// BFS 全排列 素数

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

p[i] = i;// 全排列 为 对数组排列 p[0]=0,p[1]=1,p[2]=2,p[3]=3...p[k]=k

while (next\_permutation(p, p + k));

//全排列公式 0 到 k a[i][0~~~k]

/\*

定义：divisor[i] ：以i为因子的最后一个数的下标；

用一个数组标记一个数，作为因子 最后出现在 第vis【i】个数

开始vis数组初始化为-1；

①对每个输入的数找出它的所有因子，

比较该因子是否在给定范围内也是其它数的因子；

②若不是，则累加计数；

③更新当前数的因子的位置为当前下标\

\*/

for(int j = 1; j \* j <= x; j++)

if(x % j == 0)

{

if(i - y > divisor[j]) // i - yi <= j < i 不满足已经存在的

res++;

if(j \* j != x && i - y > divisor[x / j])

res++;

divisor[j] = divisor[x / j] = i;

}

void bfs()

{

queue <Node> q;

memset(vis, false, sizeof(vis));

q.push(Node(1, 1, 0));

int len = 1;

while( !q.empty())

{

Node h = q.front();

int xx = h.x, yy = h.y, step = h.step;

if( vis[xx][yy] ) continue;

vis[xx][yy] = true;

rep(i, 0, 3)

{

int xi = xx+dir[i][0];

int yi = yy+dir[i][1];

if(check(xi, yi) && !vis[xi][yi] && g[xi][yi] == '.')

{

num[len] ++;

q.push(Node(xi, yi, len + 1 ));

}

}

}

}

priority\_queue<int,vector<int>,cmp1>que1;//最小值优先

priority\_queue<int,vector<int>,cmp2>que2;//最大值优先