9．プログラミング	1/2

迷路を縦か横にたどるC言語プログラムを考える。

ただし、以下を満たすものとする。

- 迷路は二次元整数配列mapで与えられる。
- スタート位置はmap[1][1]の地点とする。
- 与えられる迷路の各地点は、0なら通路、1なら壁、2ならゴールを意味する。
- 迷路の上下左右の端は必ず壁になっている。
- スタートからゴールまでの（後戻りやループなどの無駄のない）経路をたどることができれば終了してよい。
- ゴールまでの無駄のない経路に含まれる地点は、「リーチ可能」であると呼ぶことにする。
- プログラムの中で、通った地点には足跡3を付けていく。ただし、リーチ可能と分かった地点には2を書き込んでいく。

迷路を解くために、次ページのようなプログラムを考えた。このとき、次の問に答えよ。

1) 【ア】にあてはまる記述を答えよ。

2) 【イ】～【オ】にあてはまる記述（順不同）を答えよ。

3) reach(map, 1, 1)を実行したときの処理を、順を追って説明せよ。

```
// 【迷路探索プログラム】
#include <stdio.h>

/* 迷路を二次元整数配列 map で定義 */
int map[5][5] ={{1,1,1,1,1},
                {1,0,0,1,1},
                {1,0,1,1,1},
                {1,0,0,2,1},
                {1,1,1,1,1}};

/*迷路の y 行 x 列がリーチ可能かを調べ、可能なら 1，それ以外は 0 を返す関数 */
int reach(int m[5][5], int y, int x ) {
  if ( 【ア】 ) { //y 行 x 列が壁か足跡なら，
    return(0);
  }
  if (m[y][x] == 2) { //ゴールなら，
    return(1);
  }
  m[y][x] = 3; //足跡をつける．
  if ( 【イ】 || 【ウ】 || 【エ】 || 【オ】 ) {
  //上下左右のどれかがリーチ可能なら，
    m[y][x] = 2; // y 行 x 列もリーチ可能．
    return(1);
  }
  return(0); //リーチ可能ではない．
}

/* メインルーチン */
void main(){
  reach(map, 1, 1);
}
```