

این کد یک تابع توان رساندن پیاده‌سازی می‌کند و از تابع Power برای محاسبه توان یک عدد استفاده می‌کند. در زیر توضیحی کامل درباره هر خط کد آورده شده است:

```
#include <iostream>
```

این دستور، کتابخانه <iostream> را وارد می‌کند که امکان ورود و خروج از استاندارد ورودی/خروجی را فراهم می‌کند.

```
double Power(double base, int exponent) {  
    if (exponent == 0)  
        return 1;  
    else if (exponent < 0)  
        return 1 / (base * Power(base, -exponent - 1));  
    else  
        return base * Power(base, exponent - 1);  
}
```

این بخش شامل تعریف تابع Power است. این تابع دو ورودی به عنوان پایه (base) و توان (exponent) را دریافت می‌کند و توان رساندن مقدار پایه را به توان مشخص شده انجام می‌دهد. اگر توان برابر با صفر باشد، یک برمی‌گرداند. در صورتی که توان کوچکتر از صفر باشد، مقدار یک بر توان منفی با پایه و توان منفی منهای یک را برمی‌گرداند. در غیر این صورت، حاصل ضرب پایه در تابع توان پایه و توان کم یک را برمی‌گرداند.

```
void CheckResult(const std::string& testName, double result, double  
expectedResult) {  
    std::cout << "Test: " << testName << std::endl;  
    std::cout << "Result: " << result << std::endl;  
    std::cout << "Expected Result: " << expectedResult << std::endl;  
    std::cout << "Test " << (result == expectedResult ? "Passed" : "Failed") <<  
std::endl;  
    std::cout << std::endl;  
}
```

این بخش شامل تعریف تابع CheckResult است که دریافت کننده نام تست (testName)، نتیجه (result) و نتیجه مورد انتظار (expectedResult) است. این تابع نتیجه را به همراه مقدار مورد انتظار و نتیجه مشخص شده (Passed) یا (Failed) در خروجی نمایش می‌دهد.

```
void RunTests() {  
    double powerResult = Power(2,  
3);  
    CheckResult("Power Test", powerResult, 8);  
  
    powerResult = Power(-2, 3);  
    CheckResult("Power Test with Negative Base", powerResult, -8);  
  
    powerResult = Power(2, -3);  
    CheckResult("Power Test with Negative Exponent", powerResult, 0.125);  
}
```

این بخش شامل تابع RunTests است که برای اجرای تست‌های واحد تعریف شده است. در این تابع، توابع Power با ورودی‌های مختلف فراخوانی می‌شوند و نتیجه‌ی هر تست با استفاده از تابع CheckResult بررسی می‌شود.

```
int main() {
    RunTests();

    char choice;
    do {
        double base;
        int exponent;

        std::cout << "Enter the base number: ";
        std::cin >> base;

        std::cout << "Enter the exponent: ";
        std::cin >> exponent;

        double result = Power(base, exponent);

        std::cout << "Result: " << result << std::endl;

        std::cout << "Do you want to continue? (y/n): ";
        std::cin >> choice;
    } while (choice == 'y' || choice == 'Y');

    return 0;
}
```

در این بخش، تابع main() تعریف شده است. در ابتدا تابع RunTests برای اجرای تست‌ها صدا زده می‌شود. سپس یک حلقه تکراری ایجاد می‌شود که کاربر را به ترتیب پایه و توان ورودی می‌گیرد. سپس تابع Power با استفاده از مقادیر ورودی فراخوانی شده و نتیجه‌ی آن در متغیر result ذخیره می‌شود و در نهایت نتیجه به همراه متن "Result: " نمایش داده می‌شود. سپس کاربر را از ادامه‌ی عملیات سوال می‌شود و در صورت تایید (y) یا (Y) حلقه تکرار ادامه می‌یابد و در غیر این صورت برنامه خاتمه می‌یابد. در انتها مقدار صفر به عنوان خروجی تابع main برگردانده می‌شود. این بدان معناست که برنامه بدون مشکل اجرا شده است.