1- یک تابع به نام add بنویسید که دو عدد صحیح دریافت کند و جمع آن‌ها را برگرداند. حاصل توسط تابع main نشان داده شود.

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

int add(int a, int b)}
   return a + b;
{

int main} ()
   int x, y;
   cout << "Enter two numbers;" :
   cin >> x >> y;
   cout << "Sum: " << add(x, y) << endl;
   return 0;
{
```

2- یک تابع به نام print بنویسید که عدد صحیحی دریافت کند و جدول ضرب آن عدد را تا ۱۰ در خروجی چاپ کند. نمایش جدول ضرب توسط تابع print باشد.

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

void printMultiplicationTable(int num) {
   for (int i = 1; i <= 10; i++) {
      cout << num << " x " << i << " = " << num * i << endl;
   }
}

int main() {
   int number;
   cout << "Enter a number to print its multiplication table: ";
   cin >> number;
   printMultiplicationTable(number);
   return 0;
}
```

3- یک تابع به نام isPrime بنویسید که عددی صحیح دریافت کند و بررسی کند آیا آن عدد اول است یا خیر. اگر اول بود، مقدار true و در غیر این صورت مقدار false برگرداند. تابع isPrime مقدار بولین false یا true برگرداند و خروجی توسط تابع main باشد.

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

bool isPrime(int n) {
   if (n <= 1) return false;
   for (int i = 2; i <= n / 2; i++) {
      if (n % i == 0) return false;
   }
   return true;
}

int main() {
   int num;
   cout << "Enter a number: ";
   cin >> num;
   if (isPrime(num)) {
      cout << num << " is a prime number." << endl;
   } else {
      cout << num << " is not a prime number." << endl;
   }
   return 0;
}
```

4- یک تابع به نام factorial بنویسید که عددی صحیح دریافت کند و فاکتوریل آن را محاسبه کند.

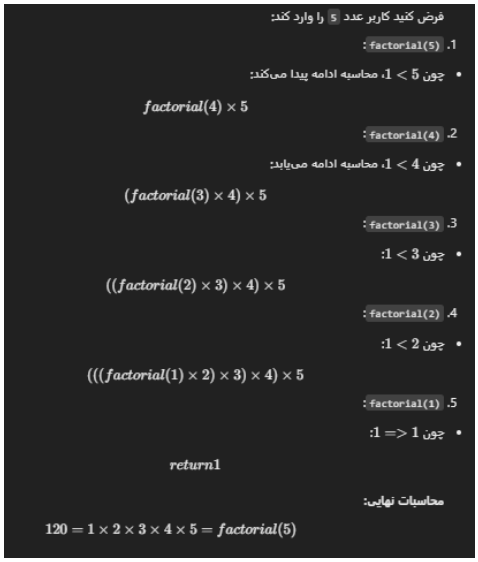
```
#include <iostream>
using namespace std;

int factorial(int n) {
   if (n <= 1) return 1;
   return n * factorial(n - 1);
}

int main() {
   int num;
   cout << "Enter a number: ";
   cin >> num;
   cout << "Factorial: " << factorial(num) << endl;
   return 0;
}
```

توضیحات:

- **ورودی :**تابع factorial یک عدد صحیح n می‌گیرد.
- **بازگشتی :(Recursion)** تابع خودش را صدا می‌زند، به شرطی که n > 1 باشد.
- **شرط پایان :(Base Case)** وقتی n <= 1 شود، تابع مقدار 1 را برمی‌گرداند و بازگشت متوقف می‌شود.
- **محاسبه :**هر بار مقدار nدر نتیجه فاکتوریل باقی‌مانده ((factorial(n - 1) ضرب می‌شود.

تابع main:

- **ورودی :**عدد numاز کاربر گرفته می‌شود.
- **فراخوانی تابع :**مقدار num به تابع factorial ارسال می‌شود.
- **خروجی :**مقدار بازگشتی از تابع فاکتوریل (num) محاسبه و نمایش داده می‌شود.



نسخه غیربازگشتی :

```
int factorialIterative(int n) {
   int result = 1;
   for (int i = 1; i <= n; i++) {
      result *= i;
   }
   return result;
}
```

بازگشت به این شکل است:
factorial(5)منتظر نتیجه factorial(4) می‌ماند.
factorial(4)منتظر نتیجه factorial(3) می‌ماند.
...
factorial(1) مقدار 1 را برمی‌گرداند.

5- یک تابع به نام printTriangle بنویسید که یک عدد صحیح دریافت کند و یک مثلث متشکل از ستاره‌ها (*) با ارتفاع داده‌شده چاپ کند. نمایش توسط خود تابع printTriangle باشد.
توضیح اگر ورودی 5 باشد خروجی به شکل زیر باشد:

*

**

***

****

*****

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

void printTriangle(int height) {
   for (int i = 1; i <= height; i++) {
      for (int j = 1; j <= i; j++) {
         cout << "*";
      }
      cout << endl; // رفتن به خط بعد
   }
}

int main() {
   int n;
   cout << "Enter the height of the triangle: ";
   cin >> n;
   printTriangle(n);
   return 0;
}
```