

```
1 //
2 // file: integrate.c
3 // Created by hfwei on 2023/12/13.
4 // A nice function pointer example on Riemann integration:
5 // https://en.wikipedia.org/wiki/Function\_pointer
6 //
7
8 #include <stdio.h>
9 #include <math.h>
10
11 #define NUM_OF_PARTITIONS 100000
12
13 double Integrate(double low, double high, double (*func)(double));
14
15 double Square(double x);
16
17 int main() {
18     double low = 0.0;
19     double high = 1.0;
20     double integration = 0.0;
21
22     // gcc -pedantic (invalid application of sizeof to a function type)
23     // See "Function to pointer conversion" (https://en.cppreference.com/w/c/language/conversion)
24     // See also https://en.cppreference.com/w/c/language/sizeof
25     printf("sizeof sin: %zu\n", sizeof(sin));
26     printf("sizeof &sin: %zu\n", sizeof(&sin));
27
28     integration = Integrate(low, high, sin);
29     printf("sin(x) from %f to %f is %f\n", low, high, integration);
30
31     integration = Integrate(low, high, cos);
32     printf("cos(x) from %f to %f is %f\n", low, high, integration);
33
34     integration = Integrate(low, high, Square);
35     printf("Square(x) from %f to %f is %f\n", low, high, integration);
36
37     double (*funcs[])(double) = { sin, cos, Square };
38
39     int len = sizeof(funcs) / sizeof(*funcs);
40     for (int i = 0; i < len; ++i) {
41         printf("integration: %f\n", Integrate(low, high, funcs[i]));
42     }
43
44     return 0;
45 }
46
47 double Integrate(double low, double high, double (*func)(double)) {
48     double interval = (high - low) / NUM_OF_PARTITIONS;
49
50     double sum = 0.0;
51     for (int i = 0; i < NUM_OF_PARTITIONS; i++) {
52         double xi = low + interval * i;
```

```
53     double yi = func(xi);
54     sum += yi * interval;
55 }
56
57 return sum;
58 }
59
60 double Square(double x) {
61     return x * x;
62 }
```