

# 【18-4作业】 孙铭泽

## 1.欧拉33题

```
1  #include<stdio.h>
2  int gcd (int a, int b) {
3      if (b > a) {
4          return gcd(b, a);
5      }
6      if (b == 0) {
7          return a;
8      } else
9          return gcd(b, a % b);
10 }
11 int main() {
12     int x = 1, y = 1;
13     for (int i = 10; i < 99; i++) {
14         for (int j = i + 1; j < 99; j++) {
15             int a = i / 10;
16             int b = i % 10;
17             int c = j / 10;
18             int d = j % 10;
19             if(b == c && i * d == j * a && d != 0 ) {
20                 printf("%d %d\n",i , j);
21                 x *= a;
22                 y *= d;
23             }
24         }
25     }
26     printf("%d\n",y / gcd(x , y));
27     return 0;
28 }
```

## 2.欧拉35题

```
1  #include<stdio.h>
2  #include<math.h>
3  #define MAX_N 1000000
4  int is_prime[MAX_N] = {0};
5  void init() {
6      for(int i = 2; i * i < MAX_N; i++) {
7          if(is_prime[i]) continue;
8          for(int j = i * i; j < MAX_N; j += i) {
9              is_prime[j] = 1;
10         }
11     }
12 }
```

```

13 int reve (int x) {
14     int num = x, n = 0, m = 1;
15     int judg;
16     while (x) {
17         n++;
18         x /= 10;
19         m *= 10;
20     }
21     while (n--) {
22         judg = num % 10;
23         judg = judg * (m / 10) + num / 10;
24         num = judg;
25         if (is_prime[num])
26             return 0;
27     }
28     return 1;
29 }
30 int main() {
31     init();
32     int ans = 0;
33     for (int i = 2; i < 1000000; i++) {
34         if (!is_prime[i]) {
35             if (reve(i)) {
36                 ans++;
37                 printf("%d\n", i);
38             }
39         }
40     }
41     printf("%d\n", ans);
42     return 0;
43 }

```

### 3.欧拉37题

```

1  #include<stdio.h>
2  #include<math.h>
3  #define MAX_N 1000000
4  int is_prime[MAX_N] = {0};
5  void init() {
6      for(int i = 2; i * i < MAX_N; i++) {
7          if(is_prime[i]) continue;
8          for(int j = i * i; j < MAX_N; j += i) {
9              is_prime[j] = 1;
10         }
11     }
12 }
13 int reve(int x) {
14     int num = x, num1 = x, n = 0, m = 1;
15     int judg;
16     while(x) {
17         n++;
18         x /= 10;

```

```

19     m *= 10;
20 }
21 m /= 10;
22 for(int i = 0; i < n - 1; i++) {
23     judg = num / m;
24     judg = num - judg * m;
25     num = judg;
26     m /= 10;
27     if(num <= 1)
28         return 0;
29     if(is_prime[num])
30         return 0;
31 }
32 for(int i = 0; i < n - 1; i++) {
33     num1 /= 10;
34     if(num1 <= 1)
35         return 0;
36     if(is_prime[num1])
37         return 0;
38 }
39 return 1;
40 }
41 int main() {
42     init();
43     int ans = 0;
44     for(int i = 11; i < 1000000; i++) {
45         if(!is_prime[i]) {
46             if(reve(i)) {
47                 ans += i;
48                 printf("%d\n", i);
49             }
50         }
51     }
52     printf("%d\n", ans);
53     return 0;
54 }
55

```

## 4.欧拉43题

```

1  #include<iostream>
2  #include<algorithm> //next_permutation函数头文件
3  #include<cstdio>
4  using namespace std;
5  int prime[7] = {2, 3, 5, 7, 11, 13, 17}; //除数
6  long long int judg(int *a) {
7      if(a[0] == 0)
8          return 0; //九位数第一位不能为0
9      long long int ans = a[0];
10     for (int i = 1; i < 8; i++) {
11         int sum = a[i] * 100 + a[i + 1] * 10 + a[i + 2];
12         if(sum % prime[i - 1] != 0)

```

```

13         return 0;
14     }
15     for(int i = 1; i < 10; i++) ans = ans * 10 + a[i]; //返回值ans的计算
16     return ans;
17 }
18 int main() {
19     long long int ans = 0;
20     int a[10];
21     for(int i = 0; i < 10; i++) {
22         a[i] = i;
23     }
24     do {
25         ans += judg(a);
26     } while(next_permutation(a, a + 10)); //全排列函数使用
27     printf("%lld\n", ans);
28     return 0;
29 }
30

```

## 5.欧拉44题中的二分查找练习题

```

1  #include<stdio.h>
2  #include<inttypes.h>
3  int64_t cube(int x) {
4      return x * x * x;    //求立方
5  }
6  int judg_cube(int64_t x) {
7      int64_t left = 0, right = x;
8      if(x < 0) {
9          left = -x, right = 0; //判断x是否为负数，负数左右改为本行
10     }
11     while(right >= left) {
12         int64_t mid = (left + right) / 2; //求中间值
13         if(cube(mid) == x)
14             return 1; //如果相等代表找到
15         if(cube(mid) > x)
16             right = mid - 1; //中间值较大更新右边界
17         if(cube(mid) < x)
18             left = mid + 1; //中间值较小更新左边界
19     }
20     return 0;
21 }
22 int main() {
23     int n;
24     scanf("%d", &n);
25     if(judg_cube(n)) {
26         printf("YES");
27     } else {
28         printf("NO");
29     }
30     return 0;
31 }

```

