

## Recursion:-

→ ① factorial.

```
#include <iostream>
using namespace std;

int factorial (int num).
```

```
{ if (n==0)
  return 1;
```

⇒ Base condition.

else

```
return n * factorial (n-1);
}
```

→ ② fibonacci Series.

```
#include <iostream>
using namespace std;

int fibonacci (int n)
```

```
{ if (n <= 1)
  return n;
```

⇒ Base condition.

else

```
{ return fibonacci (n-1) + fibonacci (n-2);
}
}
```



→ ③ Sum of digits

```
#include <iostream>
using namespace std;
int sumofdigits (int n)
```

```
{
    if (n == 0)
        return n;
```

else

```
    return (n % 10) + sumofdigits (n / 10);
```

$\frac{425}{10} \rightarrow 42$   
 $\frac{42}{10} \rightarrow 4$   
 $\frac{4}{10} \rightarrow 0$   
5  
int

$\frac{425}{10} \rightarrow 42$   
 $\frac{42}{10} \rightarrow 4$   
 $\frac{4}{10} \rightarrow 0$   
5  
int

→ ④ Calculate  $x^y$ .

```
double power (double x, double y)
```

```
if (y == 0)
```

```
    return 1;
```

```
else if (y > 0)
```

```
    return x * power (x, y - 1);
```

```
else
```

```
    return 1 / power (x, -y)
```



bool is palindrome (str, int start, int end)

if (start >= end)

return true;

else if (str[start] != str[end])

{ return false; }

return is palindrome (str, start+1, end-1);

}



→ ⑤ Reverse a string.

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
void reversed string ( string &str , int start, int end)
```

```
{ if (start >= end)
```

```
    return;
```

```
    else
```

```
    { swap (str[start], str[end]);
```

```
      reverse string (str, start + 1, end - 1); }
```

```
int main()
```

```
{ string str;
```

```
  cout << "Enter a string" << endl;
```

```
  getline (cin, str);
```

```
  reverse string (str, 0, str.length - 1) → handle Null;
```

```
  cout << "reverse string: " << str;
```

```
}
```