1. PALINDROME :-

```
-> main ()
{int m, n, S=0, x;
 printf ("enter n");
 scanf ("%d", &n);
   x = n;
 while (n != 0)
 {
    m = n % 10;
     S = S * 10 + m;
  n = n/10;
 }
 if (x == S)
 printf ("Palindrome");
 else
 printf ("Not a Palindrome");
}
```

2. REVERSE :-

```
-> main ()
{
  int m, n, S=0;
  printf ("enter n");
  scanf ("%d", &n);
   while (n != 0)
   {
      m = n % 10;
       S = S * 10 + m;
      n = n/10;
   }
  printf ("%d", S);
}
```

3. FACTORIAL :-

→
```
main ()
{
  int i, n, fact = 1;
  printf ("enter n: ");
  scanf ("%d", &n);
  i = 1;
  while (i <= n)
  {
   fact = fact * i;
     i++;
  }
  printf ("%d", fact);
}
```

TYPES OF FUNCTIONS :-

| ARGUMENTS | RETURN |
|---|---|
| without | without |
| with | without |
| without | with |
| with | with |

1. PALINDROME:-

I. Without arguments without return

-> void Palindrome ( )

```
{
  int m, n, s = 0, x;
  printf ("enter n");
  scanf ("%d", &n);
    x = n;
  while (n != 0)
  {
     m = n % 10;
     s = s * 10 + m;
     n = n / 10;
  }
  if (x == s)
  printf ("Palindrome");
  else
  printf ("not a Palindrome");
}
int main ( )
{
  Palindrome ( );
}
```

II. With arguments without return

→

```
Void Palindrome (int n)
{
  int m, s=0, x;
   x = n;
  while (n!=0)
  {
    m = n % 10;
    x = m
    s = s * 10 + m;
    n = n / 10;
  }
  if (x==s)
   printf (" Palindrome \n ");
   else
   printf (" Not Palindrome \n ");
}

int main()
{
   int n;
   printf (" Enter n: ");
   scanf (" %d ", &n);
   Palindrome (n);
}
```

(III). Without arguments with return

```
-> int Palindrome ()
{
    int m, n, s=0, x;
    printf ("Enter n: ");
    scanf ("%d", &n);
    x = n;
    while (n != 0)
    {
        m = n % 10;
        s = s * 10 + m;
        n = n / 10;
    }
    return x == s;
}
int main ()
{
    if (Palindrome ())
    printf ("Palindrome");
    else
    printf ("Not Palindrome");
}
```

(IV) With arguments with return :-

->

```
int Palindrome ( int n)
{
  int m , s = 0, x;
    x = n;
  while (n != 0)
  {
     m = n % 10;
     s = s * 10 + m;
     n = n / 10;
  }
  return x == s;
}
int main ( )
{
   int n;
   printf ("Enter n: ");
   scanf ("%d", &n);
   if ( Palindrome ( n ) )
      printf (" Palindrome ");
   else
      printf (" Not Palindrome ");
}
```

2. REVERSE A GIVEN NUMBER:-

I. without arguments without return

-> void reverse()

```
{ int m, n, s=0;
  printf("enter n: ");
  scanf("%d", &n);
  while(n!=0)
  {
    m = n %10;
    s = s*10 + m;
    n = n/10;
  }
  printf("reversed number is: %d \n", s);
}
int main()
{
  reverse();
}
```

(II) · With arguments without return

```
=> void reverse (int n)
{
   int m, s = 0;
   while (n != 0)
   {
      m = n % 10;
      s = s * 10 + m;
      n = n / 10;
   }
   printf ("reversed number is : %d \n", s);
}
int main()
{
   int n;
   printf ("enter n : ");
   scanf ("%d", &n);
   reverse (n);
}
```

(11). Without arguments with return

-> int reverse ( )

```
{
  int m, n, s = 0;
  printf (" enter n: ");
  scanf (" %d", &n);
  while ( n != 0 )
  {
    m = n % 10;
    s = s * 10 + m;
    n = n / 10;
  }
  return s;
}
int main ( )
{
  int reversed = reverse ( );
  printf (" Reversed number is : %d \n", reversed);
}
```

(IV) With arguments with return :-

->
```
int reverse (int n)
{
    int m, s = 0;
    while (n != 0)
    {
        m = n % 10;
        s = s * 10 + m;
        n = n / 10;
    }
    return s;
}
int main()
{
    int n;
    printf ("enter n: ");
    scanf ("%d", &n);
    int reversed = reverse();
    printf ("Reversed number is: %d\n", reversed);
}
```

3. FACTORIAL Of n-integers :-

I. Without arguments without return

-> void factorial ()

```
{
  int i, n, fact = 1;
  printf ("enter n: ");
  scanf ("%d", &n);
  i = 1;
  while (i <= n)
  {
    fact = fact * i;
    i++;
  }
  printf ("Factorial is %d \n", fact);
}
int main ()
{
  factorial ();
}
```

II. With arguments without return

→

```
void factorial (int n)
{
    int i, fact = 1;
    i = 1;
    while (i<=n)
    {
        fact = fact * i;
        i++;
    }
    printf ("Factorial is : %d \n", fact);
}
int main ()
{
    int n;
    printf ("enter n : ");
    scanf ("%d", &n);
    factorial (n);
}
```

(11) - without arguments with return

-> int factorial ( )

```
{ int i, n, fact = 1;
  printf ("enter n: ");
  scanf ("%d", &n);
   i = 1;
  while (i <= n)
  {
     fact = fact * i;
      i++;
  }
  return fact;
}
int main ()
{
   int fact = factorial ();
    printf ("Factorial is: %d \n", fact);
}
```

IV With arguments with return :-

-> 
```
int factorial (int n)
{
    int i, fact = 1;
    i = 1;
    while ( i <= n)
    {
        fact = fact * i;
        i++;
    }
    return fact;
}
int main()
{
    int n;
    printf ("Enter n: ");
    scanf ("%d", &n);
    int fact = factorial (n);
    printf ("Factorial is : %d \n", fact);
}
```