

2025-4-1 日 递归专题练习

练习一：斐波那契数列

青牛最近正在学习斐波那契数列，斐波那契数列 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55.....从第三项起，每一项等于前面两项之和。现在请帮助青牛计算出斐波那契数列的第 n 项，通过递归函数实现。

输入格式

输入所求的项数。(样例 ： 10)

输出格式

输出数据项的值。(样例 ： 34)

参考程序：

```
#include<bits/stdc++.h>

using namespace std;

int solve(int n){

    if(n==1) return 0;

    else if(n==2) return 1;

    else return solve(n-1)+solve(n-2);

}

int main(){

    int n;

    cin>>n;

    cout<<solve(n);
```

```
    return 0;
}
```

练习二：最大公约数 GCD

输入 2 个正整数 A , B , 求 A 与 B 的最大公约数。

输入格式

2 个数 A,B , 中间用空格隔开。 ($1 \leq A, B \leq 10^9$)

输出格式

输出 A 与 B 的最大公约数。

参考程序：

```
#include<bits/stdc++.h>

using namespace std;

void gcd(int a,int b){

    if(b==0) cout<<a;

    else gcd(b,a%b);

}

int main(){

    int m,n;

    cin>>m>>n;

    gcd(m,n);

    return 0;

}
```

练习三：选数字

从 1-n 中选出若干个数 ,要求同样的数字不能重复选择 ,按照字典序正序输出所有方案。

输入格式

输入 1 个数 n , ($1 \leq n \leq 15$)

输出格式

按照字典序正序输出所有方案。

输入样例

3

输出样例

1

1 2

1 2 3

1 3

2

2 3

3

参考程序 1 :

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int nums[21], n;
```

```
void F(int min, int sel) {
```

```
    for (int i = 0; i < sel; i++)
```

```

        cout << nums[i] << " ";

    if(sel > 0)

        cout << endl;

    for(int i = min; i <= n; i++) {

        nums[sel] = i;

        F(i + 1, sel + 1);

    }

}

```

```

int main() {

    cin >> n;

    F(1, 0);

    return 0;

}

```

参考程序 2 :

```

#include <bits/stdc++.h>

using namespace std;

int nums[21], n;

void F(int cur, int sel) {

```

```

    if(cur > n) return;

    nums[sel] = cur;

    for (int i = 0; i <= sel; i++)

        cout << nums[i] << " ";

    cout << endl;

    F(cur + 1, sel + 1);

    F(cur + 1, sel);

}

```

```

int main() {

    cin >> n;

    F(1, 0);

    return 0;

}

```

练习四：选数字 2

已知 n 个整数 $X_1, X_2, \dots, X_n$, 以及 1 个整数 $k(k < n)$ 。从 n 个整数中任选 k 个整数相加 ,可分别得到一系列的和。例如当 $n = 4, k = 3$,4 个整数分别为 3,7,12,19 时 ,

可得全部的组合与它们的和为:

$$3+7+12=22$$

$$3+7+19=29$$

$$7+12+19=38$$

$$3+12+19=34$$

现在，要求你计算出和为素数共有多少种。例如上例，只有一种的和为素数：

$$3+7+19=29.$$

输入格式

第一行两个空格隔开的整数 n, k ($1 \leq n \leq 20, k < n$)

第二行 n 个整数，分别为 $x_1, x_2, \dots, x_n$ ($1 \leq x_i \leq 5 \times 10^6$)

输出格式

屏幕输出，格式为：1 个整数（满足条件的种数）。

输入样例

4 3

3 7 12 19

输出样例

1

参考程序：

```
#include<bits/stdc++.h>

using namespace std;

int a[25],b[25];

//不必设置数组 b，但是为了便于理解

//假设从数组 a 中选择 k 个数保存到数组 b 中

int n,k,cnt,ans=0;

//函数：判断是不是素数

bool isprime(int n){

    if(n<2) return false;
```

```

    for(int i=2;i*i<=n;i++)

        if(n%i==0) return false;

    return true;

}

//idx 代表正在选择的 b 的元素下标

//min 记录数组上一个元素的下标

//sum 记录已选择元素的总和

void solve(int idx,int min,int sum){

    if(idx==k){

        //判断总和是素数则 ans 加 1

        if(isprime(sum)) ans++;

        return;

    }

    else{

        for(int i=min;i<n;i++){

            //b[idx]=a[i];//此句测试，数组 b 更直观

            solve(idx+1,i+1,sum+a[i]);

            //solve(idx+1,i+1,sum+b[idx]);假设利用了数组 b 应该这么写(b[idx]=a[i])

        }

    }

}

int main(){

```

```

        cin>>n>>k;

        for(int i=0;i<n;i++) scanf("%d",&a[i]);

        solve(0,0,0);

        cout<<ans;

        return 0;

    }

```

练习五：分数字

将 n 分成若干个大于 0 的数的和，按照字典序输出所有方案。

输入格式

一个数 n ($1 \leq n \leq 15$)

输出格式

按照字典序，输出所有方案。数字之间，用空格划分。

输入样例

4

输出样例

```

1 1 1 1
1 1 2
1 2 1
1 3
2 1 1
2 2
3 1
4

```

参考程序：

```

#include<bits/stdc++.h>

using namespace std;

```

```

int a[20];

//idx 拆第几个数   n 还有多大的数要拆

void solve(int idx,int num){

    if(num==0) {

        for(int i=0;i<idx-1;i++) cout<<a[i]<<" ";

        cout<<endl;

        return;

    }

    for(int i=1;i<=num;i++){

        a[idx-1]=i;

        solve(idx+1,num-i);

    }

}

int main(){

    int n;

    cin>>n;

    solve(1,n);

    return 0;

}

```

练习六：分数字 2

将 n 分成 m 个大于 0 的不同数的和， $1\ 2$ 同 $2\ 1$ 视作相同的划分。按照字典序输出所有方案。数据保证存在解，即不会出现 $1+2+\dots+m > n$ 的情况。

输入格式

2 个数 n, m ($1 \leq m \leq 10, 1 \leq n \leq 50$)。 $1+2+\dots+m > n$

输出格式

按照字典序，输出所有方案。数字之间，用空格划分。

输入样例

13 3

输出样例

1 2 10
1 3 9
1 4 8
1 5 7
2 3 8
2 4 7
2 5 6
3 4 6

参考程序：

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int a[15];
```

```
int m,n;
```

```
void solve(int total,int sel){
```

```
    if(total==0){//需分的数是 0，代表数字 n 已被分成了 0
```

```
        if(sel==0) { //已分 m 个数，待分的个数为 0
```

```

        for(int i=1;i<=m;i++){
            cout<<a[i]<<" ";

            cout<<endl;

        }

        return;

    }

    for(int i=a[m-sel]+1;i<=total;i++){

        a[m-sel+1]=i;

        solve(total-i,sel-1);

    }

}

int main(){

    cin>>n>>m;

    solve(n,m);

    return 0;

}

```