题目要求：<24-P1036-选数-题目要求.docx>

## 一、分析

这个题的关键在于怎么求从n中取k个数的组合。以下简单介绍取全组合的方法，这里采用递归。

其实这里的难点是：如何去重？

答案是：不降原则

保证枚举的这些数是升序排列

其实真正的不降原则还可以平

比如 1 2 2 3 3 4......

但是请注意这道题也不能平

否则就有重复数字了

代码中还是一样的老套路

一样的dfs

一样的参数

咱们最主要看其中不降原则的部分

在dfs函数中，我们明显能看到一个参数a

这是个什么东西呢？

就是

也就是最小的，符合不降原则的参数

这样一来，就可以去重了！

## 二、我的代码

附上文件路径：<24-P1036-选数.cpp>

1. #include<stdio.h>
2. #include<math.h>
3. int n,k;
4. int isprime(int n)
5. {
6. for(int i=2;i<=sqrt(n);i++)
7. if(n%i==0)return0;
8. return 1;
9. }*//判断是否为素数*
10. int create(int a,int k,int prior,int \*num)
11. {*//a指示当前的升序初始值，避免出现重复*
12. *//k指示剩余的需要再取的数*
13. *//prior为前面累加的和*
14. *//num为传入的数组*
15. int res=0;
16. if(!k) return isprime(prior);
17. *//如何需要再取的数已归零，则返回累加和是否为素数*
18. for(int i=a;i<n;i++)
19. res+=create(i+1,k-1,prior+num[i],num);
20. *//递归下一个过程，把当前的初始值加一，需再取的数减一*
21. *//同时更新已得到的累加和*
22. return res;
23. *//调用递归生成全组合，在过程中逐渐把K个数相加*
24. *//当选取的数个数为0时，直接返回前面的累加和是否为质数*
25. }
26. int main(void)
27. {
28. scanf("%d%d",&n,&k);
29. int num[n];
30. for(int i=0;i<n;i++)scanf("%d",&num[i]);
31. *//读入数据*
32. printf("%d",create(0,k,0,num));
33. }