        if(abs(a-b)<2)
23.        {
24.            printf("Square root : %10.4f",low);
25.            break;
26.        }
27.        low=(low+high)/2;
28.    }
29.    getch();
30. }
```

4.3 WAP to convert decimal no. to its binary equivalent

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
```

M@trix Computers

```

4.    {
5.        int n,i,a,r;
6.        clrscr();
7.        printf("enter the no.: ");
8.        scanf("%d",&n);
9.        for(i=15;i>=0;i--)
10.       {
11.           a=1<<i;
12.           r=n&a;
13.           if(r==0)
14.               printf("0");
15.           else
16.               printf("1");
17.       }
18.       getch();
19.   }

```

4.4 WAP to compute the natural logarithm of a given number

```

1. #include<stdio.h>
2. #include<conio.h>
3. #include<math.h>
4. main()
5. {
6.     int x, i;
7.     float result=0;
8.     clrscr();
9.     printf("\nEnter the value of x:");
10.    scanf("%d",&x);
11.    for(i=1;i<=7;i++)
12.    {
13.        if(i==1)
14.            result=result+pow((x-1.0)/x,i);
15.        else
16.            result=result+(1.0/2)*pow((x-1.0)/x,i);
17.    }
18.    printf("Log(%d) = %f",x,result);
19.    getch();
20. }

```

4.5 WAP to generate n random numbers.

```

1. #include<stdio.h>
2. #include<conio.h>
3. #include<stdlib.h>
4. main()
5. {
6.     int a,i,n;
7.     clrscr();
8.     randomize(); /*initializes the random number generator with a random
9.                     number*/
10.    printf("enter the no. to be printed: ");
11.    scanf("%d",&n);
12.    i=1;
13.    while(i<=n)

```

M@trix Computers

```

14.     {
15.         a=random(100)+1; /*generates a number between 0 and 99*/
16.         printf("%d\n",a);
17.         i++;
18.     }
19.     getch();
20. }
```

4.6 WAP to produce the following output:

```

a b c d e f g f e d c b a
a b c d e f   f e d c b a
a b c d e     e d c b a
a b c d       d c b a
a b c         c b a
a b           b a
a             a
```

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     int n,i,j,k,l;
6.     clrscr();
7.     printf("Enter the number of Rows: ");
8.     scanf("%d",&n);
9.     for(i=n;i>=1;i--)
10.    {
11.        for(j=1;j<=i;j++)
12.            printf("%c",j+97-1);
13.        for(k=1;k<=2*(n-i)-1;k++)
14.            printf(" ");
15.        for(l=i;l>=1;l--)
16.        {
17.            if(l==n)
18.                continue; /*passes the control of the loop directly to the decrement
19.                           statement*/
20.            printf("%c",l+97-1);
21.        }
22.        printf("\n");
23.    }
24. }
25. }
```

M@trix

4.7 WAP to print the total no. of characters typed by the user. Input will be terminated by enter key.

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     char ch;
6.     int i=0;
7.     clrscr();
8.     printf("Enter a sentence: ");
9.     while((ch=getche())!="\r") /*getche( ) is an input function */
```

M@trix Computers

```

10.    {
11.        i++;
12.    }
13.    printf("\n%d",i);
14.    getch();
15. }
```

4.8 WAP to reverse a number and find its octal equivalent.

```

1. #include<stdio.h>
2. #include<conio.h>
3. #include<math.h>
4. main()
5. {
6.     int n, rev, a, oct, cnt;
7.     clrscr();
8.     printf("\nEnter a number");
9.     scanf("%d",&n);
10.    /*reverse the number*/
11.    rev=0;
12.    while(n>0)
13.    {
14.        a=n%10;
15.        rev=rev*10+a;
16.        n=n/10;
17.    }
18.    n=rev;
19.    cnt=oct=0;
20.    /*converting to octal*/
21.    while(n>0)
22.    {
23.        a=n%8;
24.        n=n/8;
25.        oct=oct+a*pow(10,cnt);
26.        cnt++;
27.    }
28.    printf("The octal equivalent of %d is %d",rev,oct);
29.    clrscr();
30. }
```

Lab Exercise:-

- WAP to find the maximum, second maximum, position of maximum and position of second maximum from the sequence of n numbers.
- WAP to print all prime numbers from 1 to 300.
- WAP to add first n terms of the following series using for loop: $1/1!+2/2!+3/3!+\dots$
- WAP to determine whether a specified value is prime or not in a given sequence of values. The sequence of values to be read from the input is terminated by 0.
- Print the sum of following series.

$$1 \times 2 + 2 \times 3 + 3 \times 4 + 4 \times 5 + \dots + (n-1) \times n.$$
- Print the sum of following series.

$$e^x = 1 - x + x^2/2! - x^3/3! + x^4/4! \dots$$
- WAP to read n numbers and count even and odd numbers.

$$1+1/2+1/3+1/4+\dots+1/n$$

- 8 WAP to print the sum of n numbers , sum of squares of first n even numbers and sum of the cube of first n odd numbers.
- 9 WAP to calculate net pay of n employees. Net pay is basic + da +hra cca - pf (da is 39% of the basic, hra is 15% of basic less than or equal to rs 8000/- and 30% of the basic above rs 8000, cca is fixed to rs 800 and pf deduction is rs 600).
- 10 Write a program to calculate overtime pay of 10 employees. Overtime is paid at the rate of rs. 12.00 per hour for every hour worked above 40 hours. Assume that employees do not work for fractional part of an hour.
- 11 While purchasing certain items, a discount of 10% is offered if the quantity purchased is more than 1000.if quantity and price per item are input through the keyboard, write a program to calculate the total expenses.
- 12 WAP to find the sum and average of values appearing at the positions divisible by 3 in the given sequence of n values.
- 13 WAP to input the marks of n students and count the number of students who have obtained a, b, c,d and f grades. The grades are awarded according to the following rules.

Marks	grade
≥ 80	A
≥ 70	B
≥ 60	C
≥ 50	D
< 50	F

14. $x - x^2/2 + x^3/3 - x^4/4 + x^5/5 \dots \dots \dots (\log(1+x))$
15. WAP to input two numbers and calculate their LCM & HCF.

M@trix

65 = A
90 = Z

```
int a[10], i;
for(i=0; i<=20; i++)
    a[i] = i;
```

What would be the output of the following programs?

1.

```
int array[26], i;
for (i = 0; i <= 25; i++)
{
    array[i] = 'A' + i;
    printf("\n%d\t%c", array[i], array[i]);
}
```

2.

```
int sub[50], i;
for( i = 0; i <= 49; i++)
{
    sub[i] = i;
    printf("\n%d", sub[i]);
}
```

3.

```
static int a[5];
int i;
for( i = 0; i <= 4; i++)
    printf("%d ", a[i]);
```

4.

```
int a[ 5 ] = {5,1,15,20,25};
int i,j,k = 1,m;
```

Max i++

```
i = ++a[1]; 5 -> 1
j = a[1]++; 1 -> 3
m = a[i++];
```

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

Run

5.

```
int i, a = 2, b = 3; 3
int a[2+3];
for( i = 0; i < a+b; i++)
{
    scanf("%d", &a[i]);
    printf("\n%d", a[i]);
}
```

Run

6.

2 1200
2 1202
4 1204
1 1206

6 1208

Assume that array begins at 1200
int a[] = {2, 3, 4, 1, 6};
printf("%d %d", a, sizeof(a));

Assume that the array begins at 65486?
int a[] = {12, 14, 15, 23, 45};
printf("%u %u", a+1, &a+1);

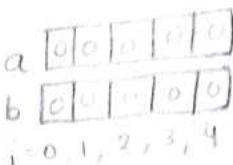
Run

8.

```
int a[5], i;
```

M@trix Computers

```
static int b[5] ;
for(i = 0 ; i < 5 ; i++)
    printf("%d %d %d\n", i, a[i], b[i]);
```



Run

⑨

```
static int sub[5] = {10, 20, 30, 40, 50} ;
int i ;
for(i = 0 ; i <= 4 ; i++) {
    if(i <= 4)
    {
        sub[i] = i*i ;
        printf("%d\n", sub[i]) ;
    }
}
```

0, 4

Run

⑩

```
int size = 10 ;
int arr[size] ;
for(i = 1 ; i <= size ; i++)
{
    scanf("%d", &arr[i]) ;
    printf("\n%d", arr[i]) ;
}
```

11.

```
static int b[] = {10, 20, 30, 40, 50} ;
int i ;
for(i = 0 ; i <= 4 ; i++)
    printf("%d", i[b]) ;
```

Output:

1. 65 A
66 B

90 Z
2. 49
3. 0 0 0 0
4. 3 2 15
5. It will scan 5 values & print them at the same time
6. 1200 10
7. 65488 65496

Matrix

8. 0 100 0
1-75 0
2 123 0
3 1245 0
4 347 0
no output
10. 9.
11. error message: constant expression required in function main
10 20 30 40 50

State whether the following statements are True or False:

1. The array int num[26] has twenty-six elements.
2. The expression num[1] designates the first element in the array.
3. It is necessary to initialize the array at the time of declaration F
4. The expression num[27] designates the twenty-eighth element in the array.
5. Address of a floating-point variable is always a whole number. T

Answers:

1. False
2. False
3. False
4. True
5. True

Programs to be solved in class:

- Notes*
1. WAP to find the position of an element in a given array of n numbers using Linear Search Technique.
 2. WAP to find the position of an element in a given array of n numbers using Binary Search Technique.
 3. WAP to sort a given array of n numbers in ascending order using Sequential Sorting Technique.
 4. WAP to sort a given array of n numbers in ascending order using Bubble Sorting Technique.
 5. WAP to sort a given array of n numbers in ascending order using Insertion Sorting Technique.
 6. WAP to find the maximum of numbers stored in an array of n.
 7. WAP to insert a new element in an unsorted array at a given position.
 8. WAP to insert a new element in a sorted array.
 9. WAP to merge two sorted arrays into a third array.

Some Solved Programs:-

Q5.1 Write a program to print the average of n numbers.

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     int a[50],n,i,sum=0;
6.     float avg;
7.     clrscr();
8.     printf("Enter the number of elements in the array: ");
9.     scanf("%d",&n);
10.    /*input & sum of array*/
11.    for(i=0;i<n;i++)      /*the first element in the array is numbered as 0, so
12.                           the last element is one less than the size of the array*/
13.    {
14.        printf("Enter number %d",i+1);
15.        scanf("%d",&a[i]);
16.        sum += a[i];
17.    }
18.    avg=(float)sum/n;      /*typecasting sum from interger to float*/
19.    /*output*/
20.    printf("average = %f",avg);
21.    getch();
22. }
```

Q5.2 WAP that reads a float array and reverse this array.

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     float a[10];
6.     int n,i;
7.     clrscr();
8.     printf("Enter the number of terms: ");
9.     scanf("%d",&n);
10.    /*input*/
11.    for(i=0;i<n;i++)
12.    {
13.        printf("Enter the number %d",i);
14.        scanf("%f",&a[i]);
15.    }
16.    /*to print the reverse*/
17.    for(i=n-1;i>=0;i--)
18.    {
19.        printf("%.2f, ",a[i]); /*to print the number upto two decimal places only*/
20.    }
21.    getch();
22. }
```

5.3 Some repeated random numbers are given, write a program to print them in increasing order with their frequency

```
1. #include<stdio.h>
2. #include<conio.h>
3. #define SIZE 100
4. main()
5. {
6.     int a[SIZE],b[SIZE],freq[SIZE],i,j,k,n,found,t;
7.     clrscr();
8.     printf("Enter how many numbers");
9.     scanf("%d",&n);
10.    /*input array*/
11.    for(i=0;i<n;i++)
12.    {
13.        printf("enter element %d ",i+1);
14.        scanf("%d",&a[i]);
15.    }
16.    /* frequency count */
17.    for(i=0,k=0;i<n;i++)
18.    {
19.        found = 0;
20.        for(j=0;j<k;j++)
21.        {
22.            if(a[i] == b[j])
23.            {
24.                freq[j]++;
25.                found = 1;
26.                break;
27.            }
28.        }
29.        if(found == 0)
30.        {
31.            b[k] = a[i];
32.            freq[k++] = 1;
33.        }
34.    }
35.    /*selection sorting*/
36.    for(i=0;i<k-1;i++)
37.    {
38.        for(j=i+1;j<k;j++)
39.        {
40.            if(freq[i] < freq[j])
41.            {
42.                t = b[i];
43.                b[i] = b[j];
44.                b[j] = t;
45.                t = freq[i];
46.                freq[i] = freq[j];
47.                freq[j] = t;
48.            }
49.        }
50.    }
51.    /*output*/
52.    for(i=0;i<k;i++)
53.        printf("%d\t%d\n",b[i],freq[i]);
54.    getch();
```

M@trix Computers

5.4 WAP to merge two array a and b into third array c.

```

1. #include<stdio.h>
2. #include<conio.h>
3. #define SIZE 20
4. main()
5. {
6.     int n1,n2,i,j,k,a[SIZE],b[SIZE],c[SIZE];
7.     clrscr();
8.     /*input in first array*/
9.     printf("Enter the no. of elements in the first array: ");
10.    scanf("%d",&n1);
11.    for(i=0;i<n1;i++)
12.    {
13.        printf("Enter the element %d: ",i+1);
14.        scanf("%d",&a[i]);
15.    }
16.    /*input in second array*/
17.    printf("\nEnter the no. of elements in the second array: ");
18.    scanf("%d",&n2);
19.    for(i=0;i<n2;i++)
20.    {
21.        printf("Enter the element %d: ",i+1);
22.        scanf("%d",&b[i]);
23.    }
24.    /*merge*/
25.    for(i=0,j=0,k=0;i<n1&&j<n2;k++)
26.    {
27.        if(a[i]<b[j])
28.            c[k]=a[i++];
29.        else
30.            c[k]=b[j++];
31.    }
32.    /*remaining of first list*/
33.    while(i<n1)
34.        c[k++]=a[i++];
35.    /*remaining of second list*/
36.    while(j<n2)
37.        c[k++]=b[j++];
38.    /*output*/
39.    for(i=0;i<(n1+n2);i++)
40.        printf("%d\n",c[i]);
41.    getch();
42. }
```

5.5 WAP to find the kth smallest number from a given list of numbers:

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     int a[20], i, j, n,t,k;
6.     clrscr();
7.     printf("Enter the number of elements in the list: ");
```

M@trix Computers

```

8.         scanf("%d",&n);
9.         /*input in the array*/
10.        for (i=0;i<n;i++)
11.        {
12.            printf("Enter the number %d: "i+1);
13.            scanf("%d",&a[i]);
14.        }
15.        /* loop to sort the array*/
16.        for(i=0;i<n-1;i++)
17.        {
18.            for(j=i+1;j<n;j++)
19.            {
20.                if(a[i]>a[j])
21.                {
22.                    t=a[i];
23.                    a[i] = a[j];
24.                    a[j]=t;
25.                }
26.            }
27.        }
28.        printf("Enter the position of element: " );
29.        scanf("%d",&k);
30.        printf("the %d smallest element of the list is %d",k, a[k-1]);
31.        getch();
32.    }

```

5.6 WAP that will read an array of integers and print even and odd element separately.

```

1. #include<stdio.h>
2. #include<conio.h>
3. void main()
4. {
5.     int a[100],n,i;
6.     clrscr();
7.     printf("Enter how many nos.:");
8.     scanf("%d",&n);
9.     /*Array Input*/
10.    for(i=0;i<n;i++)
11.    {
12.        printf("Enter element %d:",i+1);
13.        scanf("%d",&a[i]);
14.    }
15.    /*Display Even elements*/
16.    printf("\nEven numbers are:\n");
17.    for(i=0;i<n;i++)
18.    {
19.        if(a[i]%2==0)
20.            printf("%d ",a[i]);
21.    }
22.    /*Display Odd elements*/
23.    printf("\nOdd numbers are:\n");
24.    for(i=0;i<n;i++)
25.    {

```

```

26.         if(a[i]%2!=0)
27.             printf("%d ",a[i]);
28.     }
29.     getch();
30. }

```

- 5.7 WAP to find sum of element appearing at even and odd subscript position of an array of integers.

```

1. #include<stdio.h>
2. #include<conio.h>
3. void main()
4. {
5.     int a[100],n,i;
6.     clrscr();
7.     printf("Enter how many nos.:");
8.     scanf("%d",&n);
9.     /*Array Input*/
10.    for(i=0;i<n;i++)
11.    {
12.        printf("Enter element %d:",i+1);
13.        scanf("%d",&a[i]);
14.    }
15.    /*Display Even subscript elements*/
16.    printf("\nEven position numbers are:\n");
17.    for(i=0;i<n;i++)
18.    {
19.        if(i%2==0)
20.            printf("%d ",a[i]);
21.    }
22.    /*Display Odd subscript elements*/
23.    printf("\nOdd position numbers are:\n");
24.    for(i=0;i<n;i++)
25.    {
26.        if(i%2!=0)
27.            printf("%d ",a[i]);
28.    }
29.    getch();
30. }

```

Lab Exercise:-

1. WAP to find maximum and the minimum values from a set of values stored in an array, along with their positions in the array.
2. WAP to read 6 digits and find out if they are in a strictly ascending order. For example, the sequence 5,6,7,9,11,14 is in strict ascending order whereas the sequence 5,5,6,7,9,11 is not in a strict ascending order. Display an appropriate message.
3. WAP to read a set of height and find out the average height. The program should then calculate the deviation of each height from the average. The deviation d_i is defined as: $d_i = m - a$. Where a represents the average height, and m_i represents the height.
4. WAP that will read 10 integers into an array and then display their averages.
5. WAP that will display the maximum and its position in an array of integers. If the maximum occurs more than once its last position should be displayed.
6. WAP that will read roll no. and marks of 10 students in two different arrays. Program will print the marks of students whose roll no. is provided by user.

M@trix Computers

- ✓ 7. WAP that will read 2 array, sum their individual element in third array and display the third array.
- ✓ 8. WAP that will read an array of integers. after reading array, the program should check if there is any duplicate value in the array. The program should display the appropriate message.
- ✓ 9. WAP that will read 2 arrays. Sum individual elements of both arrays in reverse order and stores it in third array. Display the third array. *and look from outside*
- ✓ 10. WAP that will read an array and insert an integer at the end of array.
- ✓ 11. WAP to insert an integer at the beginning of an array.
- ✓ 12. WAP to insert an array at the end of another array.
- ✓ 13. WAP to insert an array into another array at a positions specified by user.
- ✓ 14. WAP to find whether a array is palindrome or not.
- ✓ 15. WAP to delete an array element present at the beginning.
- ✓ 16. WAP to left rotate an array by one element.
- ✓ 17. WAP to right rotate an array by one element.
- ✓ 18. WAP to left rotate an array by n elements, where value of n will be provided by user.
- ✓ 19. WAP to right rotate an array by n elements, where value of n will be provided by user.
- ✓ 20. WAP that will read an array, replaces multiple occurrence of any element by 0 and then display the resultant array. For eg. If input is 1 1 2 2 3 4 2 1 5 4 output is 1 0 2 0 3 4 0 0 5 0 .
- ✓ 21. WAP to shift multiple occurrence of element in the following manner. for eg. If input is 1 1 2 2 3 4 2 1 5 4 output is 1 2 3 4 5 0 0 0 0 0 .
- ✓ 22. WAP to delete the multiple occurrence of elements in an array.
- ✓ 23. WAP to replace any nth element of an array at the first position, the (n+1)th element at second position etc.
- ✓ 24. WAP to rearrange kth elements of an array so as to replace the elements at the odd suffixes with the elements at even suffixes. for eg, 1 2 3 4 5 6 7 8 should be changed to as: 1 3 5 7 2 4 6 8
- ✓ 25. WAP to store any ten numbers in an array and print the LCM and HCF of all the numbers.
- ✓ 26. WAP to store any 100 numbers in an array. Arrange the first fifty numbers in ascending order and last fifty numbers in descending order and print the sorted array.
- ✓ 27. WAP which accepts a positive decimal integers input from the keyboard converts the integer into its binary equivalent and outputs the integer with its binary equivalent.
- ✓ 28. WAP to create a array age[20] to store any twenty ages and print the sum of all even and odd ages respectively.
- ✓ 29. 25 numbers are entered from the keyboard into an array. WAP to find out how many of them are positive, how many are negative , how many are even and how many are odd.
- ✓ 30. WAP to store any ten numbers in an array number and print the smallest, the largest and the average.
- ✓ 31. WAP to send all the negative elements of an array to the end without altering the original sequence. for e.g. If array contains 5 -3 2 6 8 -4 7 -6 9 -1 then the resultant array should be 5 6 8 7 9 -3 -4 -6 -1 .you may use two arrays.
- ✓ 32. WAP that will display the max and min and their respective positions in an array. If the max and min occurs more than once their first position should be displayed.

M@trix Computers

Write the output of the following:

```
1. static int n[3][3] = {2,4,3,6,8,5,3,5,1} ;
/*assume that array begins at address 404*/
printf ("%d %d %d", n, n[2], n[2][2]);
```

Output:

```
1. 404 416 1
```

Programs that will be solved in class:

1. WAP to print the transpose of a matrix.
2. WAP to sum two matrices and store the result in 3rd matrix.
3. WAP to multiply two matrices and store the result in 3rd matrix.
4. WAP to convert a 3x3 matrix into 4x4 matrix by adding its rows & columns.
5. WAP to check whether a given matrix is diagonal matrix or not.

Some solved Programs:-

6.1 Write a program to find if a square matrix is symmetric.

```
1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     int m[10][10], i, j, r,c;
6.     clrscr();
7.     printf("Enter number of rows & columns of the Matrix:");
8.     scanf("%d%d",&r,&c);
9.     if(r !=c)
10.    {
11.        printf("Symmetric matrix must be a square matrix");
12.        getch();
13.        exit(1);
14.    }
15.    /*Matrix input*/
16.    for(i=0; i<r; i++)
17.    {
18.        for(j=0; j<c; j++)
19.        {
20.            printf("Enter element %d,%d",i+1,j+1);
21.            scanf("%d",&m[i][j]);
22.        }
23.    }
24.    /* Check for symmetry*/
25.    for(i=0; i<r; i++)
26.    {
27.        for(j=0; j<c; j++)
28.        {
29.            if(m[ i ][ j ] != m[ j ][ i ])
30.            {
31.                printf("The matrix is not a symmetric");
32.                getch();
33.                exit(0);
34.            }
35.        }
}
```

```

36.         }
37.         printf("The matrix is a symmetric");
38.         getch();
39.     }

```

6.2 WAP to find the row sum & column sum of a matrix.

```

1. #include<stdio.h>
2. #include<conio.h>
3. void main()
4. {
5.     int m1[10][10],r,c,rowsum,colsum[10]={0},i,j;
6.     clrscr();
7.     printf("Enter how many rows & Columns:");
8.     scanf("%d%d",&r,&c);
9.     /*Input matrix*/
10.    for(i=0;i<r;i++)
11.        for(j=0;j<c;j++)
12.        {
13.            printf("Enter element %d,%d:",i+1,j+1);
14.            scanf("%d",&m1[i][j]);
15.        }
16.    /*Display matrix & its row, column sum*/
17.    for(i=0;i<r;i++)
18.    {
19.        rowsum=0;
20.        for(j=0;j<c;j++)
21.        {
22.            printf("%d\t",m1[i][j]);
23.            rowsum+=m1[i][j];
24.            colsum[j]+=m1[i][j];
25.        }
26.        printf("\n");
27.    }
28.    for(j=0;j<c;j++)
29.        printf("==\t");
30.    printf("\n");
31.    for(j=0;j<c;j++)
32.        printf("%d\t",colsum[j]);
33.    getch();
34. }

```

6.3 WAP to prepare a one-dimensional array a[n2] from a 2 dimensional array m[nxn] that will have all the elements of array m if they are stored in row-major form and a one-dim array b[n2] in column-major form. for example for the following array

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

the resultant array a will be 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16, and
 the resultant array b will be 1 5 9 13 2 6 10 14 3 7 11 15 4 8 12 16

```

1. #include<stdio.h>
2. #include<conio.h>
3. void main()
4. {
5.     int m[10][10],a[100],b[100],i,j,n,cnt1=0,cnt2=0;

```

M@trix Computers

```

6.     clrscr();
7.     printf("Enter number of rows of square matrix:");
8.     scanf("%d",&n);
9.     /*Input matrix*/
10.    for(i=0;i<n;i++)
11.        for(j=0;j<n;j++)
12.        {
13.            printf("Enter element %d,%d:",i+1,j+1);
14.            scanf("%d",&m[i][j]);
15.        }
16.    /*copy matrix in a & b array*/
17.    for(i=0;i<n;i++)
18.        for(j=0;j<n;j++)
19.        {
20.            a[cnt1++]=m[i][j];
21.            b[cnt2++]=m[j][i];
22.        }
23.    /*Display a array */
24.    printf("\nRow major   =");
25.    for(i=0;i<cnt1;i++)
26.        printf("%d ",a[i]);
27.    printf("\nColumn major =");
28.    for(i=0;i<cnt2;i++)
29.        printf("%d ",b[i]);
30.    getch();
31. }
```

Ques 6.4 WAP to create a square matrix and print the first and the second diagonal elements on a clear screen. Also find the sum of all the elements lie on either diagonal.

```

1. #include<stdio.h>
2. #include<conio.h>
3. #include<stdlib.h>
4. void main()
5. {
6.     int m1[10][10],r,c,i,j,sumdiag1=0,sumdiag2=0;
7.     clrscr();
8.     printf("Enter row & column of diagonal matrix:");
9.     scanf("%d%d",&r,&c);
10.    if(r!=c)
11.    {
12.        printf("Row & Column must be same for diagonal matrix");
13.        getch();
14.        exit(1);
15.    }
16.    /*input matrix*/
17.    for(i=0;i<r;i++)
18.        for(j=0;j<c;j++)
19.        {
20.            printf("Enter element %d,%d:",i+1,j+1);
21.            scanf("%d",&m1[i][j]);
22.        }
23.    /*Calculating sum of first diagonal*/
24.    for(i=0;i<r;i++)

```

M@trix Computers

```

25.         for(j=0;j<c;j++)
26.         {
27.             if(i==j)
28.             {
29.                 printf("%d\t",m1[i][j]);
30.                 sumdiag1+=m1[i][j];
31.             }
32.         }
33.         printf("\nSum of first diagonal is:%d\n",sumdiag1);
34. /*Calculating sum of second diagonal*/
35.         for(i=0;i<r;i++)
36.             for(j=0;j<c;j++)
37.             {
38.                 if(i+j==r-1)
39.                 {
40.                     printf("%d\t",m1[i][j]);
41.                     sumdiag2+=m1[i][j];
42.                 }
43.             }
44.             printf("\nSum of second diagonal is:%d\n",sumdiag2);
45.             getch();
46.     }

```

6.5 WAP to print all those elements of a matrix that are not present in either of the two diagonals.

```

1. #include<stdio.h>
2. #include<conio.h>
3. #include<stdlib.h>
4. void main()
5. {
6.     int m1[10][10],r,c,i,j;
7.     clrscr();
8.     printf("Enter row & column of diagonal matrix:");
9.     scanf("%d%d",&r,&c);
10.    if(r!=c)
11.    {
12.        printf("Row & Column must be same for diagonal marix");
13.        getch();
14.        exit(1);
15.    }
16. /*input matrix*/
17.    for(i=0;i<r;i++)
18.        for(j=0;j<c;j++)
19.        {
20.            printf("Enter element %d,%d:",i+1,j+1);
21.            scanf("%d",&m1[i][j]);
22.        }
23. /*Displaying non diagonal elements*/
24.    for(i=0;i<r;i++)
25.        for(j=0;j<c;j++)
26.        {
27.            if(i!=j && i+j!=r-1)
28.            {
29.                printf("%d\t",m1[i][j]);
30.            }

```

Matrix

```

31.         }
32.         getch();
33.     }
6.6 WAP to sort all the elements of a matrix.
1. #include<stdio.h>
2. #include<conio.h>
3. void main()
4. {
5.     int m1[10][10],r,c,i,j,k,l,t,pos1,pos2;
6.     clrscr();
7.     printf("Enter row & column of matrix:");
8.     scanf("%d%d",&r,&c);
9.     /*input matrix*/
10.    for(i=0;i<r;i++)
11.        for(j=0;j<c;j++)
12.        {
13.            printf("Enter element %d,%d:",i+1,j+1);
14.            scanf("%d",&m1[i][j]);
15.        }
16.    /*Selection Sorting*/
17.    for(i=0;i<r;i++)
18.        for(j=0;j<c;j++)
19.        {
20.            pos1=i*c+j;
21.            for(k=0;k<r;k++)
22.                for(l=0;l<c;l++)
23.                {
24.                    pos2=k*c+l;
25.                    if(pos2>pos1 && m1[i][j]>m1[k][l])
26.                        t=m1[i][j];
27.                        m1[i][j]=m1[k][l];
28.                        m1[k][l]=t;
29.                }
30.            }
31.        }
32.    /*Output Matrix*/
33.    for(i=0;i<r;i++)
34.    {
35.        for(j=0;j<c;j++)
36.            printf("%d\t",m1[i][j]);
37.            printf("\n");
38.        }
39.        getch();
40.    }
41. }

```

Lab Excercise:-

1. WAP that will transpose a 2-d array but the array is stored as 1-d array.
2. WAP to extract the maximum and the minimum elements from a matrix.
3. WAP to double all the elements of the matrix.

What would be the output of the following programs?

1. char c[2] = "A";
printf("%c", c[0]);
printf("%s", c);
2. char s[] = "Get organized! learn c!!";
printf("\n%s", &s[2]);
printf("\n%s", s);
printf("\n%s", &s);
printf("\n%c", s[2]);
3. printf("%c", "matrix"[4]);
4. char str1[] = {'H', 'e', 'l', 'l', 'o', 0};
char str2[] = "Hello";
printf("\n%s", str1);
printf("\n%s", str2);
5. printf(5+"Good Morning");

Output:

1. A A
2. t organized! learn c!!
Get organized! learn c!!
Get organized! learn c!!
t

3. i
4. Hello
Hello
5. Morning.

Fill in the blanks:

1. "A" is a _____ whereas 'A' is a _____.
2. A string is terminated by a _____ which is written as _____.
3. The array char name[10] can consist of a maximum of _____ characters.
4. The array elements are always stored in _____ memory locations.

Answers:

- | | |
|---------------------|----------------------|
| 1. string character | 2. null character \0 |
| 3. 9 | consecutive |

Programs to be solved in class:-

1. WAP to calculate the length of a given string input through user.
2. WAP to convert a string from lowercase to uppercase.
3. WAP to copy a string into another string.
4. WAP to concatenate second string with first string.
5. WAP to concatenate two strings in a third string.

M@trix Computers

6. WAP to reverse a string.
7. WAP to check whether a given string is palindromme or not.
8. WAP that compares two strings & print the string which is greater.
9. WAP to count the number of words in a given line input through user. There may be more than one space between two words.
10. WAP to input a name as a string and print first letter of each word of that name.
11. WAP to search a word in a given sentence & print the position of first occurrence of that word in the given sentence. Input sentence & word from the user.
12. WAP to search a word in a given sentence & print the position of last occurrence of that word in the given sentence. Input sentence & word from the user.
13. WAP to search a word in a given sentence & print the position of second occurrence of that word in the given sentence. Input sentence & word from the user.
14. WAP to search a word in a given sentence & count the number of occurrence of that word in the given sentence. Input sentence & word from the user.
15. WAP to input a list of n names of students and print the list in ascending order.

Some solved Programs:-

- 7.1 Input a string through keyboard write a program to print a string in reverse order as per word: e.g. How Are You = You Are How.*/**

```

1. #include<stdio.h>
2. #include<conio.h>
3. #include<string.h>
4. main()
5. {
6.     char s1[50],t1[50],t2[50] ="";
7.     int i=0,j=0;
8.     clrscr();
9.     printf("Enter a string");
10.    gets(s1);
11.    strcat(s1," ");
12.    while(s1[i]!='\0')
13.    {
14.        if(s1[i] == ' ')
15.        {
16.            t1[j++] = ' ';
17.            t1[j] = '\0';
18.            strcat(t1,t2);
19.            strcpy(t2,t1);
20.            j = 0;
21.        }
22.        else
23.            t1[j++] = s1[i];
24.        i++;
25.    }
26.    puts(t1);
27.    getch();
28. }
```

- 7.2 WAP to input a text and replace an entered string occurring within the text with equal number of "", at all occurring.**

```

1. #include<stdio.h>
2. #include<conio.h>
```

kamal bhatia
3. main()
4. {
5. char ch;
6. clrscr();
7. printf("Enter text: ");
8. while((ch=getch())!='r')
9. printf("*");
10. getch();
11. }

7.3 Write a program that converts a string like "124" to an integer 124.

```
1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     char str[6];
6.     int num= 0, i;
7.     clrscr();
8.     printf("Enter a string containing a number:");
9.     scanf("%s",str);
10.    for(i=0; str[i]!='\0'; i++)
11.  {
12.        if(str[i]>=48 && str[i]<=57)
13.            num = num*10+(str[i]-48);
14.        else
15.        {
16.            printf("Not a valid string");
17.            getch();
18.            return;
19.        }
20.  }
21.  printf("\nThe numer is: %d\n",num);
22.  getch();
23. }
```

Note Function atoi also converts string to integer.

7.4 WAP that will print out all the relations of a string typed into it. for eg, the rotations of word " space " are: space paces acesp cespa espac*/

```
1. #include<stdio.h>
2. #include<conio.h>
3. void main()
4. {
5.     char s1[10],t;
6.     int i,j,n;
7.     clrscr();
8.     printf("Enter a string:");
9.     gets(s1);
10.    n=strlen(s1);
11.    for(i=1;i<=n;i++)
12.  {
13.        puts(s1);
14.        t=s1[0];
15.        for(j=1;s1[j]!='\0';j++)
16.            s1[j-1]=s1[j];
```

M@trix Computers

```

17.         s1[j-1]=t;
18.         s1[j]='\0';
19.     }
20.     getch();
21. }
```

- 7.5 WAP to enter any name and print the same as per the following format:**

input : shruti
output: s
 sh
 shr
 shru
 shrut
 shruti

```

1. #include<stdio.h>
2. #include<conio.h>
3. void main()
4. {
5.     char s1[10];
6.     int n,i,j;
7.     clrscr();
8.     printf("Enter a string:");
9.     gets(s1);
10.    n=strlen(s1);
11.    for(i=0;i<n;i++)
12.    {
13.        for(j=0;j<=i;j++)
14.            printf("%c",s1[j]);
15.        printf("\n");
16.    }
17.    getch();
18. }
```

- 7.6 WAP that replaces two or more or consecutive blanks in a string by a single blank. For e.g., if the input is: "matrix computers" The output should be :"matrix computers".**

```

1. #include<stdio.h>
2. #include<conio.h>
3. void main()
4. {
5.     char s1[50];
6.     int i=0,j=0;
7.     clrscr();
8.     printf("Enter a string");
9.     gets(s1);
10.    while(s1[i]!='\0')
11.    {
12.        s1[j++]=s1[i];
13.        if(s1[i]==32)
14.        {
15.            while(s1[i]==32)
16.                i++;
17.        }
    }
```

M@trix Computers

```

18.         .   else
19.                 i++;
20.     }
21.     s1[j]='\0';
22.     puts(s1);
23.     getch();
24. }
```

7.7 WAP to input a string and print the character which occurs the maximum number of times within the string.

```

1. #include<stdio.h>
2. #include<conio.h>
3. void main()
4. {
5.     char s1[50],ch,maxchar;
6.     int n,i,j,freq,maxfreq=0;
7.     clrscr();
8.     printf("Enter a string:");
9.     gets(s1);
10.    n=strlen(s1);
11.    for(i=0;i<n;i++)
12.    {
13.        if(s1[i]==0)
14.            continue;
15.        ch=s1[i];
16.        freq=1;
17.        for(j=i+1;j<n;j++)
18.        {
19.            if(ch==s1[j])
20.            {
21.                s1[j]=0;
22.                freq++;
23.            }
24.        }
25.        if(freq>maxfreq)
26.        {
27.            maxchar=ch;
28.            maxfreq=freq;
29.        }
30.    }
31.    printf("Character is %c & its frequency is %d",maxchar,maxfreq);
32.    getch();
33. }
```

Lab Exercise:-

1. WAP to copy a string into another in reverse order.
2. WAP to remove all the leading blanks in a string inputted by an user.
3. WAP to input a string and a character to be searched within string. Print the frequency of the character within the string.
4. WAP to input any string and print the frequency of each character within the string. The character with multiple frequencies should be displayed only once in the output, with the frequency value.

5. WAP to input any sentence and arrange the characters of each word in alphabetical order separately and then print the sentence. For eg , input: computer program in basic.
output: cemoprtu agmoprr in abcis.
6. WAP input a string from the user and then output the frequency of the vowels in the string.
7. A city hotel has 10 floors ranging from 1 to 10,each having 60 rooms. Make use of a single dimensional array for the name of the occupant and a 2-d array place(a, b), where a and b represent the floor and the room number respectively. WAP to allocate room for an occupant after inputting the name of the occupant and the room number.
8. WAP to input a text and print the word containing the maximum number of vowels.
9. WAP to store any five names in two different arrays. Read the arrays and print the list of common names of both the arrays.
10. WAP to read in a string and output the frequency of each word in that string.

What would be the output of the following program?

1.

int add(int);

main()
{

M@trix Computers

```

int i = 3, k, l ;
k = add (++i) ;
l = add (i++) ;
printf("i=%d k=%d l=%d", i, k, l) ;
}

int add(int ii)
{
    ++ii ;
    return(ii) ;
}

2.
int function(int, int);
main()
{
    int i = 135, a = 135, k ;
    k = function (!++i, !a++) ;
    printf("i=%d a=%d k=%d", i, a, k) ;
}

int function (int j, int b)
{
    int c ;
    c = j + b ;
    return (c) ;
}

3.
int func1(int);
main()
{
    int k = 35, z ;
    k = func1(k = func1(k = func1(k))) ;
    printf("k = %d", k) ;
}

int func1(int k)
{
    k++ ;
    return (k) ;
}

4.
int func(int);
int func1(int);
main()
{
    int k = 35, z ;
    z = func (k) ;
    printf("z = %d", z) ;
}

int func(int m)
{
    ++m ;
    return (m = func1 (++m)) ;
}

```

Matrix

```

int func1(int m)
{
    m++ ;
    return (m) ;
}

```

5.

```

int packman (int, int);
main()
{
    int p = 23, f = 24 ;
    packman (p, f) ;
    printf("p = %d f = %d", p, f) ;
}

int packman (int q, int h)
{
    q = q + q ;
    h = h + h ;
    return (q) ;
    return (h) ;
}

6.
int addsub(int, int);
main()
{
    int i = 10, j = 20, k ;
    k = addsub(i, j) ;
    printf("k = %d", k) ;
}

int addsub(int c, int d)
{
    int x, y ;
    x = c-d ;
    y = c+d ;
    return(x, y) ;
}

7.
int check(int);
main()
{
    int k = 35, z ;
    z = check(k) ;
    printf("z = %d", z) ;
}

int check(int m)
{
    if(m > 40)
        return(!m++) ;
    else
        return(!++m) ;
}

```

8.
int z = 4;
printf("%d", printf("%d %d", z, z));

9.
void display();
main()
{
 printf("\n welcome in C:");
 display();
}
void display()
{
 printf("History of C:");
 main();
}

10.
int circle(int);
main()
{
 float area;
 int radius = 1;
 area = circle (radius);
 printf("\n%f",area);
}
int circle(int r)
{

Output:

1. i = 5 k = 5 l = 5
2. i = 136 a = 136 k = 0
3. k = 38
4. z = 38
5. p = 23 f = 24
6. k = 30
7. 0
8. 4 4 3
9. Both message will get printed indefinitely.
10. 3.000000
11. 5 2
12. Declaring
- 13.

Point out the errors, if any, in the following programs:

1.
void fun();
main()
{

```
float a;
a = 3.14*r*r;
return(a);
}
```

11.
void hello(int, int);
main()
{
 int i = 5, j = 2,
 hello (i, j);
 printf("\n%d %d", i,j);
}
hello(int i, int j)
{
 i = i*i;
 j = j*j;
}

12.
When we mention the prototype of a function
we are defining or declaring the function?

What are the two notations of defining functions
commonly known as:

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

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

The first one is known as ANSI notation and the second is known as Kernighan and Ritchie or simply K&R notation.

```
int a=1;
while(a<=5)
{
    printf("%d",a);
```

```

        if( a>2)
            goto abc;
    }
}

void fun()
{
    abc:
    printf("Matrix computers");
}

2.
main()
{
    void matrix()
    {
        printf("Welcome in the world of
computer:");
    }
}

```

Output:

1. **Error.** goto cannot take control to a different function.
2. **Error.** One function can't be defined into the body of another function.
3. **Error.** The variable a is redeclared

```

    }

3.
void f(int a, int b)
{
    int a;
    a=20;
    return a;
}

4.
int f(int a)
{
    a>20?return(10):return(20);
}

```

4. **Error.** return statement cannot be used as shown with the conditional operator. Instead the following can be used:
return(a>20?10:20);

State whether the following statements are True or False:

1. The variables commonly used in C functions are available to all functions in a program.
2. To return the control back to the calling function we must use the keyword return. The same variable names can be used in different functions without any conflict.
3. Every called function must contain a return statement.
4. A function may contain more than one return statement.
5. Each return statement in a function may return a different value.
6. A function can still be useful even if you don't pass any arguments to it and the function doesn't return any value back.
7. Same names can be used for different functions without any conflict.
8. A function may be called more than once from any other function.
9. It is necessary for a function to return some value.
10. A function can have several declarations but only one definition.
11. A function cannot be defined inside another function.
12. In a function two return statements should never occur.
13. In a function two return statements should never occur successively.
14. In C all functions except main() can be called recursively.
15. Usually recursion works slower than loops
16. Too many recursive calls may result in stack overflow

Answers:

- | | | | |
|----------|-----------|----------|-----------|
| 1. False | 2. False | 3. True | 4. False |
| 5. True | 6. True | 7. True | 8. False |
| 9. False | 10. True | 11. True | 13. False |
| 14. True | 15. False | 16. True | 17. True |

Programs to be solved in class:-

1. Write a function which takes to integer as argument and return there sum. WAP to test this function.
2. Write a function which takes to integer as argument and return there average in float. WAP to test this function.
3. WAP that uses a function that converts a lowercase character to its uppercase.
4. WAP that uses a function that calculates factorial of a given number.
5. WAP that uses a function power that calculates the power of a given number.
6. WAP that uses a function that finds the largest of three integer quantities.
7. WAP that uses a function toupper which accepts a character argument and return its equivalent uppercase character.
8. WAP that uses a function isupper which should return a non-zero if the given number is a uppercase character and 0 if not.
9. WAP that uses a function isdigit which should return a non-zero if the given number is a digit and 0 if not.
10. WAP that uses a function isalpha which should return a non-zero if the given number is a alphabet and 0 if not.
11. WAP that uses a function isalnum which should return a non-zero if the given number is a alpha numeric and 0 if not.
12. WAP that uses a function that returns the gcd(greatest common divisor) of two integers.
13. WAP to use a function swap which interchanges two numbers using call by value.
14. WAP to use a function sort which sorts an array in ascending order.
15. WAP that uses a function to copy a string into another string.
16. WAP that uses a function to concatenate two strings.
17. WAP that uses a function to a string into uppercase.
18. WAP that uses a recursive function to calculate factorial of N.
19. WAP that uses a recursive function to multiply two numbers.
20. WAP that uses a recursive function to print the n^{th} term of a fibbonicci series.
21. WAP that uses a recursive function to print the fibbonicci series.
22. WAP that uses a recursive function to calculate sum of digits of a number.

Solved Programs:-

8.1 WAP to calculate the Greatest common divisor of two numbers using recursive function.

```

1. #include<stdio.h>
2. #include<conio.h>
3. int gcd(int, int);
4. main()
5. {
6.     int n,m,ans;
7.     clrscr();
8.     printf("\n Enter two integer numbers");
9.     scanf("%d %d",&n,&m);
10.    ans = gcd(n,m);
11.    printf("GCD of %d and %d is %d",n,m,ans);
12.    getch();
13. }
14. int gcd(int n, int m)
15. {
16.     if(n >= m && n%m == 0)
17.         return(m);
18.     else
19.         return gcd(m,n%m);
20. }
```

M(a)trix Computers

- 8.2 WAP to calculate the Greatest common divisor of two numbers using non recursive function.

```

1. #include<stdio.h>
2. #include<conio.h>
3. int gcd(int,int);
4. main()
5. {
6.     int n,m,ans;
7.     clrscr();
8.     printf("Enter two integers");
9.     scanf("%d %d",&n,&m);
10.    ans = gcd(n,m);
11.    printf("GCD of %d and %d is %d", n,m,ans);
12.    getch();
13. }
14. int gcd(int n, int m)
15. {
16.     int t;
17.     while(m!=0)
18.     {
19.         t = m;
20.         m = n % m;
21.         n = t;
22.     }
23.     return (n);
24. }
```

- 8.3 Write a program to search a substring.

```

1. #include<stdio.h>
2. #include<conio.h>
3. int strsearch(char [ ],char [ ]);
4. main()
5. {
6.     char s1[50],s2[10];
7.     int ans=0;
8.     clrscr();
9.     printf("Enter a string"); /*input of string */
10.    gets(s1);
11.    printf("Enter a string"); /*input of substring to search*/
12.    gets(s2);
13.    ans = strsearch(s1,s2); /*function call*/
14.    if(ans == -1) /*using the returned value*/
15.        printf("String is not found");
16.    else
17.        printf("String is found at pos %d ",ans+1);
18.    getch();
19. }
20. int strsearch(char s1[ ],char s2[ ]) /*definition*/
21. {
22.     int i,j;
23.     i=j=0;
24.     while(s1[i] != '\0')
```

```

25.    {
26.        if(s1[i] == s2[j])
27.        {
28.            while(s1[i+j] == s2[j] && s2[j] != '\0')
29.                j++;
30.            if(s2[j] == '\0')
31.                return (i),
32.            j=0;
33.        }
34.        i++;
35.    }
36.    return (-1);
37. }

```

8.4 WAP to calculate the determinants of a matrix

```

1. #include<stdio.h>
2. #include<conio.h>
3. #define MAXROW 4
4. #define MAXCOL 4
5. int det_mat(int [ ][MAXCOL],int,int);
6. main()
7. {
8.     int mat[MAXROW][MAXCOL]={0};
9.     int r,c,i,j,sum;
10.    clrscr();
11.    printf("Enter dimension of matrix");
12.    scanf("%d%d",&r,&c);
13.    /*Input Matrix */
14.    for(i=0;i<r;i++)
15.        for(j=0; j<c;j++)
16.        {
17.            printf("Enter element %d %d", i+1, j+1);
18.            scanf("%d",&mat[i][j]);
19.        }
20.    /*function call*/
21.    sum = det_mat(mat,r,c);
22.    /*output*/
23.    printf("%d",sum);
24.    getch();
25. }
26. int det_mat(int mat[ ][MAXCOL], int r, int c)
27. {
28.     int i,j,k,sign,sum,a;
29.     int mat2[MAXROW][MAXCOL]={0};
30.     sign = 1;
31.     sum = 0;
32.     if(c == 1)
33.         return(mat[0][0]);
34.     for(i=0 ; i<c; i++,sign *= -1)
35.     {
36.         for(j=1; j<r; j++)
37.         {
38.             for(k=0; k<c; k++)

```

```

39.         {
40.             if(k == i)
41.                 continue;
42.             if(k>i)      mat2[j-1][k-1]=mat[j][k];
43.             else        mat2[j-1][k]=mat[j][k];
44.         }
45.     }
46. }
47. sum = sum + mat[0][i] * sign * det_mat(mat2,r-1,c-1);
48. }
49. return (sum);
50. }
51. }
```

Lab Exercise:-

1. WAP that receives any year from the keyboard and uses a function to determine whether the year is a leap year or not.
2. WAP that uses a function which receives a float and an int from main(), finds the product of these two and returns the product which is printed through main().
3. WAP that uses a function to calculate the sum of n odd integers.
4. WAP that uses a function to calculate the sum of n even integers starting from a given even integer.
5. WAP that uses a function to determine whether a given positive integer is a prime number or not.
6. WAP that uses a function to determine whether a given positive integer is a fibonacci number or not.
7. WAP that uses a function that finds the length of the largest monotonically increasing subsequence in a sequence of real numbers.
8. WAP that uses a function for finding the absolute value of the integer parameter passed to it (do not use any library function).
9. WAP that uses a function that takes two arguments: a character and an integer and prints the character given number of times. if however the integer is missing the function prints the character twice.
10. WAP that uses a function to sum n natural numbers starting from a given number.
11. WAP that uses a function that takes a character argument and prints it number of times equal to number of times that function has been called to the input.
12. WAP that uses a function which takes a real number as its argument and returns the sum of digits(complete including fraction parts) of this number.
13. WAP that uses a function that checks whether the given number is divisible by 11 or not by using the algorithm which states that a number is divisible by 11 if and only if the difference of the sums of digits at odd positions and even positions is either zero or divisible by 11.
14. WAP that invokes a function satis() to find whether four integers a, b, c, d sent to satis() satisfy the equation $a^3+b^3+c^3=d^3$ or not. The function satis() returns 0 if the above equations satisfied with the given four numbers otherwise it returns -1.
15. WAP that uses a function called carea() to calculate area of a circle. The function carea() receives radius of float type and returns area of double type. the function main() gets radius value from the user, calls carea(), and displays the result. the function carea() is local to main().
16. WAP that uses various functions to sum following series:
 - a) $(1)+(1+2)+(1+2+3)+(1+2+3+4)+\dots$ upto n terms
 - b) $(2^2)+(2^2+4^2)+(2^2+4^2+6^2)+(2^2+4^2+6^2+8^2)+\dots$ upto n terms
 - c) $1+1/3+1/5+1/7+1/9+\dots$ upto n terms

17. d) $1 + 1/x + 1/(x^2) + 1/(x^3) + 1/(x^4) + \dots$ upto n terms
 WAP that receives a positive number from the keyboard and uses a function to obtain the prime factors of this number. for eg, prime factors of 24 are 2,2,2, and 3, whereas prime factors of 35 are 5 and 7.
18. WAP that uses a function that can compute sum of any geometric series.
19. WAP that uses a function that generates every third integer, beginning with i=1 and continuing for all integers that are less than 100. Calculate sum of those integers that are evenly divisible by 5.
20. WAP that uses a function for calculating volume and surface area of a sphere given diameter of the sphere.
21. WAP to print the largest element of an array using a function.
22. WAP that uses a function that takes a double array name and an array size as arguments and that swaps the first and last value in that array.
23. WAP that uses a function which will accept an array of integers as an argument. it should find and return the smallest element in the array after sorting it. the calling program requires the sorted array. The size of the array can be defined to be a global constant.
24. WAP that uses a function that receives an int array, its size and a character '+' or '-'. by default the character should be '+'. for the character '+', the function returns the sum of positive numbers stored in the array and for the character '-', the function returns the sum of negative numbers stored in the array.
25. WAP that reads a float array having 15 elements. the program uses a function reverse() to reverse this array. make suitable assumptions wherever required.
26. WAP that uses various functions to express the following algebraic formulas in a recursive form:
 a) $y = (x_1 + x_2 + x_3 + \dots + x_n)$
 b) $y = 1 - x + (x^2)/2 - (x^3)/6 + (x^4)/24 + \dots + (-1)^n x^n/n!$
 c) $p = (f_1 * f_2 * f_3 * \dots * f_t)$
27. WAP that uses a function that takes a decimal number as a parameter and returns its octal equivalent.
28. WAP that uses a function that takes a decimal number as a parameter and returns its hexadecimal equivalent.
29. WAP that uses a function that takes a six digit integer as a parameter. if the number is even, then adds up its digits else multiplies the individual digits and returns the result.
30. WAP that uses a function to send all -ve elements of an array to the end without altering the original sequence i.e. if array contains 5,-4,3,-2,6,-11,12,-8,9 then the return array will be 5,3,6,12,9,-4,-2,-11,-8
31. WAP that uses a function for the binary search algorithm without using recursion technique.
32. WAP that uses a function to print the length of a string.
33. WAP using following functions to reverse a string:
 a) func1() to reverse string using another array.
 b) func2() to reverse string without using another array.
34. WAP that uses a function to find the first occurrence of a string into another string.
35. WAP that uses a function to find the last occurrence of a string into another string.
36. WAP that uses a function to compare two strings and returns 0 if the strings are equal and -1 if the strings are unequal.
37. WAP that uses two functions, code() and decode(), that accepts a string for an argument. the code() function should modify the argument string adding 1 to all characters in it except the null terminator. the decode() function restores the coded string to its original form. this program should accept a string on the command line, print the string coded, and then print it decoded. if no string is specified on the command line, prompt for none.
38. WAP that uses a function that accepts two strings as the arguments and compares them to find the length of the greatest common substring between the two.
39. WAP that uses a function for analyzing a line of text by examining each of the characters and determining into which of several different categories it falls. In particular, we want to count the number of vowels, constants, digits, white space character and other characters.

40. WAP that uses a function that will read characters in a character type array and write the characters backwards into another character array. assume that text contains 80 characters.
41. WAP that uses a function to sort all the elements in an array between the position lb and ub (lb is the lower bound and ub is the upper bound)
42. WAP that uses a function that takes a string as a parameter and returns the frequency, of each character, in that string.
43. WAP that uses a function that takes two strings consisting of maximum 80 characters as parameters. examine both these strings and remove all the common characters from both these strings display the resultant string in the main function.
44. WAP that uses a function that takes two strings as parameter and return the position of the first occurrence of the first string in the main string and null if not present.
45. WAP that uses function that takes a string as a parameter and replace one or more blank between words by a single blank.
46. WAP that uses function that takes a string as a parameter and set the string such that every sentence should start with an upper case character.
47. WAP that uses function that takes two strings as parameters and compares these strings lexicographically. the function should return -1 or 0 or 1 depending on whether str1 is lexicographically "less than" or "equal to" or "greater than" str2.
48. WAP that uses a function to search for a given word in a dictionary. The dictionary is a lexicographically sorted array of characters strings. use binary search method. you can make use of the standard c library function to compare two strings.
49. WAP that uses a function that will accept a set os strings and output them in increasing order of lengths of the strings and sorted alphabetically.
50. WAP that uses a function that takes a string and replace all occurrence of the string "and" in the text by "or" and the modified text should be print by main() function.
51. WAP that uses a function that will accept a set of names separated by newline and check whether they are written properly. a name should begin with an upper case alphabet, following which each string in a name should begin with an upper case letter. the only non-alphabetic character allowed in a name are ":" And '-' (period and hyphen).
52. WAP that uses that recognizes whether a telephone number is valid or not by checking for the following criterion. a telephone number is a 10 digit string whose first digit is a '0' followed by 2 digits lying within 1-4.the remaining positions may be occupied by any digit from 0-9,other than the 4th position which can be occupied by any digit from 1-9.
53. WAP that uses a function called pr_rev() that reads a string input from the keyboard and prints it in reverse. for example, hello would be printed olleh.
54. WAP that uses a recursive function called print_num() that has one integer argument. it will print the number from 1 to n on the screen, where n is the value of the argument.
55. WAP that uses a function that calculates factorial of a given number using recursion method.
56. WAP that uses a function that calculates multiplication of two given numbers using recursion method.
57. WAP that uses a function that prints the nth element of fibonnacci series using recursion method.
58. WAP that uses a function that prints the entered string in reverse order using recursion method.
59. WAP to solve the tower of hanoi problem.
60. WAP to sort the array using quick sort using recursion technique.
61. WAP that uses a recursive function to convert given decimal number into its binary equivalent.
62. WAP that uses a function for the binary search algorithm using recursion technique.
63. WAP that uses a function to find the determinant value of a matrix.
64. A positive number is entered through the keyboard. Write a function to obtain the prime factors of this number. For e.g., prime factors of 24 are 2,2,2 and 3,whereas prime factors of 35 are 5 and 7.
65. A 5 digit positive integer is entered through the keyboard, write functions to calculate sum of digits of the 5-digit number:
- 1) without using recursion
 - 2) using recursion

66. WAP to use the suitable function to obtain the prime factors recursively.
67. WAP to use the suitable function to generate the first n terms of the fibonnacci sequence recursively.
68. WAP to use the suitable function to find the binary equivalent of a given decimal integer and display it.
69. WAP to use a function that compute the binomial coefficient $n!/(k!(n-k)!)$
70. WAP to use the function that compute the distance between two points and use it to develop a function that will compute the area of the triangle whose vertices are a(x₁,y₁), b(x₂,y₂),and c(x₃,y₃).use these function to develop a program which returns a value 1 if the point (x,y) lies inside the triangle abc, otherwise a value 0.
71. Given three variables x,y,z write a function to circularly shift their values to right. in other words if x=5,y=8,z=10 after circular shift y=5,z=8 and x=10.call the function with variables a,b,c to circularly shift their values.

M@trix Computers

What would be the output of the following program:

1. char far *s1,*s2;
printf("%d%d",sizeof(s1),sizeof(s2));
2. Are the expressions $*p++$ and $++*p$ same?
3. Can there be another statement which does the same job as $++*p$?
4. What would be the equivalent pointer expression for referring the same element as $a[i][j][k][l]$?
5. What would be the output of the following program:
int a[]={12, 13, 14, 15, 16};
printf("%d %d %d", sizeof(a), sizeof(*a),
sizeof(a[0]));
6. What will be the output of the following program assuming that the array begins at location 1002.
int a[3][4]={1, 2, 3, 4,
5, 6, 7, 8,
9, 10, 11, 12};
printf("%u %u %u", a[0]+1, *(a[0]+1),
*(a+0)+1);
7. In the following program how will you print 50 using p?
main()
{
 int a[]={10, 20, 30, 40, 50};
 char *p;
 p=(char*)a;
}
8. In the following program add a statement in the function fun() such that address of a gets stored in j
void fun(int **);
main()
{
 int *j;
 fun(&j);
}
void fun(int **k)
{
 int a=10;
 /*add statement here*/
}

9.

How will you declare an array of three function pointers where each function receives two *ints* and returns a *float*?

10.

Would the following program give a compilation error or warning?

```
float i=10, *j; (point type pointer)
void *k;
k=&j;
j=k;
printf("%f", *j);
```

11.

Would the following program compile?

```
main()
{
    float i=10, *j;
    void *k;
    j=k=&a;
    j++;
    k++;
    printf("%u %u", j, k);
```

12.

Would the following program code compile successfully?

```
printf("%c", 7["Computer"]);
```

13.

What is a null pointer?

14.

Is the NULL pointer same as the uninitialised pointer?

15.

In which header file is the NULL macro defined?

16.

What is the difference between a null pointer, a NULL macro, the ASCII NUL character and a null string?

17.

What will be the output of the following program?

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

18.

```
int a[ ] = {10,20,30,40,50};
int *j;
```

M@trix Computers

j = a;
j = j+3. (This means 3 steps)
printf("\n%d", j); does print of j.

19.

float a[] = {3, 24, 15, 2, 5, 3, 5, 4, 5, 5, 5};

float *j, *k.

j = a;

k = a+4.

j = j*2;

k = k/2;

printf("\n%d %d", *j, *k);

20.

What will be the output of the following program?

#include<stdio.h>

main()

```
{     printf("%d %d", sizeof(NULL), sizeof("")); }
```

21.

How many bytes are occupied by near, far and huge pointers?

22.

What would be the output of the following program:

```
char *f();  
main()  
{     char *s;  
      s=f();  
      printf("%s", s); }
```

char * f()

```
{     char string[30];  
      strcpy(string, "Matrix Computers");  
      return(string); }
```

23.

What is the solution of the above problem?

24. Does there exist some other solution of the above problem?

25.

Would the following code work all the time: *No*

main()

```
{     char *p;  
      gets(p);  
      printf("%s", p); }
```

26.

The following code is improper though it may work sometimes. How would you improve it:

main()

{

```
(char *p1= "Matrix",  
char *p2= "Computers",  
strcat(p1,p2);  
printf("%s", p1);
```

}

27.

What would be the output of the second `printf()` in the following program.

#include<alloc.h>

main()

{

```
int *p;  
p=(int*)malloc(20);  
printf("%u", p); /* suppose it prints 1314 */  
free(p);  
printf("%u", p);
```

}

28.

To `free()` we only pass the pointer to the block of memory which we want to deallocate. Then how does `free()` know how many bytes it should deallocate?

29.

What would be the output of the following program:

#include<alloc.h>

main()

{

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

}

30.

What is the difference between `malloc()` and `calloc()` functions?

31.

How much maximum memory can be allocated in a single call to `malloc()`?

32.

What should be the output of the following program?

main()

{

```
char a[] = "Matrix Computers";  
char *b= "Matrix Computers";  
printf("%d %d", sizeof(a), sizeof(b));  
printf("\n%d %d", sizeof(*a), sizeof(*b));
```

}

33.

M@trix Computers

For the following statements would a[3]nd p[3] fetch the same character?

```
char a[ ]= "Matrix";
char *p= "Matrix";
```

34. When are *char a[]* and *char *a* treated as same by the compiler?

35. Would the program compile successfully:

```
main()
{
    char a[ ]= "Matrix";
    char *p= "Computers";
    a= "Computers";
    p= "Matrix";
    printf("%s %s", a, p);
}
```

36. What does the following declaration mean:
*int (*p)[10];*

37. What will be the output of the following program:

```
main()
{
    char *s[ ]={"Frogs", "Do", "Not", "Die",
    "They", "Croak!"};
    printf("%d %d", sizeof(s), sizeof(s[0]));
}
```

38. If int s[5] is a one-dimensional array of integers, which of the following refers to third element in the array?

- a) *(s + 2)
- b) *(s + 3)
- c) s + 3
- d) s + 2

39.

```
int k = 35, *z, *y;
z = &k; /* suppose address of k is 1008*/
y = z;
*z++ = *y++;
k++;
printf("k = %d z = %d y = %D", k, z, y);
```

40.

```
void junk(int, int *);
main()
```

```
{
    int i = -5, j = -2;
    junk(i, &j);
    printf("i = %d j = %d", i, j);
```

```
}
```

```
void junk(int i, int *j)
{
```

```
i = i * i,
*j = *j * *j;
```

}

41.

```
float a = 7.999999;
float *b, *c;
b = &a;
c = b;
printf("%d %d %d\n", &a, b, c);
printf("%d %d %d\n", a, *(&a), *b, *c);
```

42.

```
float *jamboree();
main()
{
    float p = 23.5, *q;
    q = &p;
    printf("q before call = %d\n", q);
    q = jamboree();
    printf("q after call = %d\n", q);
```

43.

```
float *jamboree(float *r)
```

```
{
    r = r + 1;
    return(r);
```

44.

```
main()
{
    int x = 5, y = 2;
    hello(&x, &y);
    printf("\n%d %d", x, y);
```

45.

```
hello(int *i, int *j)
```

```
{
    *i = *i * *i;
    *j = *j * *j;
```

46.

```
void change(int *, int);
```

main()

```
{
    int a[ ] = {2, 6, 4, 8, 10};
    int i;
    change(a, 5);
    for(i = 0; i <= 4; i++)
        printf("%d ", a[i]);
```

47.

```
void change(int *b, int n)
```

```
{
    int i;
    for(i = 0; i < n; i++)
        *(b+i) = *(b+i)+5;
```

M@trix Computers

45.

```
int b[ ] = {10,20,30,40,50};
int i;
for (i = 0; i<=4; i++)
    printf("%d ",*(b+i))
```

46.

```
int b[ ] = {10,20,30,40,50};
int i, *k;
k = b;
for (i = 0; i<= 4; i++)
{
    printf("%d ",*k);
    k++;
}
```

47.

```
static int b[] = {10, 20, 30, 40, 50} ;
int i,*k ;
k = &b[4]-4 ;
for(i = 0 ; i <= 4 ; i++)
{
    printf("%d",*k) ;
    k++ ;
}
```

48.

```
static int a[] = {2, 4, 6, 8, 10,} ;
int i ;
for(i = 0 ; i <= 4 ; i++)
{
    *(a + i) = a[i] + i[a] ;
    printf("%d",*(i + a)) ;
}
```

49.

```
void f(int, int *);
main()
{
    static int a[5] = {2, 4, 6, 8, 10} ;
    int i, b = 5 ;
    for (i = 0 ; i < 5 ; i++)
    {
        f(a[i], &b) ;
        printf("%d %d\n", a[i], b) ;
    }
    void f(int x, int *y)
    {
```

x = *(y) + = 2 ;

}

50.

```
int arr[] = {0, 1, 2, 3, 4} ;
int i,*ptr ;
for(ptr = &arr[0] ; ptr <= &arr[4] ; ptr++)
    printf("%d",*ptr) ;
```

51.

```
int arr[] = {0, 1, 2, 3, 4} ;
int i,*ptr ;
for(ptr = &arr[0], i = 0 ; i <= 4 ; i++)
    printf("%d", ptr[i]) ;
```

52.

```
int arr[] = {0, 1, 2, 3, 4} ;
int i,*p ;
for(p = arr, i = 0 ; p + i <= arr + 4 ; p++, i++)
    printf("%d",*(p + i)) ;
```

53.1

```
int arr[] = {0, 1, 2, 3, 4} ;
int i,*ptr ;
for(ptr = arr + 4, i = 0 ; i <= 4 ; i++)
    printf("%d",ptr[-i]) ;
```

54.

```
int arr[] = {0, 1, 2, 3, 4} ;
int *ptr, i ;
for(ptr = arr + 4 ; ptr >= arr ; ptr--)
    printf("%d", arr[ptr-arr]) ;
```

55.

```
static int a[] = {0, 1, 2, 3, 4} ;
static int *p[] = {a, a + 1, a + 2, a + 3, a + 4} ;
int **ptr ;
ptr = p ;
**ptr++ ;
printf("%d %d %d\n", ptr-p, *ptr-a, **ptr) ;
*++*ptr ;
printf("%d %d %d\n", ptr-p, *ptr-a, **ptr) ;
++**ptr ;
printf("%d %d %d\n", ptr-p, *ptr-a, **ptr) ;
```

M@trix Computers

Output:

1. 4 2 ** p++ increments the pointer and not the value pointed by it, whereas ++p increments the value being pointed by p.*
2. No. here no typecasting is required while assigning the value to and from k because conversions are applied automatically when other pointer types are assigned to void *.
3. $(^p)^{++}$
4. $^{*}(^{*}(^{*}(a+i)+j)+k)+l)$
5. 10 2 2
6. 1004 2 2
7. `printf("%d", ^((int*)p+4));`
8. `*k=&a;`
9. `float(*a[3])(int,int);`
10. No. An error will be reported in the statement `k++` since arithmetic on void pointers is not permitted unless the void pointer is appropriately typecasted.
11. Yes it will print r of Computer
12. For each pointer type C defines a special pointer value which is guaranteed not to point to any object or function of that type. Usually, a null pointer constant is used for representing a null pointer is the integer 0.
13. In files `<stdio.h>`
14. A null pointer is a pointer which does not point anywhere. A NULL macro is used to represent the null pointer in source code. It has a value 0 associated with it. The ASCII NUL character has all its bits as 0 but doesn't have any relation with null pointer. The null string is just another name for an empty string "".
15. 5
16. Error. Because multiplication and division are not allowed on pointers.
17. 40
18. The near pointer is 2 bytes long and the far and huge pointers are 4 bytes long.
19. The output is unpredictable since string is an auto type of array and would die when the control goes back to `main()`. Thus s would be pointing to an array which no longer exists.
20. `char *f();`
21. `main()`

- Matrix Computers*
22. `char *s;
s=f();
printf("%s",s);`
 23. `char * f()
{
 static char string[30];
 strcpy(string, "Matrix
Computers");
 return(string);
}`
 24. `char *f();
main()
{
 char *s;
 s=f();
 printf("%s",s);
 free(s);
}`
 25. `char *p;
p=(char*)malloc(30);
strcpy(p, "Matrix Computers");
return(p);
}`
 26. `No, since p is an uninitialised pointer it must be pointing at some unknown location in memory. The string that we type would get stored at the location to which p is pointing thereby overwriting whatever is present at that location.`
 27. `main()
{
 char p1[25]= "Matrix";
 char *p2= "Computers";
 strcat(p1,p2);
 printf("%s",p1);
}`
 28. `1314`
 29. `In most implementations of malloc() the number of bytes allocated is stored adjacent to the allocated block. Hence it is simple for free() to know how many bytes to deallocate.`
 30. `2`
 - As against `malloc()`, `calloc()` needs two arguments, the number of elements to be allocated and the size of each element. For example,

M@trix Computers

- p=(int*) calloc(10, sizeof(int));
would allocate space for a 10 integer
array. Additionally, `calloc()` would
initialize each element with 0.
31. 64 KB.
32. 17 2
1 1
33. Yes
34. When using them as formal parameters
while defining a function.
35. No, because we may assign a new
string to a pointer but not to an array.
36. p is a pointer to an array of 10 integers.
37. 12 2
38. a) *(s +2)
39. k = 36 z = 1010 y = 1010
40. i = -5 j = 4
41. 4200 4200 4200
0 24576 -3 16415
42. q before call = 5498
q after call = 5502
43. 25 4

44. 7 11 9 13 15
10 20 30 40 50
45. 10 20 30 40 50
46. 10 20 30 40 50
47. 10 20 30 40 50
48. 4 8 12 16 20
49. 2 7
4 9
6 11
8 13
10 15
50. 0 1 2 3 4
51. 0 1 2 3 4
52. 0 2 4
53. 4 3 2 1 0
54. 4 3 2 1 0
55. 111
122
123

Some Solved Problems:-

9.1 WAP to extract a substring from a string

```

1. #include<stdio.h>
2. #include<conio.h>
3. #include<string.h>
4. main()
5. {
6.     char str[20], news[20];
7.     char *s, *t;
8.     int pos, n, i;
9.     clrscr();
10.    printf("Enter the string:");
11.    scanf("%s",str);
12.    printf("Enter the position and number of characters to extract:");
13.    scanf("%d%d",&pos,&n);
14.    s=str;
15.    t=news;
16.    if(n==0)
17.        n=strlen(str);
18.    s=s+pos-1;
19.    for(i=0;i<n;i++)
20.    {
21.        *t=*s;
22.        s++;
23.        t++;
24.    }
25.    *t='\0';
26.    printf("The substring is: %s\n",news);
27.    getch();
28. }
```

9.2 Write a program to sort all the elements of a 4x4 matrix.

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     int mat[4][4], *arr, i, j, k, t;
6.     clrscr();
7.     printf("Enter the elements of 4x4 matrix:");
8.     /*matrix input*/
9.     for(i=0; i<4; i++)
10.    {
11.        for(j=0; j<4; j++)
12.        {
13.            printf("Enter element %d %d", i+1, j+1);
14.            scanf("%d", &mat[i][j]);
15.        }
16.    }
17.    /*sort the elements of the matrix*/
18.    arr = mat; /*Base address of the matrix array*/
19.
20.    /* Print the matrix as entered using pointer*/
21.    printf("\nThe matrix formed is...\n");
22.    for(i=0; i<15; i++)/*number of passes*/
23.    {
24.        for(j=i+1; j<16; j++)
25.        {
26.            if(*(arr + i) > *(arr + j))
27.            {
28.                t = *(arr + j),
29.                *(arr + i) = *(arr + j);
30.                *(arr + j) = t;
31.            }
32.        }
33.    }
34.    /*print the sorted matrix*/
35.    printf("\n The sorted matrix is:\n");
36.    for(i=0; i<4; i++)
37.    {
38.        for(j=0; j<4; j++)
39.            printf("%d", mat[i][j]);
40.            printf("\n");
41.    }
42.    printf("\n\nPress any key to exit... ");
43.    getch();
44. }
```

9.3. WAP to input any string and delete the extra blanks spaces present in the same.

```

1. #include<stdio.h>
2. #include<conio.h>
3. #include<string.h>
4. main()
5. {
```

M@trix Computers

```

6.     static char s1[50];
7.     char s2[50];
8.     char *s, *t;
9.     int i, l;
10.    printf("Enter a string");
11.    gets(s1);
12.    s=s1;
13.    t=s2;
14.    l=strlen(s);
15.    for(i=0;i<=l-1;i++)
16.    {
17.        if(*s==' ')/*check for a blank*/
18.        {
19.            if(*(s+1)!=' ')
20.            {
21.                *t=*s;
22.                t++;
23.            }
24.            s++;
25.        }
26.        else
27.        {
28.            *t=*s;
29.            t++;
30.            s++;
31.        }
32.    }
33.    *t='\0';
34.    printf("Original Statement:%s\n",s1);
35.    printf("Modified Statement:%s\n",s2);
36.    getch();
37. }
```

9.4 Write a program to concatenate 2 strings using pointers. Donot use strcat function.

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     char s1[20],s2[20],*p1,*p2;
6.     /* Input 1st string*/
7.     printf("Enter 1st string");
8.     gets(s1);
9.     /*Input 2nd string*/
10.    printf("Enter 2nd string");
11.    gets(s2);
12.    /*Assigning address of strings in two pointers*/
13.    p1 = s1;
14.    p2 = s2;
15.    /*Moving pointer 1 to the end of 1st string*/
16.    while(*p1 != '\0')
17.        p1++;
18.    /* Adding 2nd string to the end of 1st string*/
```

M@trix Computers

```

19.     while(*p2 != '\0')
20.         *p1++ = *p2++;
21.     /* Adding NULL to the end of string*/
22.     *p1 = '\0';
23.     /*Output string after concatenation*/
24.     puts(s1);
25.     getch();
26. }
```

- 9.5 WAP that receives the month and year from the keyboard as integers and prints the calendar in the following format.**

```

1. #include<stdio.h>
2. #include<conio.h>
3. static char *months[ ]={ "January",
4.                         "February",
5.                         "March",
6.                         "April",
7.                         "May",
8.                         "June",
9.                         "July",
10.                        "August",
11.                        "September",
12.                        "October",
13.                        "November",
14.                        "December"
15. };
16. main()
17. {
18.     static int days[12]={31,28,31,30,31,30,31,31,30,31,30,31};
19.     long int ndays, ldays, tdays, fdays;
20.     int d, i, m, fday, y;
21.     char ch;
22.     clrscr();
23.     printf("Enter year(1900 onwards)& month (number):");
24.     scanf("%d %d",&y,&m);
25.     ndays = (y-1)*365;
26.     ldays = (y-1)/4-(y-1)/100+(y-1)/400;
27.     tdays = ndays+ldays;
28.     if((y%100==0&&y%400)||((y%4==0&&y%100!=0))
29.         days[1]=29;
30.     else
31.         days[1]=28;
32.     d= days[m-1];
33.     tdays = 0;
34.     for(i=0; i<m-2; i++)
35.         tdays = tdays + days[i];
36.     tdays = tdays + tdays;
37.     fday = tdays%7;
38.     cal(y, m, fday, d);
39.     getch();
40. }
41. cal(int yr, int mo, int fd, int da)
```

```

43. {
44.     int i, r, c;
45.     char a;
46.     clrscr();
47.     gotoxy(25,2);
48.     printf("%s %d",months[mo-1],yr);
49.     gotoxy(5,5);
50.     printf("-----");
51.     gotoxy(10,6);
52.     printf("Mon Tue Wed Thu Fri Sat Sun");
53.     gotoxy(5,7);
54.     printf("-----");
55.     r = 9;
56.     c=11+6*fd;
57.     for(i=0; i<=da; i++)
58.     {
59.         gotoxy(c,r);
60.         printf("%d",i);
61.         if(c<=41)
62.             c=c+6;
63.         else
64.         {
65.             c=11;
66.             r=r+1;
67.         }
68.     }
69.     gotoxy(5,15);
70.     printf("-----");
71.     printf("\n\n\n\n\nPress any key to exit... ");
72.     getch();
73. }
```

Matrix

9.6 Write a program that will read a line and delete from it all occurrences of the word 'the'.

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     char str[80], str2[80];
6.     char *s,*q,*p;
7.     int i;
8.     clrscr();
9.     printf("\nEnter a sentence not more than 80 chars long:\n");
10.    gets(str);
11.    s = str; /*Base address of the string*/
12.    p = str2; /*Base address of new string*/
13.    while(*s)
14.    {
15.        q = s;
16.        if(*s=='t'|| *s=='T')
17.        {
18.            s++;
19.            if(*s=='h')
```

```
20.    {
21.        s++;
22.        if(*s=='e')
23.            ;
24.        else
25.        {
26.            for(i=0; i<=2;i++)
27.                *p++=*q++;
28.        }
29.    }
30.    else
31.    {
32.        *p++=*q++;
33.        s--;
34.    }
35. }
36. else
37. {
38.     *p++=*s;
39.     s++;
40. }
41. *p='\0';
42. printf("\n\nSentence after deleting all occurrences of 'the' is:\n");
43. puts(str2);
44. getch();
```

Matrix

What would be the output of the following programs?

1.

```
main()
{
    struct message
    {
        int n;
        char mess1[50];
        char mess2[50];
    }m;
    m.n = 1;
    strcpy(m.mess1,"you can win");
    strcpy( m.mess2, "If you believe");

    /*assume that the structure is located at
address 2005*/
    printf("\n%u %u
%u",&m.n,m.mess1,m.mess2);
}
```

Matrix

2.

```
main()
{
    union a
    {
        int i;
        char ch[2];
    };
    union a z=512;
    printf("%d %d",z.ch[0],z.ch[1]);
}
```

3.

```
main()
{
    struct emp
    {
        char name[20];
        int age;
        float sal;
    }
```

};

```
struct emp e={"Matrix"};
printf("%d %f ",e.age,e.sal);
}
```

4.

```
struct emp
{
    char name[20];
    int age;
};

fun (int aa)
{
    int bb;
    bb=aa*aa;
    return(bb);
}

main()
{
    int a;
    a=fun(20);
    printf("\n%d",a);
}
```

5.

```
void f(struct emp);
struct emp
{
    char name[20];
    int age;
};

main()
{
    struct emp e= {"matrix", 30};
    f(e);
}

void f (struct emp ee)
{
    printf("%s %d",ee.name,ee.age);
}
```

}

6.

What is the similarity between structure, union and enum?

7.

Would the following declaration work:

```
typedef struct s
{
    int a;
    float b;
}s;
```

8.

Can a structure contain a pointer to itself:

9.

Point out the error if any in the following code

```
typedef struct
{
    int data;
    NODEPTR link;
}*NODEPTR;
```

10.

How will you eliminate the above problem?

11.

Output:

1. 2005 2007 2057
2. 0 2 binary of 512 is(00000010 00000000)
3. 0.000000 When an automatic structure is partially initialized, the remaining elements of the structure are initialized to 0.
4. Error. The semicolon at the end of the structure definition is missing, the compiler believes that the fun() would return something of the type struct emp, where as in reality it is attempting to return an int. this causes a mismatch, hence an error results.
5. Error occurs which can be rectified by declaring the structure before the declaration of the function f()
6. All of them let us define a new data type.
7. Yes
8. Yes, such structures are known as self referential structures.

What would be the output of the following program?

```
main()
{
    struct emp
    {
        char *n;
        int age;
    };
    struct emp e1= { "David", 23};
    struct emp e2=e1;
    strupr(e2.n);
    printf("%s",e1.n);
}
```

12.

Point out the error if any in the following code:

```
main()
{
    struct emp
    {
        char n[20];
        int age;
    };
    struct emp e1= { "David", 23};
    struct emp e2=e1;
    if(e1==e2)
        printf("The structures are equal");
}
```

9. A **typedef** defines a new name for a type, in this case however the error is that a **typedef** cannot be used until it is defined. in the given code fragment the **typedef** declaration is not yet defined at the point where the link field is declared.

10. To fix this problem, first a name ("struct node") must be given to the structure. Then declare the link field as a simple **struct node *** as shown below:

```
typedef struct node
{
    int data;
    NODEPTR link;
}*NODEPTR;
Another way to eliminate the problem is to disentangle the typedef declaration from the structure definition shown below:
struct node
{
    int data;
```

```

NODEPTR link;
};

typedef struct node *NODEPTR;
Yet another way to eliminate the
problem is to precede the structure
declaration with a typedef, in which case
you should use the NODEPTR typedef
when declaring the link field as shown
below:
typedef struct node *NODEPTR;
struct node
{
    int data;
    NODEPTR link;
};

In this case, you declare a new typedef
name involving struct node even though
struct node has not been completely
defined yet; this you are allowed to do.

```

11. David
 Structures cannot be compared using the built in `= =` and `!=` operators. This is because there is no single, good way for a compiler to implement structure comparison. A single byte by byte comparison the bits present in unused paddings in the structure (such padding is used to keep the alignment of later fields correct). A field by field comparison might require unacceptable amounts of repetitive code for large structures. Also, any compiler generated comparison could not be expected to compare pointer fields appropriately in all cases; for example, it's often appropriate to compare `char *` fields with `strcmp()` rather than with `= =`.

Some Solved Problems:-

- 10.1 Write a program that compares two given dates. To store a date use a structure that contains three members namely date, month and year. If the dates are equal then display the message as "Equal" otherwise "Unequal".

```

1. #include<stdio.h>
2. #include<conio.h>
3. struct date
4. {
5.     int day, month, year;
6. };
7. int check_date(struct date *dt)
8. main()
9. {
10.     int chkdt;
11.     /*The dates to be compared*/
12.     struct date d1, d2;
13.     /*input the dates to be compared*/
14.     printf("\nEnter the dates to be compared:");
15.     chkdt = check_date(&d1);
16.     if(chkdt == 0)
17.         exit();
18.     fflush(stdin);
19.     chkdt = check_date(&d2);
20.     if(chkdt == 0)
21.         exit();
22.     /* Compare the two structures*/
23.     if((d1.day == d2.day)&& (d1.month == d2.month)&&(d1.year == d2.year))
24.         printf("\nDate are Equal");
25.     else
26.         printf("\nDate are Unequal");
27.     getch();
28. }
29. /*Function to check the date entered*/

```

M@trix Computers

```

30. int check_date(struct date *dt)
31. {
32.     printf("\nEnter date(dd)");
33.     scanf("%d",&dt->day);
34.     printf("\nEnter month(mm)");
35.     scanf("%d",&dt->month);
36.     printf("\nEnter date(yyyy)");
37.     scanf("%d",&dt->year);
38.     if((dt->day >31 || dt->day<0) || (dt->month >12 || dt->month <0) ||
39.         (dt->year > 9999 || dt->year < 1000))
40.     {
41.         printf("\nImproper date entered");
42.         return(0);
43.     }
44.     else
45.         return(1);
46. }

```

Lab Exercise:-

- WAP to store information of 10 employees and to display information of an employee depending upon the employee number input from the user.
- WAP to accept and print a student's result using a structure.
- WAP that uses a function that takes two date (day, month, year) structure objects as arguments and returns the structure with later date.
- WAP that uses a structure called point(x, y co-ordinates) to model a point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point. Interaction with the program might look like this: Enter co-ordinates for p1: 2 3 enter coordinates for p2: 6 8 coordinates of p1+p2 are: 8 11
- WAP to add given number of days to a given date. Make use of structures wherever possible.
- Give necessary declarations for an array of 20 voter records, each record of which consists of four data values viz. Id-no, name, address, age. Make use of above declarations to write a program segment that prints id-no., name for all those voters whose age exceeds 60. (Assuming suitable data types.)
- Declare a structure to represent a complex number (a number having a real part and imaginary part).write a c program to add two complex numbers.
- Declare a structure to represent a complex number (a number having a real part and imaginary part).write a c program to subtract two complex numbers.
- Declare a structure to represent a complex number (a number having a real part and imaginary part).write a c program to multiply two complex numbers.
- Declare a structure to represent a complex number (a number having a real part and imaginary part).write a c program to divide two complex numbers.
- WAP to record score of a cricket match. one array stores information of batting team such as batsman's name, run scored, indication if out mode by which out along with total runs, overs played, total overs and extras. The other array stores information about bowling team such as bowler's name, overs bowled, maiden overs, runs given and wickets taken. The program reads in the above information and depending upon the user's choice, it displays either the batting team's information or the bowling team's information.
- WAP to prepare the invoice from the following data: customer number, customer name and address, data of sale, description, quantity, unit price, discount percentage, sales tax percentage.
- WAP to prepare and print payroll (payslip) of a group of employees for a particular month of the year. the employee information contains the following items: name and designation of employee, basic pay(bp),special pay(sp), contribution to general provident fund(pf),contribution to group scheme(gis),income tax deduction(it),city compensatory allowance(cca)= rs. 250,dearness allowance(da)=114% for basic pay < rs. 3500 85% for basic pay > 3500 and < 6000 74% for basic pay > 6000 house rent allowance(hra)=rs. 250.00 for basic pay < rs. 1500 rs. 450.00 for basic pay > 1499 and 2800 rs. 800.00 for basic pay > 2799 and < 3500 rs. 1000.00 for basic > 3499. The program computes the above quantities, gross pay, total deductions net pay and prints in a specified format. (hint: gross=bp+sp+hra+da+cca net=gross-deductions(i.e.,pf+gis+it) make use of structures and arrays in the program.
- WAP to calculate income tax of a group of employee from the following data. Total income, life insurance premiums (lic),unit-linked insurance plan (ulip),provident fund(pf),post-office cumulative time deposit(ctd), national saving certificates(nsc) Assume the following norms for the calculation of income tax: a tax total income slab rates of income tax
- | | |
|---------------------|-----|
| upto 35000 | nil |
| from 35001 to 60000 | 20% |

from 60001 to 120000 30%
 120000 and above 40%

b exemptions contributions to lic, gpf, ppf, ulip, nsc, cd etc, are exempt from paying income tax subject to a maximum of rs. 120000 is admissible.

15. A linear array of size 50 stores following information's: name of the country, country's capital and per capita income of the country. write a complete program in c to do the following:
 a) to read a country's name and display capital and per capita income.
 b) to read name of the capital city and displays country's name and per capita income. display an error message incase of an incorrect input.

16. WAP using structure to store price list of 50 items and to print the largest price as well as the sum of all prices.
 17. WAP in c using structure to simulate result preparation system for 20 students. the data available for each student includes rollno, name and marks in 3 subjects. the percentage marks and grade are to be calculated from the above information, the percentage marks are the average marks are the average marks and the grade is calculated as follows:

% marks	grade
< 50	'f'
$\geq 50 < 60$	'd'
$\geq 60 < 75$	'c'
$\geq 75 < 90$	'b'
$\geq 90 < 100$	'a'

18. WAP to make a structure named "student" having following as structure member: 1) name 2) roll-no 3) marks of three subjects viz. English, hindi, maths. do the following operations using the structure:
 a) accept name, roll no and marks in three subjects.

- b) calculate total and percentage.
 c) show the information on the screen in given below format XYZ school half yearly examination

Name: roll no:

Marks in Hindi:

Marks in English:

Marks in Maths:

Total marks: per:

WAP to make a structure "contestant" for a beauty contest in which check the following condition & accept details for 5 contestants only if they satisfy following criteria:

- a) if age is between 18 to 20
 b) Weight is between 45 to 60
 c) Qualification is graduate

Structure members are: 1) Name 2) Age 3) Weight 4) Qualifications-->

1. Below graduate 2. Graduate 3. Postgraduate. Now display the details of all 5 contestant in tabular manner.

20. WAP to make structure "stock". Accept details of 10 stock items. The structure members are : 1) item_name 2) item_code 3) rate 4) qty_in_stock 5) amount. now ask of the user item code which he want to see, search it display it if it exit otherwise give appropriate message.

21. WAP to create a structure to specify data on students given below: roll number, name, department, course, year of joining assume that there are not more than 450 students in the college. do the following operations using the structure:

- a) print names of all students who joined in a particular year.
 b) print the data on a student whose roll number is given.

22. Create a structure to specify data of customers in a bank. the data to be stored is: account number, name, balance in account. assume maximum of 200 customers in the bank. do the following operations using the structure:

- a) to print the account number and name of each customer with balance RS. 100. if a customer requests for withdrawal or deposit, it is given in the form: acct. no, amount,(1 for deposit,0 for withdrawal)
 b) to give a message, "the balance is insufficient for the specified withdrawal".

23. An automobile company has serial numbers for engine parts starting from aa0 to ff9. The other characteristics of parts be specified in a structure are: year of manufacture, material and quantity manufactured. now, do the following:
 a) specify a structure to store information corresponding to a part.

b) WAP to retrieve information on parts with serial numbers between bb1 and cc6.

24. A record contains name of cricketer, his age, number of test matches that he has played and the average runs that he has scored in each test match. create an array of structures to hold records of 20 such cricketers and then write a program to read these records and arrange them in ascending order by average runs.

25. Create a structure to represent a book in a library. It include the following members: book number, book name, author, publisher, price, no. of copies, no. Of copies issued. now do the following operations using the structure:
 a) to assign initial values.
 b) to issue a book after checking for its availability.
 c) to return a book.
 d) to display book information.

26. Create a structure to represent bank account of 10 customers with the following data members: name of the depositor, account number, type of account (s for saving and c for current account),balance amount. now, do the following operations using the structure:

- a) To initialize data members
 b) To deposit money.
 c) For withdrawal of money after checking the minimum balance(minimum balance is rs. 1000).
 d) To display the data members.
27. Create a structure to represent batsman in a cricket team. It includes the following members: first name, last name, runs made, number of fours, number of sixes. Now do the following operations using the structure:
 a) to assign the initial values
 b) to update runs made(it should simultaneously update fours and sixes, if required)
 c) to display the batsman's information.
 make appropriate assumptions about access labels.
28. Create a structure to represent bowlers in a cricket team. include the following members: first name, last name, overs bowled, number of maiden overs, runs given, wickets taken. now do the following operations using the structure:
 a) to assign the initial values
 b) to update the information
 c) to display the bowler's information.
 make appropriate assumptions about access labels.
29. WAP to manage a room's statistics. the room structure includes the following members: length, width, height. now do the following operations using the structure:
 a) to assign initial values.
 b) to calculate area.
 c) to display information (length, width, height & area).
 modify the above program so that length, width and height become the variable of structure distance that includes meters, centimeters.
30. A company pays normal wage for work during weeks days from monday to friday and 1.5 times wage for work on saturday and sunday. Given data in the following form:
 employee number, wage/hour, hours worked on monday, hours on tuesday, ..., hours on sunday.
31. WAP to write out the employee number and weekly wages. use enumerated data type in your program.
32. Define a structure for a student having name, roll number and marks obtained in six subjects. assume that "all students" is an array of students. WAP to print the name and roll numbers of the students who have secured highest marks in each subject.
33. Define a structure "mca2_oops" which has the members entry_no, marks, marks_minor, marks_major, total. WAP to initialize the variables of objects, finding the total marks which is sum of marks_major and marks_minor. This program will handle 30 students and displaying their marks.
34. Create a structure of big cities bigcity of India, the data member of the structure are name of the city, std code(say for calcutta std code is 033) etc. WAP which interactively ask the name and address, local phone number of residents and print in the following format:
 1. Name: S.P.Rama Rao 2. Address: 3/2 APC road
 3. Pincode no.: 700052 4. Phone no: (033)-4347270
35. Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. the savings account provides compound interest and withdrawal facilities but not cheque book facility. the current provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. create a structure account that stores customer name, account number and opening balance. from this derive the structures current and savings to make them more specific to their requirements. now do the following tasks:
 a) deposit an amount for a customer and update the balance.
 b) display the account details.
 c) compute and deposit interest.
 d) withdraw amount for a customer after checking the balance and update the balance.
 e) check for the minimum balance(for current account holders), impose penalty, if necessary, and update the balance.
36. WAP defining an union which can hold an "integer" or "float" string. define a variable "union type" to keep track of the type of data stored in the union. write a function to print the value stored in the union.
37. WAP to define a union of type "ans" containing two members-an integer quantity and a floating quantity. Compute the average and standard deviation of the numbers and print them.
38. WAP that can maintain the name, roll number and marks of a class of students. the size of the class is variable. include functions to compute the average marks of the class.
39. A bookshop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the required copies is displayed, otherwise the message "Sorry! These many copies are not in stock" is displayed. Design a system using a structure called stock. This program includes the following operations:
 a) The price gets updated as and when required.
 b) The stock value of each book should be automatically updated as soon as transaction is completed.
 c) The total number of books (titles) sold get displayed (along with total sales in RS.) As and when required

40. WAP to print the score board of a cricket match in real time. The display should contain the batsman's name, runs scored, indication if out, mode by which out, bowler's score (overs played, maiden overs, runs given, wickets taken).as and when a ball is thrown, the score should be updated.(hint: use separate arrays to store batsman's and bowlers, information).
41. WAP to prepare the invoice from the following data: customer name, customer name, customer address, date of sale, item no, item description, quantity sold, unit price of item, discount percentage and sales tax percentage.
note: identify different structures possible here.
42. A college maintains a list of its students graduating every year. at the end of the year, the college produces a report that lists the following:
year:
number of working graduates :
number of non-working graduates :
details of the top-most scorer
name :
age :
subject :
average marks :
x% of the graduates this year are non-working and n % are first divisioners.
WAP for it that uses the following structure path:
person —> student —> graduate student
(name, age) (roll no, average marks) (subject, employed)
the data members of these structures have been shown in the parenthesis.
43. WAP to handle 10 account holders. the program should use the structure as defined in q.33.make necessary changes in the class definition - if required.
44. Write a structure to represent a vector (1-d numeric array).now do the following operations using this structure:
a) for vector creation
b) for modification of a given element.
c) for displaying the largest value in the vector.
d) for displaying the entire vector.
e) for adding two vectors and displays the resultant vector.
45. Create two structures mc and fi which store the value of distances. mc stores distances in meters and centimeters whereas fi stores in feet and inches. WAP that reads value for variables of both the structures and can add one variable of mc with an variable of fi.
46. Imagine a ticket selling both at a fair. people passing by are required to purchase a ticket. A ticket is priced as RS. 2.50/- . The booth keeps track of the number of people that have visited the booth, and of the total amount of money collected. Model this ticket selling booth with a structure called ticbooth including following members: number of people visited, total amount of money collected. Now do the following operations:
a) to assign initial values (assign 0 to both data members).
b) to increment only people total in case ticket is not sold out
c) to increment people total as well as amount total if a ticket is sold out.
d) to display the totals.
e) to display the number of tickets sold out(a tricky one).
WAP to include this structure.
48. WAP to process the sales activity for 20 salesman. Each salesman deals in separate product and is assigned an annual target. At the end of the month, his monthly sale is added into the sales till date. At the end of the year, his commission is calculated as follows: if sales made is more than target then the commission is 25% of the extra sales made + 10% of the target if sales made is equal to the target then the commission is 10% of the target. Otherwise commission is zero.
49. WAP that reads several different names and rearranges the names into alphabetical order, and then writes out the alphabetized list. make use of structure variables within the program.
50. Create a structure called volume that uses three variables (length, width, height) of type distance (feet and inches) to model the volume of a room. read the three dimensions of the room and calculate the volume it represents, and print out the result. The volume should be in $(\text{feet})^3$ form i.e., you will have to convert each dimension in to feet and fractions of foot. For instance, the length 12 feet 6 inches will be 12.5 feet.
51. WAP to store 20 records containing country, capital and name of its president. the president name is a record containing last name, first name, prefix(mr, miss, mrs.).the program should display the entire record whenever the country name or capital is given.
52. Suppose a store has a number of items in their inventory and that each item is supplied by almost two suppliers. WAP to store details of 20 items in an array and then print it.
53. An array stores details of 25 students (rollno, name, marks in three subjects).WAP to create such an array and print out a list of students who have failed in more than one subjects. assume 40% as pass marks.
54. WAP a c program to simulate an arithmetic calculator for integers. the program should be able to produce the last result calculated and the number of arithmetic operations performed so far. any wrong operations is to be reported.

12.1 Write a program to count the number of characters, spaces, tab, new lines in a file

```
1. #include<stdio.h>
2. #include<conio.h>
3. main( )
4. {
5.     FILE *fp;
6.     char ch;
7.     clrscr();
8.     int lines=0,tab=0,space=0, characters=0;
9.     fp=fopen("text.c","r");
10.    while(1) /* infinite loop*/
11.    {
12.        ch = fgetc(fp);
13.        if(ch==EOF)
14.            break;
15.        characters++;
16.        if(ch==32)
17.            space++;
18.        if(ch=='\n')
19.            lines++;
20.        if(ch=='\t')
21.            tab++;
22.    }
23.    fclose(fp);
24.    printf("number of lines = %d\ntabs = %d\ncharacters = %d\nspaces =
25.          %d",lines,characters,spaces);
26.    getch( );
27. }
```

Matrix ✕

12.2 Write a program to receive some strings from keyboard and print it to a file

```
1. #include<stdio.h>
2. #include<conio.h>
3. main( )
4. {
5.     FILE *fp;
6.     char s[50];
7.     fp=fopen("text.c","w"); /*opening file*/
8.     if(fp==NULL) /*to check opening errors*/
9.     {
```

```

10.         puts("file opening error");
11.         exit();
12.     }
13.     printf("Enter some text");
14.     while(strlen(gets(s))>0)
15.     {
16.         fputs(s,fp);
17.         fputs("\n",fp);
18.     }
19.     fclose(fp);
20. }
```

12.3. Write a program to read all the strings from file and print it on screen

```

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5.     FILE *fp;
6.     char s[50];
7.     fp=fopen("text.c","r"); /*opening file*/
8.     if(fp==NULL)           /*to check opening errors*/
9.     {
10.         puts("file opening error");
11.         exit();
12.     }
13.     while(fgets(s,49,fp)!=NULL)
14.         printf("%s",s);
15.     fclose(fp);
16.     getch();
17. }
```

Multiple Choice Questions:-

- | | |
|---|--|
| 1. What values are returned from <code>fclose</code> if the file is closed properly: | 5. Given: <code>FILE file_p;</code>
<code>file_p=fopen("text.dat","a+");</code>
The error in the above is |
| a. zero
b. one
c. -1
d. none | a. <code>FILE FILE_p</code>
b. <code>FILE *filem_p</code>
c. <code>FILE *file_p</code>
d. none |
| 2. The function that is used to see an error, if occurred, is | 6. To close a file, we use |
| a. <code>ferror</code>
b. <code>fiota</code>
c. <code>fseek</code>
d. none | a. <code>fexit(file_p)</code>
b. <code>fclose</code>
c. <code>break</code>
d. <code>fclose(file_p)</code> |
| 3. Which mode opens the file for reading and writing mode | 7. Which condition is used to test the end of file condition |
| a. <code>r++</code>
b. <code>i++</code>
c. <code>w++</code>
d. <code>r+</code> | a. <code>eof</code>
b. <code>ferror</code>
c. <code>feof</code>
d. none |
| 4. In the following code:
<code>FILE *fp;</code> | 8. Which function takes a file pointer and resets the position to start a file |
| | a. <code>fseek</code> |

- fp = fopen("trial", "r");
 fp points to
- The first character in the file.
 - A structure which contains a *char* pointer which points to the first character in the file.
 - The name of the file.
 - None of the above.

- rewind
- fseek
- all the above

Answers:- 1 (a) 2 (a) 3 (d) 4 (b) 5 (c) 6 (b) 7 (c) 8 (b)

Point out the errors, if any, in the following program segments:

1.

```
#include <stdio.h>
void openfile(char *, FILE **);
main()
{
    FILE *fp;
    openfile("Myfile.txt",&fp);
    if(fp == NULL)
        printf("Unable to open file... ");
}
```
2.

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    FILE *fp;
    char c;
    fp = fopen( "try.c" , "r" );
    if(fp== null)
    {
        puts("Cannot open file");
        exit( );
    }
    while((c = getc(fp))!=EOF);
        putch(c);
    fclose( fp );
}
```
3.

```
main()
{
    char fname[ ] = "c:\\students.dat";
    FILE *fp;
    fp = fopen( fname , "tr" );
    if (fp == NULL)
        printf("\nUnable to open file...");
```
4.

```
main( )
{
    FILE *fp;
    char str[80];
    fp = fopen("Try.c","r");
    while(fgets(str,80,fp)!= EOF)
        fputs(str);
    fclose(fp);
```
5.

```
unsigned char;
FILE *fp;
fp=fopen("matrix", "r");
while((ch=getc(fp))!=EOF)
    printf("%c",ch);
fclose(fp);
```
6.

```
unsigned char;
FILE *fp;
fp=fopen("matrix", "r");
if(!fp)
{
    printf("file opening error");
    exit();
}
fclose(fp);
```
7.

```
#include <stdio.h>
main()
{
    unsigned char;
    FILE *fp;
    fp = fopen( "trial" , 'r' );
    while((ch = getc(fp))!=EOF)
        printf("%c",ch);
```

M@trix Computers

```

8.
main( )
{
    FILE *fp;
    char name[25];
    int age;
    fp = fopen("yours", "r");
    while(fscanf(fp, "%s%d", name, &age)
!=NULL)
        printf("%s %d\n", name, age);
    fclose(fp);
}

9.
main( )
{
    FILE *fp;
    char names[20];
    int i;
    fp = fopen("student.dat", "wb");
    for(i = 0; i<10; i++)
    {
        puts("Enter name");
        gets(name);
        fwrite(name, size of(name), 1, fp);
    }
}

10.
FILE *fp;
fp=fopen("matrix", "r");
fseek(fp, 20, SEEK_SET);
fclose(fp);

11.
FILE *fp;
char s[80];
fp=fopen("matrix", "r");
while(!feof(fp))
{
    fgets(s, 80, fp);
    puts(s);
}
fclose(fp);

12.
FILE *fp;
fp=fopen("c:\tcl\matrix", "w");
if(!fp)
    exit();
fclose(fp);

```

Output:

1. No Error.
 2. Error. "null" must be in upper case because it is predefined in stdio.h as `#define NULL 0`.
 3. No Error. but remember "stdio.h" must be included.
 4. Error. because fputs() needs two argument and written as fputs(str,fp) and because we use string in loop so there is NULL in the place of EOF.
 5. EOF has been defined as `# define EOF -1` in the file "stdio.h" and an *unsigned char* ranges from 0 to 255 hence when EOF is read from the file it cannot be accommodated in *ch*. Solution is to declare *ch* as an *int*.
 6. No error
 7. Error. Type mismatch in parameter '*mode*' in call to fopen. Because the mode must be enclosed in double quotes.

fp); Max

8. Error. EOF is used instead of NULL because with fscanf will read the record from the file and when it will go to the last record it will meet with EOF.
 9. Error. Because sizeof should be one word and fclose should be used to close the file.
 10. Instead of 20 use 20L since fseek() needs a long offset value.
 11. The last line of file "matrix" will be read twice. To avoid this, use:

```
while(fgets(s,80,fp)!=NULL)
    puts(s);
```
 12. The path of the filename should be written as "c:\\tcl\\matrix".

Multiple Choice:

1.

What is a preprocessor directive?

1. a message from compiler to the programmer
2. a message from compiler to the linker
3. a message from programmer to the preprocessor
4. a message from programmer to the microprocessor

2.

Which of the following are correctly formed
#define statements

1. #define INCH PER FEET 12
2. #define SQR(X) (X*X);
3. #define SQR(X) (X*X)

3.

Which of the following is not a preprocessor directive?

1. #if

Output:

(1) 3 (2) 3 (3) 2 (4) 2

2. #elseif
3. #undef
4. #pragma

4.

A header file is:

1. a file that contains standard library functions
2. a file that contains definitions and macros
3. a file that contains user-defined functions
4. a file that is present in current working directory

5.

All macro substitutions in a program are done

1. Before compilation of the program
2. After compilation
3. During execution
4. None of the above

(5) 1

What will be the output of the following program:

1.

```
#define sqr(x) (x*x)
main()
{
    int a,b=3;
    a=sqr(b+2);
    printf("%d",a);
}
```

2. How would you define the sqr macro in above question such that it gives the result of a as 25.

3.

```
# define cube(x) (x*x*x)
main()
{
    int a,b=3;
    a=cube(b++);
    printf("%d %d",a,b);
}
```

4.

```
#define MESS junk
main()
{
    printf("MESS");
}
```

5.

```
#define PRINT(int) printf("%d",int)
main()
{
    int x=2,y=3, z=4;
    PRINT(x);
    PRINT(y);
    PRINT(z);
}
```

6. How would you define the above macro such that it print x=2 y=3 z=4

7.

```
#define max(a,b) (a>b?a:b)
main()
{
    int x;
    x=max(3+2,2+7);
```

Output:

1. 11 because, on preprocessing the expression becomes $a=(3+2*2+3)$.
2. **#define sqr(x) ((x)*(x))**
3. 27 6.
4. MESS
5. 2 3 4

6.

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

8. What is the difference between the following two **#include** directives:

```
#include<conio.h>
#include"conio.h"
```

9. Indicate what would the swap macro be expanded to on preprocessing. Would the code compile?

```
#define swap (a, b, c) (c t; t=a; a=b;
b=t)
main()
{
    int x=10,y=20;
    swap (x, y, int);
    printf("%d %d",x,y);
}
```

10. How would you modify the swap macro in the above question such that it is able to interchange two integers?

In which line of the following program an error would be reported?

1. **#define area(r) (3.14*r *r);**
2. main()
3. {
4. float r=1.0, c;
5. c=area(r);
6. printf("%f",c);
7. if(area(r)==6.28)
8. printf("Matrix");
9. }

11. What is the type of the variable b in the following declaration?

```
#define floatptr float*
floatptr a,b;
```

12. Is it necessary that the header files should have .h

13. What do header files usually contain?
 14. On inclusion, where are the header files searched for?

15.

```
#define PRINT(int) printf(#int " = %d",
int)
main()
{
    int x=2,y=3, z=4;
    PRINT (x);
    PRINT (y);
```

PRINT (z);

}

The rule is if the parameter name is preceded by a # in the macro expansion, the combination (of # and parameter) will be expanded into a quoted string with the parameter replaced by the actual argument. This can be combined with the string concatenation to print the output desired in our program. On expansion the macro becomes printf("x" "%d",x); The two strings get concatenated, so the effect is printf("x=%d",x);

7.

#include "conio.h": This command would look for the file conio.h in the current directory as well as the specified list of directories as mentioned in the search path that might have been set up.

8.

#include<conio.h>: This command would look for the file conio.h in the specified list of directories only.

9.

(int t;t=a,a=b,b=t);
This code will not work since declaration of t cannot occur within parenthesis.

10.

#define swap(a,b,c) c t;t=a, a=b, b=t;

State whether true or false:

- Matrix*
- a. If the file to be included doesn't exist, the preprocessor flashes an error message.
 - b. The preprocessor can trap simple errors like missing declarations, nested comments or mismatch of braces.
 - c. Would it result in an error if the header file is included twice.
 - d. Would the following program print the message infinite number of times?

```
#define INFINITELOOP while (1)
main()
{
    INFINITELOOP
    printf("\nMatrix");
}
```
 - e. Would the following program compile successfully?

```
main()
{
    printf("Matrix" "computers");
}
```

Answer:

- | | | | | |
|---------|----------|----------|---------|---------|
| a. True | b. False | c. False | d. True | e. True |
|---------|----------|----------|---------|---------|

Register

CPU Registers

Garbage

Global
LocalGlobal
Local

What will be the output of the following:-

1.

int f(int);

int g(int);

main()

{

int x, y, s = 2;

s *= 3;

y = f(s);

x = g(s);

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

}

int t = 8;

int f(int a)

{

M@trix Computers

```

        a += -5;
        t -= 4;
        return(a+t);
    }
    Int g(int a)
    {
        a = 1;
        t += a;
        return(a+t);
    }
2.
main()
{
    static int c = 5;
    printf("\n c = %d",c--);
    if (c !=0)
        main();
}
3.
main()
{
    int i,j;
    for( i = 1; i<5; i++)
    {
        j = g(i);
        printf("\n%d",j);
    }
}
Int g( int x)
{
    static int v = 1;
    int b = 3;
    v+=x;
    return(v+x+b);
}
4.
main()
{
    static int a[20];
    int i=0;
    a[i]=i++;
    printf("%d %d %d", a[0], a[1], i);
}
5.
float x =4.5;
main()
{
    float y, float f(float);
    x*=2.0;
    y = f(x);
    printf("\n%f %f",x,y);
}
float f(float a)
{

```

Matrix

```

        a+=1.3;
        x-= 4.5;
        return(a+x);
    }
    6.
    int x = 10;
    main()
    {
        int x = 20;
        {
            int x = 30;
            printf("%d ",x);
        }
        printf("\n %d",x);
    }
    7.
    main()
    {
        extern int i;
        i=20;
        printf("%d",sizeof(i));
    }
    8.
    main()
    {
        extern int a;
        printf("%d",a);
    }
    int a=20;
    9.
    int recsum(int);
    main()
    {
        int z, y ;
        z = recsum(1) ;
        y = recsum(1) ;
        printf("z = %d\n y = %d", z, y) ;
    }
    int recsum(int i)
    {
        static int sum = 0 ;
        if (i == 3)
            return(sum) ;
        else
        {
            sum = sum + 10 ;
            i++ ;
            recsum(i) ;
        }
    }
    10.
    main()
    {
        static int c = 5 ;

```

```

printf("c = %d\n", c--) ;
if(c)
    main() ;
}
11.
main()
{
    int c = 5 ;
    printf("c = %d\n", c--) ;
    if(c)
        main() ;
}
12.
int i ;
int function(int) ;
main()
{

```

Output:

1. 6 5 6
2. 5 4 3 2 1
3. 6 9 13 18
4. 0 0 1
5. 4.500000 14.800000
6. 30 20
7. Error extern int i is a declaration and not definition
8. 20
9. z = 20
10. y = 40
- c = 5

```

int j ;
for(..)
{
    if(j = function (i))
        printf("j = %d\n", j) ;
    else
        break ;
}
int function(int x)
{
    static int v = 2 ;
    v-- ;
    return(v-x) ;
}

```

c = 4
c = 3
c = 2
c = 1
11. c = 5
c = 5
c = 5
... ...

12. j = 1
... ...

State the True or False:

1. An extern storage class variable is not available to the functions that precede its definition, unless the variable explicitly declared in these functions.
2. The value of an automatic storage class variable persists between various functions invocations.
3. If the CPU registers are not available, the register storage class variables are treated as static storage class variable.
4. The register class variables cannot hold float values.
5. If we try to use register storage class for a float variable the compiler will flash an error message.
6. If the variable x is defined outside all functions and a variable x is also defined as a local variable of some function, then the global variable get preference over the local variable.
7. The default value for automatic variable is zero.
8. The life of static variable is till the control remains within the block in which it is defined
9. If a global variable is to be defined, then the **extern** keyword is necessary in its declaration.
10. The address of register variable is not accessible.

Answers:

1. True	2. False	3. False	4. True
5. False	6. False	7. False	8. False
9. False	10. True		

Chapter 15 Algorithm

Algorithm No. 1: Sum of two numbers

1. Read A and B.
2. Set SUM:=A + B.
3. Write SUM.
4. Exit.

Algorithm No. 2: Maximum of two numbers

1. Read A and B.
2. If A > B, then:
 Set MAX:=A.
Else:
 Set MAX:=B.
[End of If structure]
3. Write MAX.
4. Exit

Algorithm No. 3: Maximum of three numbers

1. Read A , B,C.
2. If A > B, then:
 If A > C, then:
 Set MAX:=A.
 Else:
 Set MAX:=C.
 [End of If structure]
Else:
 If B > C, then:
 Set MAX:=B.
 Else:
 Set MAX:=C.
 [End of If Structure.]
[End of If Structure]
3. Write MAX.
4. Exit.

Algorithm No. 4: To input percentage from user and print the grade.(Using Else if)

1. Read P.
2. If $P \geq 90$, then:
 Set GRADE:='A'.
Else If $P \geq 70$, then:
 Set GRADE := 'B'.
Else if $P \geq 50$, then:
 Set GRADE:='C'.
Else:
 Set GRADE:='F'.
[End of If Else Structure].
3. Write GRADE.
4. Exit

Algorithm No. 5: To print the series 1 to N using for loop.

1. Read N.

2. Repeat Step 3 for $I = 1$ to N :
 Write I
3. [End of Step 2 loop]
4. Exit.

Algorithm No. 6: To print the series 1 to N using while loop.

1. Read N
2. Set $I := 1$
3. Repeat Step 4 and 5 While $I \leq N$:
 Write I
4. Set $I := I + 1$.
5. [End of Step 3 loop]
6. Exit.

Algorithm No. 7: To print the series 1 to N using do while loop.

1. Read N
2. Set $I := 1$
3. Write I
4. Set $I := I + 1$.
5. if $I \leq N$, then: goto step 3.
6. Exit.

Algorithm No. 8: Reverse the digits of given n digit numbers.

1. Read N .
2. Set $REV := 0$.
3. Repeat steps 5 and 6 While $N >= 0$.
4. Set $N := N / 10$
5. Set $REV := REV * 10 + \text{remainder}$.
6. Print the value of REV .
7. Exit.

Algorithm No. 9: To verify whether a given number is prime or not.

1. Read N .
2. Set $I := 2$ and $PRIME = 1$.
3. Repeat Steps 4 while $I \leq N/2$
4. If $(N \% I = 0)$
 Set $PRIME := 0$. and exit the loop structure.
Else:
 Set $I := I + 1$.
- [End of If Structure.]
- [End of While loop]
5. If $PRIME = 1$, then:
 Print the number is prime.
- Else
 Print the number is not prime.
- [End of If structure.]
6. Exit.

Algorithm No. 10: Linear search in an array

1. Read N
2. Repeat step 3 For $I := 0$ to $n-1$
 Read $A[I]$.
- [End of Step 2. loop]

M@tix Computers

4. Read ITEM.
5. Set FOUND:=FALSE.
6. Repeat step 7 For I:=0 to n-1 [Searching]
 7. If A[I]= ITEM, then:
 - a. Set FOUND.=TRUE.
 - b. GoTo Step 8.
 - [End of If Structure.]
- [End of Step 6. loop]
8. If FOUND=TRUE, then:
 - Write "Item is found at position:", I+1.
- Else:
 - Write "Item not Found".
- [End of If Structure.]
9. Exit.

Algorithm No. 11: Calculate factorial using function

1. Read N.
2. ANS=FACT (N)
3. Write ANS.
4. Exit

[This procedure is used to calculate the factorial of N]

1. Set ANS:=1
2. Repeat step 3 for I:=N to 1
3. Set ANS:=ANS*I
4. [End of Step 3 loop]
5. Return ANS.
6. Exit

Matrix +