Polyfunc函数的构建与调用。刘鹏，学号20151910042。

% filename: QuickSort.m

% Sort the vector x, an old but efficient function,

% especially in solving the random sequence sorting problem.

**function** y **=** QuickSort**(**x**)**

A **=** **[];**

B **=** **[];**

C **=** **[];**

**if(**length**(**x**)** **<=** 1**)**

y **=** x**;**

**return**

**else**

a **=** floor**(**length**(**x**)/**2**);**

**for** i **=** 1**:**length**(**x**)**

**if(**x**(**i**)** **<** x**(**a**))**

A **=** **[**A**,**x**(**i**)];**

**elseif(**x**(**i**)** **==** x**(**a**))**

B **=** **[**B**,**x**(**i**)];**

**else**

C **=** **[**C**,**x**(**i**)];**

**end**

**end**

**end**

A **=** QuickSort**(**A**);**

C **=** QuickSort**(**C**);**

y **=** **[**A**,**B**,**C**];**

**end**

% filename: PolyFun.m

% This function could compute the functions with the following form

% y = a\_0 \* x^0 + a\_1 \* x^1 + ... + a\_n \* x^n

% where x is an constant produced by the rand built-in function, and the

% matrix [a\_0, a\_1, ..., a\_n] should be input as an arguement.

% Try to make the computing complexitity as small as possible.

**function** y **=** PolyFun**(**coe**,**x**)**

n **=** length**(**coe**);**

% a is the size of the matrix coe

**if(**n **==** 1**)**

y **=** coe**;**

**else**

y **=** coe**(**1**)** **+** x **\*** PolyFun**(**coe**(**2**:**length**(**coe**)),**x**);**

**end**

**end**

clc

a **=** input**(**'Please input the coefficient matrix:\n'**);**

**for** i **=** 1**:**10

x**(**i**)** **=** randi**([**1**,**50**]);**

y**(**i**)** **=** PolyFun**(**a**,**x**(**i**));**

**end**

x

Sorted **=** QuickSort**(**y**)**

Main函数如上。