

```

1  // C program for Picard's iterative method
2
3  #include <math.h>
4  #include <stdio.h>
5  #define Y1(x) (1 + (x) + pow(x, 2) / 2)
6  #define Y2(x) (1 + (x) + pow(x, 2) / 2 + pow(x, 3) / 3 + pow(x, 4) / 8)
7  #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)
8
9  int main()
10 {
11     double start_value = 0, end_value = 3,
12           allowed_error = 0.4, temp;
13     double y1[30], y2[30], y3[30];
14     int count;
15
16     for (temp = start_value, count = 0;
17         temp <= end_value;
18         temp = temp + allowed_error, count++) {
19
20         y1[count] = Y1(temp);
21         y2[count] = Y2(temp);
22         y3[count] = Y3(temp);
23     }
24
25     printf("\nX\n");
26     for (temp = start_value;
27         temp <= end_value;
28         temp = temp + allowed_error) {
29
30         printf("%.4lf ", temp);
31     }
32
33     printf("\n\nY(1)\n");
34     for (temp = start_value, count = 0;
35         temp <= end_value;
36         temp = temp + allowed_error, count++) {
37
38         printf("%.4lf ", y1[count]);
39     }
40
41     printf("\n\nY(2)\n");
42     for (temp = start_value, count = 0;
43         temp <= end_value;
44         temp = temp + allowed_error, count++) {
45
46         printf("%.4lf ", y2[count]);
47     }
48
49     printf("\n\nY(3)\n");
50     for (temp = start_value, count = 0;
51         temp <= end_value;
52         temp = temp + allowed_error, count++) {
53
54         printf("%.4lf ", y3[count]);
55     }
56     return 0;
57 }

```