【18-4作业】 孙铭泽

1.欧拉33题

```c
#include<stdio.h>
int gcd (int a, int b) {
    if (b > a) {
        return gcd(b, a);
    }
    if (b == 0) {
        return a;
    } else
        return gcd(b, a % b);
}
int main() {
   int x = 1, y = 1;
    for (int i = 10; i < 99; i++) {
        for (int j = i + 1; j < 99; j++) {
            int a = i / 10;
            int b = i % 10;
            int c = j / 10;
            int d = j % 10;
            if(b == c && i * d == j * a && d != 0 ) {
                printf("%d %d\n",i , j);
            x *= a;
            y *= d;
            }
        }
    }
    printf("%d\n",y / gcd(x , y));
    return 0;
}
```

2.欧拉35题

```c
#include<stdio.h>
#include<math.h>
#define MAX_N 1000000
int is_prime[MAX_N] = {0};
void init() {
    for(int i = 2; i * i < MAX_N; i++) {
        if(is_prime[i]) continue;
        for(int j = i * i; j < MAX_N; j += i) {
            is_prime[j] = 1;
        }
    }
}
```

```c
int reve (int x) {
    int num = x, n = 0, m = 1;
    int judg;
    while (x) {
        n++;
        x /= 10;
        m *= 10;
    }
    while (n--) {
        judg = num % 10;
        judg = judg * (m / 10) + num / 10;
        num = judg;
        if (is_prime[num])
        return 0;
    }
    return 1;
}
int main() {
    init();
    int ans = 0;
    for (int i = 2; i < 1000000; i++) {
        if (!is_prime[i]) {
            if (reve(i)) {
                ans++;
                printf("%d\n",i);
            }
        }
    }
    printf("%d\n",ans);
    return 0;
}
```

3.欧拉37题

```c
#include<stdio.h>
#include<math.h>
#define MAX_N 1000000
int is_prime[MAX_N] = {0};
void init() {
    for(int i = 2; i * i < MAX_N; i++) {
        if(is_prime[i]) continue;
        for(int j = i * i; j < MAX_N; j += i) {
            is_prime[j] = 1;
        }
    }
}
int reve(int x) {
    int num = x,num1 = x, n = 0, m = 1;
    int judg;
    while(x) {
        n++;
        x /= 10;
```

```
        m *= 10;
    }
    m /= 10;
    for(int i = 0; i < n - 1; i++) {
        judg = num / m;
        judg = num - judg * m;
        num = judg;
        m /= 10;
        if(num <= 1)
        return 0;
        if(is_prime[num])
        return 0;
    }
    for(int i = 0; i < n - 1; i++) {
        num1 /= 10;
        if(num1 <= 1)
        return 0;
        if(is_prime[num1])
        return 0;
    }
    return 1;
}
int main() {
    init();
    int ans = 0;
    for(int i = 11; i < 1000000; i++) {
        if(!is_prime[i]) {
            if(reve(i)) {
                ans += i;
                printf("%d\n",i);
            }
        }
    }
    printf("%d\n",ans);
    return 0;
}

```

4.欧拉43题

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

```
            return 0;
    }
    for(int i = 1; i < 10; i++) ans = ans * 10 + a[i];//返回值ans的计算
    return ans;
}
int main() {
    long long int ans = 0;
    int a[10];
    for(int i = 0; i < 10; i++) {
        a[i] = i;
    }
    do {
        ans += judg(a);
    } while(next_permutation(a, a + 10));//全排列函数使用
    printf("%lld\n",ans);
    return 0;
}

```

5.欧拉44题中的二分查找练习题

```
#include<stdio.h>
#include<inttypes.h>
int64_t cube(int x) {
    return x * x * x;    //求立方
}
int judg_cube(int64_t x) {
    int64_t left = 0, right = x;
    if(x < 0) {
        left = -x, right = 0;//判断x是否为负数，负数左右改为本行
    }
    while(right >= left) {
        int64_t mid = (left + right) / 2;//求中间值
        if(cube(mid) == x)
            return 1;//如果相等代表找到
        if(cube(mid) > x)
            right = mid -1;//中间值较大更新右边界
        if(cube(mid) < x)
            left = mid + 1;//中间值较小更新左边界
    }
    return 0;
}
int main() {
    int n;
    scanf("%d", &n);
    if(judg_cube(n)) {
        printf("YES");
    } else {
        printf("NO");
    }
    return 0;
}
```