```
// C++ program for Picard's iterative method

#include <iostream>
#include <cmath>

using namespace std;

// required macros defined below:
#define Y1(x) (1 + (x) + pow(x, 2) / 2)
#define Y2(x) (1 + (x) + pow(x, 2) / 2 + pow(x, 3) / 3 + pow(x, 4) / 8)
#define Y3(x) (1 + (x) + pow(x, 2) / 2 + pow(x, 3) / 3 + pow(x, 4) / 8 +
pow(x, 5) / 15 + pow(x, 6) / 48)

int main()
{
    double start_value = 0, end_value = 3,
           allowed_error = 0.4, temp;
    double y1[30], y2[30], y3[30];
    int count;

    for (temp = start_value, count = 0;
         temp <= end_value;
         temp = temp + allowed_error, count++) {

        y1[count] = Y1(temp);
        y2[count] = Y2(temp);
        y3[count] = Y3(temp);
    }

    cout << "\nX\n";
    for (temp = start_value;
         temp <= end_value;
         temp = temp + allowed_error) {

        // considering all values
        // upto 4 decimal places.
        cout << temp;
    }

    cout << "\n\nY(1)\n" ;
    for (temp = start_value, count = 0;
         temp <= end_value;
         temp = temp + allowed_error, count++) {

        cout << y1[count];
    }

    cout << "\n\nY(2)\n";
    for (temp = start_value, count = 0;
         temp <= end_value;
         temp = temp + allowed_error, count++) {

        cout << y2[count];
    }

    cout << "\n\nY(3)\n";
    for (temp = start_value, count = 0;
         temp <= end_value;
         temp = temp + allowed_error, count++) {

        cout << y3[count];
    }
    return 0;
}
```