```
int bin=0, m=1
while(n!=0)
{ r=n%2
  bin=bin+(r*m)
  m=m*10
  n=n/2 }
```

1) Reverse of binary or palindrom (reverse = orignal)
or any no.

5)

```
scanf("%d",&n);
* orignal=n;
  while (n!=0)
  { remainder= n%10;
    rev= rev*10 + remainder;
    n=n/10;
  }
  printf("%d is reverse of %d", rev, orignal);
  if (Orignal==rev)
  { printf("palindrome");
  }
  else
  { printf("not");
  }
```

(num%10 == num/100)

Case1: 1000 ↔ 0001 x Only 1 ✓

3) amstrong

$371 = 3^3+7^3+1^3$

```
temp=n
{
r=n%10;
sum = sum + (r*r*r)
n=n/10;
}
if (temp==sum)
```

amstrong.

4)

```
while(n!=0)
{ r=num%10.
sum= sum+r.
} num=num/10
printf("sum: %d", num);
```

sum of digits.

1000 ↔ 0001

2) even or odd with %

```
if((num & 0001)==0)
{
printf("even");
}
else
{ printf("odd");
}
return 0;
}
```

swap.

temp=num1
num1=num2
num2=temp.

a → a+b
b → a-b
a → a-b.

D00001
+
convert
karegha.

6)

```
while (n>0)
{
if (n<10)
{ printf("%d", n);
}
else if(n%10==0)
{ printf("0");
}
else
{ printf("1");
}
n=n/10;
}
return;
```

num → binary
or normal 10 or 1010

7) Sum until 0 entered.

```
#include <stdio.h>
int main()
{
int num=1;
int sum=0;
while (num!=0)
{ printf ("enter number\n");
scanf ("%d", &num);
sum=sum+num;
}
printf ("sum=%d", sum);
return 0;
}
```

8) binary.

```
while (n>0)
{
if (n%2==0)
{ printf ("0"); }
else
{ printf ("1"); }
n=n/2;
}
return 0;
}
```

Reverse.

```
int n=10
while (n>0)
{ printf ("%d", n%2);
n=n/2;
}
return;
}
```

9) equal without ÷

```
if !(num1^num2)
} equal {
else
} not equal {
```

10) 1 is odd
2 is even
3 is odd .....

```
for (num=1; num<=10, num++
{ if (n%2==0)
{ printf ("%d is even", num);
else
{ printf ("%d is odd", num);
} }
return 0;
}
```

11) 9 is not div by 2
.. 9 is not div by 8

```
printf (Enter any no.\n");
scanf ("%d", &num);
for (i=2; i<num; i++)
if (num%i==0)
{ printf ("no. is div by %d", num,i);
else
{ printf ("no is div by %d", num,i);
}
return 0;
}
```

12)

```
for (j=1 ; j<=10; j++)
{ printf ("%d x %d = %d\n",
n, j, n*j);
}
return 0;
}
```

13)

num = 0001 → last digit right shift
(num) & (1<<2) → 2nd digit right shift
(1101) & (1<<2) → 0.
↳ other than 0 → (1)

not prime

```
for (i=1; i<100; i++)
  for ( j=2 )
  { if (i%j==0)
    { "\n", i)
    } break; }
  if (j==i)
  { print("") }
  }
  return
}
```


Code Class notes →

Q14 to check leap year or not

```
#include <stdio.h>
int main ()
{
int
printf("enter any year");
Scanf ("%d", &num);
if ( num % 4 != 0)
{ printf (" not leap year"); }
else
{ if (year % 100 && year % 400 ==0)
                 ==0
printf ("leap");
else if (year % 400 =0 && year % 400 !=0)
printf ("not leap");
else
printf ("leap");
}
return 0;
}
```

16)

```
int num=0;
while (num<=10);
{
num ++;
if (num % 2==0)
{ continue;
}
if (num % 5 ==0)
{ break;
}
printf ("%d", & num);
}
printf ("out of loop")
```

output:
1 3
out of loop.

continue } true } loop mein
loop } false.

break — loop out true.

false [↓ chalti] loop rehta.

(PRINT PRIME)

```
#include<stdio.h>
int main ()
{ int i, j;
for (i=2; i<100; i++)
{
  for (j=2, j<i, j++
  { if (i%j==0)
    { break; }
  }
  if (j==i)
  { printf ("%d\n", j); } }
return 0
}
```

15) prime ?

```
#include <stdio.h>
int main () {
int num;
printf ("enter no");
scanf ("%d", & num);
for( i=2; i<num; i++)
{
  if (num % i==0)
  {printf ("not prime");
   break; }
}
if (i== num)
{printf ("prime"); }
return 0;
```

```
int flag=0
for(i=2; i<num && flag==0;
i++)
if (num % i ==0)
{ printf ("prime");
flag =1
}
```

int num, d, k ...
k=0

loop { { not prime k=1; i=num; } }

if (num==2 || i==num)
prime (2)

loop { (not prime) }

if (k==0)
print [prime]

[%] use nahi kar sakti

num % i = num-(i*d);
where :- d= num/i
logic use

[break] use nahi
i = num after num % i condition
loop bahar.

* without

x+y while(y!=0)
{ mulp = x+multip.
y--
}

17) count the no.

count=0
while (n!=0)
{ count=count+1
n=n/10;
}

18)

factorial
fact=1;
while(num>0)
{ fact=fact*num
num--;
}
printf("fact is %d", fact);

19)

count odd/even/zeros in array

scanf("%d", &t);
for (i=1; i<10; i++);
{ if(t%2==0 && t!=0)
t++;
else if (t<0)
++;
else
++;
}

{ print scanf;
if ( )
else even
t!=0
{ if → odd
else → 0.
}
}
top

printf("print f even : %d \n", K);

20) divide by 8 → right shift
or multiply → left shift

21)

{
printf("enter any hexadecimal no");
scanf("%x", &num);
n1 = num/16;
n2 = num % 16;
printf("simple hexadecimal = %x%x", n2, n1);
return 0;
}
}

AB → BA

22)

Largest = (a>b && a>c && a>d) ? a :
(b>c && b>d) ? b : (c>d) ? c :

char var;
printf
scanf("%c", &var);
if (var >= 'A' && var <= 'Z')
{ printf("Capital"); }
else if (var >= 'a' && var <= 'z')
{ printf("Small"); }
else
printf("invalid");
return 0;
}

23)

24)

if (var == 'a' || var == 'e' || var == 'i' || var == 'o' || var == 'u')
printf("vowel");
else if (var == 'A'
printf("vowel");
else
{ printf("consonant"); }

25)

Switch (num%2)
{ case 0: printf("even"); break;
case 1: printf("odd"); break;
}
return 0;

GoodLuck | Page No.
Date

Octal / Hexa decimal / decimal / binary converter

/8 %8 /16 %16 /10 %10 /2 %2

#1 swap:

```
scanf("%d %d", &num)
n1 = num/16
n2 = num % 16;
printf("swapped = %x %x", n2, n1);
return 0;
```

#2

AB ⟷ BA

hexadecimal A: 10

DECIMAL
↓
conversions.

```
int i, n, a[10];
printf(
scanf("%d", &n);
for(i=0; n!=0; i++)
{ a[i] = n%2;        8   16
} n = n/2            8   16.
n = i;
printf("value is");
for(i=0; i<n; i++)
{ printf("%d", a[n-i-1]);
}
return 0;
}
```

Octal to decimal →

%o

%d

hexadecimal → decimal

```
int n;
scanf("%x", &n);
printf("%d", n);
return 0;
```

binary → decimal

```
bin=0.
for( )
{
a[i]=n%10
n=n/10.
}
n=i;
for( )
{ bin = a[i] x pow(2*i)
}
printf("%d", bin
}
```

Pyramids →

```
A B C D E D C B A
  B C D E B C B
    C D E B C
      D E D
        E
```

```
int s, e;
s = 65;
e = 69;
i = 0; i<5, i++
j = s, j<=e, j++
printf("%c", (char) j);
for (j = e-1; j >= s; j--)
    printf("%c", (char) j);
for (j = 1; j <= 9; j++)
    printf(" ");
```

```
A
B C
D E F
G H I J
K L M N O
```

```
n = 15
i = 1
k = 5
for (i = 65; i <= n+65; i++) printf("\n");
printf("%c", (char) i);
```

```
A
A B
A B C
```

```
i = 1, i <= 5, i++
j = 1  j <= i  j++
"%c", j+64
\n
```

```
A A
B B B
C C C
```

```
A
A B
A B C
A B C D
```

```
for (i = 1; i <= 4; i++)
{
for (j = 1; j <= (4 - i); j++)
    printf(" ");
}
for (j = 65; j <= 65+i; j++)
    printf("%c", (char) j);
```

A B C D
A B C D
A B C D
A B C D

```
for (row=1 ; row<=4 ; row++)
{
for (col=1, col<=4, col++)
{
    d=col+64,
    printf ("%c", (char) d);
}
```

A
A B
A B C

```
i=1 ; i<=3; i++
col=1; col<=row ; col++
d = col+64
printf ("%c", (char) d);
}
printf ("\n");
```

```
char ch='A';
for (i=1 ; i<=5 ; i++)
{ for (j=1 ; j<=i ; j++)
    ("%c", ch);
X ← ch++
}
```

A
b c
D E F

✓ ← ch='A'; → X

("\n");

}

ch++
out

char 'A'/X

A
B B
C C C
D D D D

char 'A'

↓

same

A
A B
A B C.

.

same.

GoodLuck | Page No.
Date

```
1
2 2
3 3 3
4 4 4 4
```

```
1
1 2
1 2 3
1 2 3 4
```

```
for (i=1 ; i<=4 ; i++)
{ for (j=1 ; j<=i ; j++)
{ printf ("%d", i); }
for (j=1 ; j<=8-2i ; j++)
{ printf (" "); }
for (j=1 ; j<=i ; j++)
{ printf ("%d", j); }
printf ("\n"); }
return 0;
```

```
1
2 1
3 2 1
4 3 2 1
```

```
for (i=1; i<=4 ; i++)
{
for (j=i ; j>=1 ; j--)
{ printf ("%d", j);
}
printf ("\n");
```

```
1 1
2 1 2 2
3 1 3 2 3 3
4 1 4 2 4 3 4 4
```

```
for (row=1 ; row<=4 ; row++)
{ for (col=1 ; col<=row ; col++)
{ printf ("%d %d ", row, col)
}
printf ("\n");
},
```

(b)

```
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```

```
for (row=1 ; row<=4 ; row++)
{ for (col = 1 ; col<=5 ; col++)
{ printf ("%d", col)
}
printf ("\n")
}
return 0;
}
```

(c) (b)

```
("%d", row)
```

(c)

```
printf ("%d", 2 * row)
```

(a)

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

(d)

```
1 1 1 1 1
2 2 2 2 2
```

```
    1
   1 2
  1 2 3
 1 2 3 4
```

```
for (i=1; i<=4; i++)
{
for (ws=1; ws<=4-row; ws++)
{ printf (" "); }
for (col=1; col<=row; col++)
{ printf ("%d", col);
}
printf ("\n");
return;
}
replace ("%d", row)
```

```
    1
   1 2 1
  1 2 3 2 1
 1 2 3 4 3 2 1
```

```
for (i=1; i<=4; i++)
{ int a=row-1;
for (ws=1; ws<=4-row; ws++)
{printf (" "); }.
for (j=1; j<=i; j++)
{ printf ("%d", j); }
for (k=1; k<=row-1; k++)
{ printf ("%d", a);
a--; }
printf ("\n"); }
```

```
1 2 3 4
1 2 3
1 2
1
```

```
for (i=1; i<=4; i++)
{ for (j=1, j<=4-i+1; j++)
{ printf ("%d", &j);
}
```

```
  1
 1 3
1 3 5
1 3 5 7
```

```
for (i=1; i<=num; i++)
{
for (col=1; col<=num-1; col=col+2)
printf ("%d", col);
printf ("\n");
}
```

```
(i=1; i<=4, i++
(j=1, j<=i; j++)
("%d", 2*j-1)
("\n")
```

Star →

GoodLuck | Page No.
Date

```
*
* *
* * *
* * * *
* * * * *
```

```
for(i=1; i<=5; i++)
{
k=2;
for(j=1; j<=9; j++)
{
for(j>=6-i && j<=4+i && k)
{ printf("*");
  k=0;
}
else
{ printf(" ");
  k=1
}
printf("\n");
```

```
i=1; i<=n; i++
{
j=n-1; j>=i; j--
(" ")
k=1, k<=i, k++
("*")
(\n)
```

→ space

```
*
* * *
* * * * *
* * *
*
```

```
i=1; i<=3; i++
{
j=1, j<=3-i, j++
printf(" ");
for(j=1; j<=2*i-1; j++)
printf("*");
}
printf("\n");
}
```

```
* * * * * * * *
* * *     * * *
* *       * *
*         *
```

```
for(i=6; i>=1; i--)
{
for(j=1; j<=i; j++)
printf("*");
for(j=1; j<=2*(6-i); j++)
(" ")
for(j=1; j<=i; j++)
("*")
\n
```

```
for(i=2; i>=1; i--)
{
for(j=1; j<=2-i+1; j++)
{ printf(" ");
}
for(j=1; j<=2*i-1; j++)
{ printf("*");
}
printf("\n");
}
```

```
*
* *
* * *
```

```
→ for(i=1; i<=3; i++)
{
for(j=1; j<=i; j++)
{ printf("*");
}
printf("\n");
}
```

```
* * *
* *
*
```

```
for(i=1; i<=3; i++)
{
for(j=1; j<=4-i; j++)
{ printf("*")
}
printf("\n");
}
```

```
for(i=1; i<=4; i++)
{
for(col=4; col<=i+0; col)
{ printf("*");
}
printf("\n");
}
return;
```

GoodLuck
Date

```
    *
    *
* * * * *
    *
    *
```

```
for (i=1 ; i<=5 ; i++)
{
for (i=1; i<=5 ; i++)
{ if (row==3 || col==3)
{ printf("*"); }
```

```
  *
* * *
* * * * *
```

```
for(i=1 ; i<=3 ; i++ )
for (ws=1; ws<=4-i ; ws++)
{ printf (" "); }
for (j=1 ; j<=2i-1 ; j++)
{ printf ("*"); }
printf ("\n");
}
return
```

Numbers →

Patterns

Numbers

1) 1 2 3 4
2 3 4 5
3 4 5 6
4 5 6 7

```
for (i=1, i<=4; i++)
{
  for (j=i; j<=i+3; j++)
  { printf("%d", j); }
  printf("\n");
}
```

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

```
for (i=1, i<=4; i++)
{
  for (j=1; j<5-i; j++)
  {
    printf("%d" j);
  }
  printf("\n");
}
return 0;
}
```

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

```
for (i=1; i<=4; i++)
{
if (i%2!=0)
{ for(j=1; j<=n; j++)
  { printf("%d", j); }
}
else
{ for (j=n; j>=1; j--)
  { printf("%d", j);
  }
}
printf("\n")
}.
```

[N=3]
1 2 3
3 2 1
1 2 3

4) n → replace i

[N=3]

1	1
2 1	2 1
1 2 3	1 2 3
	4 3 2 1

1 1 1
2 2 2
3 3 3

```
for(i=1; i<=3; i++)
{
for(j=1; j<=i; j++)
printf("%d ", i);
printf("\n");
}
```

i=1; i<=3; i++
j=1; j<=3; j++

f)
```
1
1 2
1 2 3
```

```
for(i=1; i<=5; i++)
{ for(j=0; j<6; j++)
{ printf("%d", (int) pow(i+1, j));
}
printf("\n");
}
return 0;
}
```

5) 1 2 4 8 16 32

1 5 25

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

```
int n, j, i;
n=15;
j=1;
for(i=1; i<=n; i++)
{ printf("%d", i);
if(i==1||i==3||i==6||i==10||j
{ printf("\n");
}
}
return 0;
}
```

7)
```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

i=1; i<=5; i++
j=1, j<=5-i; j++
(" ")
j=1, j<=i, j++
("%d", i)
WS
("\n")

i=1; i<=n; i++
j=n-1, j>=i, j--
(" ")
k=1; k<=i, k++
"%d", i
\n
WS

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

i=1; i<=5; i++
j=1; j<=5-i; j++
" "

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

without WS.

10. 0 1 2 9 12 29

j=1, j<=i, j++
value=1
"%d", value
value++
WS below

n1=1
n2=0
pell=0
for(i=1; i<=n; i++)
{ pell = n1 + (2*n2); printf("%d", pell);
n1=n2; n2=pell;

arrays.

SWAP

REVERSE

another array use

not use.

```
int arr[10];
for(i=0; i<10; i++)
{ printf(" ", i+1);
  scanf("%d", &arr[i]);
}
int rev[10];
for(i=0
{
}
for(i=0; i<10; i++)
{ int temp = arr[i];
  arr[i] = rev[i];
  rev[i] = temp;
}
for( )
{ printf("arr[%d] = %d\n", i, arr[i]);
}
for( )
{ }
return 0;
}
```

```
for( )
{ printf("arr[%d] = %d", i, arr[i]);
}
for(i=0; i<=4; i++)
{ int temp = arr[i];
  arr[i] = arr[9-i];
  arr[9-i] = temp;
}
for(i=0; i<10; i++)
{ arr[%d] = %d
  i, arr[i]
}
```

```
for( )
{ printf
  scanf;
}
for( )
{ printf
  ["arr[%d] = %d", i, arr[i]
}
printf("\n");
int rev[10]
for(i=0, i<=9, i++)
  rev[i] = rev[9-i];
for(i=0, i<10,
  (arr i, rev[i],
```

fuid

int arr[10] ; largest element

for ( )

{ it1

} arr[i]

int max=0;

for ( )

{

if (arr[i] > max)

{ max=arr[i];

}

}

printf (max = "%d" max

int arr[10]

for(i=0; i<10, i++)

{ printf (i+1

scanf("%d" & arr[i])

}

float sum=0;

float average;

for (i=0

{ sum=arr[i]+sum;

}

sum.

avrg=sum/10;

printf (" the sum = %, f and avrg = %, f sum avrg

);

}

Second largest element.

int max = 0

int f ;

for (i=)

if (arr[i] >)

{

max = ;

f=i;

}

arr[f]=0

int smax =0

for (i=

{

if (arr[i] > smax

{

smax = arr[i]

}

}

printf (" %d", smax

Sum of array & average.

```
#include
int main
{ float a;
  int marks[5], s=0, i, j;
  for(i=0; i<5; i++)
  {
    printf( )
    scanf( ), &marks[i]
  }
  for(j=0; j<5; j++)
  { s=s+marks[j]; }
  a=s/50;
  printf("sum, s)
  scanf(Avg, a)
  return 0;
}
```

FABONACII NO.

with recussion

```
#include<stdio.h>
int series(int n);
int main()
int
{ for(int i=0;i<=n,i++)
  { printf("%d", series(n));
  }
}
int sum(int n)
{
  if (n==1)
    return 0;
  else if (n==2)
    return 1;
  else
  { return series(n-1)+series(n-2);
  }
```

without

```
#include<stdio.h>
int main()
{ int n1=0, n2=1, n3, num;
printf("no.");
scanf("%d", &num);
printf("%d %d", n1, n2);
for(i=2, i<num, ++i)
{
  n3=n1+n3
  printf("%d", n3);
n1=n2;
n2=n3;
}
return 0;
}
```

SUM & AVERAGE.

with / without recursions.

5 digit

```
#include<>
#include<>
int sum(int n)
{
  if(n==0)
  {
    return 0;
  }
  else
  {
    int k = n%10
    return k+sum(n/10);
  }
}
int main()
{
  int a;
  printf(
  scanf
  int x = sum(a);
  printf("sum = %d", x);
  return 0;
}
```

sum & avg

```
#include
void avgsum(int a, int b)
{
  printf("sum is %d", a+b)
  printf("avg is %f", (float)(a+b)/2);
}
int main()
{
  int a, b;
  scanf("%d", &a);
  scanf("%d", &b);
  avgsum(a,b);
  return 0;
}
```

multiply 2 no. using recursion.

```
#include<
int mul(int a, int b);
int main()
{
  int
  printf()
  scan()
  int x = mul(a,b);
  printf("", x);
  return 0;
}
int mul(int a, int b)
{
  if(a<b)
  {
    return mul(b,a);
  }
  else if(b!=0) { return a+mul(a,b-1); }
  else { return 0; }
}
```

is prime → 0 → 1

```
#include
int main()
{
  int n;
  scanf
  printf("%d", isprime(n))
  return 0;
}
int isprime(int n)
{
  if(n==0||n==1)
  {return 0;}
  for(int i=2, i<n, i++)
  {
    if(n%i==0)
    {return 0;}
  }
  return 1;
}
```

Prime.

```
printf (no)
scan (  ) &num
printf (factor) num
for (i=2, i<num, i++)
{   if (num % i == 0)
    {
        isPrime=1;
        for (j=2, j<=i/2, j++)
        {
            if (i%j == 0)
            {  isPrime=0
               break;
            }
        }
        if(isPrime == 1)
        { printf ("%d", i); }
    }
}
return 0;
}
```

Lab 7

Q1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

```
#include <stdio.h>
int c;
int add (int a, int b)
{ c = a+b;
return c;
}
int sub (int a, int b)
{ if (a>b)
c = a-b;
return c;
}
int mul (int a, int b)
{ c = a*b;
return c
}
int power (int a, int b){
int power =1
for( int i = 1, i<=b, i++)
{
power = a * pow;
}
return power.

int fact (int a)
{ if a==0
return 1;
else.
for (i=a, i>=1, i--)
{fact = fact * i }
. return fact } },
1 return 0.
```

```
int main ()
printf
(enter the
addon.

scanf

switch (n)
{

case 1 : printf
scanf
printf ("%d", sub
add(a, b).
```

```
#include
void swap (int *a, int *b);
int main() {
int x=3, y=5;
swap (&x, &y);
printf ("x = %d & y = %d \n", x, y); return 0; }.
void swap (int *a, int *b)
{
int t = *a;
*a = *b;
*b = t;
}
```

Swap through pointer

Call by reference →

```
void swap (int, int);
int main()
{ int x, y;
printf
scanf
printf (Before swap
swap (x, y);
printf (After swap x, y
return 0;
void (int a, int b)
{ int temp;
temp = b
b = a
a = temp
printf ( ,a,b
```

Swap through call by value.

table.

```
#include <stdio.h>
void table (int n)
{ for (int i=1; i<=10; i++)
{ printf ("%d * %d", i, n, i*n);
}
}
int main()
{ int n;
scanf ("%d", &n);
table (n);
return 0;
};
```

x = 5

y = 5 10

value of a & b 5, 10

after x = 5

y = 10.

```
void ( )
{ int temp
{
printf swapped are %d %d", a, b)
};
return 0;
```

```
#include
void swap (int a, int b);
int main()
{ int a, b;
printf ( )
scanf
swap (a, b);
return 0;
}.
```

Recursion

1 2 3 4 5 — 5 4 3 2 1

```
#include <>
void printseries(int num);
int main()
{ int num;
  printf()
  scanf()
} printseries(num);
  return 0;
```

same.

```
void printseries(int num)
{
  if(num==1)
  { printf("1"); }
  else
  { printf("%d", num);
    printseries(num-1);
  }
}
```

```
if (num==5)
  printf("5");
else
{ printf("%d", num);
  printseries(n+1);
}
```

void.

Factorial

int type

```
#include
int fact int(n);
int main {
printf("fact is %d", fact(4));
return 0;
}
int fact(int n)
if(n==0)
{return 1;
}
int fact N = fact(n-1) * n
  return fact N;
}
```

sum:

```
#include
int sum(int n);
int main()
{ printf("sum is %d", sum(5)
  return 0; }
int sum(int n)
{ if (n==1)
  { return 1; }
int sum N = sum(n-1) + n;
  return sum N;
}
```