

[Dashboard](#) / [My courses](#) / [SFP:HK2-2019-2020](#) / [Tổng hợp](#) / [Trò chơi gõ mìn](#)

Started on	Friday, 15 May 2020, 12:08 PM
State	Finished
Completed on	Saturday, 20 June 2020, 10:53 PM
Time taken	36 days 10 hours
Marks	3.00/8.00
Grade	3.75 out of 10.00 (38%)

Question **1**

Correct

Mark 0.00 out of 1.00

Cho 2 mảng toàn cục B và T dùng để lưu bản đồ mìn và trạng thái đóng/mở của các ô trong trò chơi Minesweeper.

```
#define M 5
#define N 5
#define MINE -1
/*Mảng B lưu bảng đồ mìn*/
int B[M][N];
/*Mảng T lưu trạng thái các ô*/
int T[M][N];
```

Hãy viết [hàm](#) **init()** để khởi tạo bản đồ mìn B như hình vẽ:

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	-1	0	0
2	0	0	0	0	0
3	0	-1	0	0	0
4	0	0	0	0	0

Tất cả các ô đều có trạng thái 0 (chưa mở).

Khuôn dạng (prototype)

- Tên [hàm](#): init
- Tham số: không
- Kiểu trả về: void

Thân [hàm](#):

- Khởi tạo các giá trị cho mảng B và T theo yêu cầu.

Chú ý

- Dữ liệu đầu vào luôn hợp lệ.
- Chỉ viết [HÀM](#), không viết TOÀN BỘ CHƯƠNG TRÌNH

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1
2 void init(){
3     int i,j;
4     for(i = 0; i < N; i++){
5         for(j = 0; j < M; j++){
6             B[i][j] = 0;
7             T[i][j] = 0;
8         }
9     }
10    B[1][2] = MINE;
11    B[3][1] = MINE;
12 }
```

	Expected	Got	
--	----------	-----	--

	Expected	Got	
✓	<div>B = 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 T = 0</div>	<div>B = 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 T = 0</div>	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question **2**

Correct

Mark 0.00 out of 1.00

Cho 2 mảng toàn cục B và T dùng để lưu bản đồ mìn và trạng thái đóng/mở của các ô trong trò chơi Minesweeper.

```
#define M 5
#define N 5
#define MINE -1
/*Mảng B lưu bảng đồ mìn*/
int B[M][N];
/*Mảng T lưu trạng thái các ô*/
int T[M][N];
```

Hãy viết [hàm count_mines\(\)](#) để đếm số ô chứa mìn xung quanh các ô của bảng B.
Ví dụ: giả sử bảng đồ mìn B được khởi tạo như sau:

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	-1	0	0
2	0	0	0	0	0
3	0	-1	0	0	0
4	0	0	0	0	0

thì kết quả của việc đếm số mìn **trong 8 ô xung quanh của mỗi ô** sẽ như bên dưới:

	0	1	2	3	4
0	0	1	1	1	0
1	0	1	-1	1	0
2	1	2	2	1	0
3	1	-1	1	0	0
4	1	1	1	0	0

Trong 8 ô xung quanh ô (2, 1) có 2 trái mìn (ở ô (3,1) và (-1,2))

Khuôn dạng (prototype)

- Tên [hàm](#): count_mines
- Tham số: không
- Kiểu trả về: void

Thân [hàm](#):

- Đếm số mìn trên mảng B và ghi kết quả vào mảng B luôn.

Chú ý

- Dữ liệu đầu vào luôn hợp lệ.
- Chỉ viết [HÀM](#), không viết TOÀN BỘ CHƯƠNG TRÌNH

Gợi ý

- Xem slides

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 void count_mines(){
2     int i,j;
3     for(i = 0; i < N; i++){
4         for(j = 0; j < M; j++){
5             if(B[i][j] == MINE){
6                 if (B[i-1][j] != MINE) B[i-1][j]++;
7                 if (B[i+1][j] != MINE) B[i+1][j]++;
8                 if (B[i][j-1] != MINE) B[i][j-1]++;
9                 if (B[i][j+1] != MINE) B[i][j+1]++;
10                if (B[i-1][j-1] != MINE) B[i-1][j-1]++;
11                if (B[i-1][j+1] != MINE) B[i-1][j+1]++;
12                if (B[i+1][j-1] != MINE) B[i+1][j-1]++;
13                if (B[i+1][j+1] != MINE) B[i+1][j+1]++;
14            }
15        }
16    }
17 }
```

	Input	Expected	Got	
✓	0 0 0 0 0	0 1 1 1 0	0 1 1 1 0	✓
	0 0 -1 0 0	0 1 -1 1 0	0 1 -1 1 0	
	0 0 0 0 0	1 2 2 1 0	1 2 2 1 0	
	0 -1 0 0 0	1 -1 1 0 0	1 -1 1 0 0	
	0 0 0 0 0	1 1 1 0 0	1 1 1 0 0	
✓	0 0 0 0 0	0 1 1 1 0	0 1 1 1 0	✓
	0 0 -1 0 0	1 2 -1 2 1	1 2 -1 2 1	
	0 -1 0 -1 0	1 -1 4 -1 1	1 -1 4 -1 1	
	0 0 -1 0 0	1 2 -1 2 1	1 2 -1 2 1	
	0 0 0 0 0	0 1 1 1 0	0 1 1 1 0	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question **3**

Correct

Mark 0.00 out of 1.00

Cho 2 mảng toàn cục B và T dùng để lưu bản đồ mìn và trạng thái đóng/mở của các ô trong trò chơi Minesweeper.

```
#define M 5
#define N 5
#define MINE -1
/*Mảng B lưu bảng đồ mìn*/
int B[M][N];
/*Mảng T lưu trạng thái các ô*/
int T[M][N];
```

Hãy viết [hàm printMap1\(\)](#) để in bản đồ mìn theo quy tắc:

- Nếu ô B[i][j] == -1 (mìn), in ký tự 'x'
- Nếu ô B[i][j] == 0, in ký tự '.'
- Ngược lại in giá trị của B[i][j]

Ví dụ: giả sử bảng đồ mìn B như bên dưới:

	0	1	2	3	4
0	0	1	1	1	0
1	0	1	-1	1	0
2	1	2	2	1	0
3	1	-1	1	0	0
4	1	1	1	0	0

thì kết quả in ra sẽ là:

```
0 1 2 3 4
0 . 1 1 1 .
1 . 1 x 1 .
2 1 2 2 1 .
3 1 x 1 . .
4 1 1 1 . .
```

Khuôn dạng (prototype)

- Tên [hàm](#): printMap1
- Tham số: không
- Kiểu trả về: void

Thân [hàm](#):

- In bản đồ mìn B theo quy tắc trên.

Chú ý

- Dữ liệu đầu vào luôn hợp lệ.
- Chỉ viết [HÀM](#), không viết TOÀN BỘ CHƯƠNG TRÌNH
- Phải in luôn chỉ số hàng và chỉ số cột

Gợi ý

- Xem slides
- In hàng đầu (chỉ số cột)
- Dùng vòng lặp in từng hàng, với mỗi hàng in chỉ số hàng, sau đó in theo các giá trị của B[i][j].

For example:

Input	Result
0 1 1 1 0	0 1 2 3 4
0 1 -1 1 0	0 . 1 1 1 .
1 2 2 1 0	1 . 1 x 1 .
1 -1 1 0 0	2 1 2 2 1 .
1 1 1 0 0	3 1 x 1 . .
	4 1 1 1 . .

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 void printMap1(){
2     int i,j;
3     printf(" ");
4     for(j = 0; j < M; j++){
```

```
5         printf(" %d",j);
6     }
7     for(i = 0; i < N; i++){
8         printf("\n");
9         printf("%d",i);
10        for(j = 0; j < M; j++){
11            if(B[i][j] == MINE){
12                printf(" x");
13            }
14            else if(B[i][j] == 0){
15                printf(" .");
16            }
17            else
18                printf(" %d", B[i][j]);
19        }
20    }
21 }
```

	Input	Expected	Got	
✓	0 1 1 1 0 0 1 -1 1 0 1 2 2 1 0 1 -1 1 0 0 1 1 1 0 0	0 1 2 3 4 0 . 1 1 1 . 1 . 1 x 1 . 2 1 2 2 1 . 3 1 x 1 . . 4 1 1 1 . .	0 1 2 3 4 0 . 1 1 1 . 1 . 1 x 1 . 2 1 2 2 1 . 3 1 x 1 . . 4 1 1 1 . .	✓
✓	0 1 1 1 0 1 2 -1 2 1 1 -1 4 -1 1 1 2 -1 2 1 0 1 1 1 0	0 1 2 3 4 0 . 1 1 1 . 1 1 2 x 2 1 2 1 x 4 x 1 3 1 2 x 2 1 4 . 1 1 1 .	0 1 2 3 4 0 . 1 1 1 . 1 1 2 x 2 1 2 1 x 4 x 1 3 1 2 x 2 1 4 . 1 1 1 .	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question **4**

Correct

Mark 0.00 out of 1.00

Cho 2 mảng toàn cục B và T dùng để lưu bản đồ mìn và trạng thái đóng/mở của các ô trong trò chơi MineSweeper.

```
#define M 5
#define N 5
#define MINE -1
/*Mảng B lưu bảng đồ mìn*/
int B[M][N];
/*Mảng T lưu trạng thái các ô*/
int T[M][N];
```

Hãy viết [hàm printMap2\(\)](#) để in bản đồ mìn theo quy tắc (có quan tâm đến trạng thái đóng/mở của các ô):

- Nếu ô chưa mở => in ký tự '#'
- Ngược lại
 - Nếu ô đó chứa mìn => in 'x'
 - Nếu xung quanh không có mìn => in '.'
 - Ngược lại => in giá trị B[i][j] của ô đó

Khuôn dạng (prototype)

- Tên [hàm](#): printMap2
- Tham số: không
- Kiểu trả về: void

Thân hàm:

- In bản đồ mìn B theo quy tắc trên.

Chú ý

- Dữ liệu đầu vào luôn hợp lệ.
- Chỉ viết [HÀM](#), không viết TOÀN BỘ CHƯƠNG TRÌNH
- Phải in luôn chỉ số hàng và chỉ số cột

Gợi ý

- Xem slides
- In hàng đầu (chỉ số cột)
- Dùng vòng lặp in từng hàng, với mỗi hàng in chỉ số hàng, sau đó in theo các giá trị của B[i][j].

For example:

Input	Result
0 1 1 1 0	0 1 2 3 4
0 1 -1 1 0	0 . 1 1 # #
1 2 2 1 0	1 . 1 x # #
1 -1 1 0 0	2 1 # 2 # #
1 1 1 0 0	3 # x 1 # #
	4 1 1 1 # .
1 1 1 0 0	
1 1 1 0 0	
1 0 1 0 0	
0 1 1 0 0	
1 1 1 0 1	

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 void printMap2(){
2     int i,j;
3     printf(" ");
4     for(j = 0; j < M; j++){
5         printf(" %d",j);
6     }
7     for(i = 0; i < N; i++){
8         printf("\n");
9         printf("%d",i);
10        for(j = 0; j < M; j++){
11            if(T[i][j] == 0){
12                printf(" #");
13            }
14            else if(B[i][j] == MINE){
15                printf(" x");
16            }
17            else if(B[i][j] == 0){
18                printf(" .");
19            }
20            else printf(" %d", B[i][j]);
21        }
```



```
22 |      }
23 | }
```

	Input	Expected	Got	
✓	0 1 1 1 0 0 1 -1 1 0 1 2 2 1 0 1 -1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 0 1 0 0 0 1 1 0 0 1 1 1 0 1	0 1 2 3 4 0 . 1 1 # # 1 . 1 x # # 2 1 # 2 # # 3 # x 1 # # 4 1 1 1 # .	0 1 2 3 4 0 . 1 1 # # 1 . 1 x # # 2 1 # 2 # # 3 # x 1 # # 4 1 1 1 # .	✓
✓	0 1 1 1 0 1 2 -1 2 1 1 -1 4 -1 1 1 2 -1 2 1 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 1 0 1 0 0 0 1	0 1 2 3 4 0 # 1 1 # # 1 # # # # # 2 # # 4 x # 3 # 2 x 2 # 4 . # # # .	0 1 2 3 4 0 # 1 1 # # 1 # # # # # 2 # # 4 x # 3 # 2 x 2 # 4 . # # # .	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question **5**

Correct

Mark 1.00 out of 1.00

Cho 2 mảng toàn cục B và T dùng để lưu bản đồ mìn và trạng thái đóng/mở của các ô trong trò chơi Minesweeper.

```
#define M 5
#define N 5
#define MINE -1
/*Mảng B lưu bảng đồ mìn*/
int B[M][N];
/*Mảng T lưu trạng thái các ô*/
int T[M][N];
```

Hãy viết [hàm count_remain\(\)](#) để đếm số ô chưa mở

Khuôn dạng (prototype)

- Tên [hàm](#): count_remain
- Tham số: không
- Kiểu trả về: int

Thân [hàm](#):

- Đếm và trả về số ô chưa mở (ô chưa mở là ô có T[i][j] == 0).

Chú ý

- Dữ liệu đầu vào luôn hợp lệ.
- Chỉ viết [HÀM](#), không viết TOÀN BỘ CHƯƠNG TRÌNH

Gợi ý

- Xem slides

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 int count_remain(){
2     int i,j,cnt = 0;
3     for(i = 0; i < N; i++){
4         for(j = 0; j < M; j++){
5             if(T[i][j] == 0){
6                 cnt++;
7             }
8         }
9     }
10    return cnt;
11 }
```

	Input	Expected	Got	
✓	1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1	18	18	✓
✓	1 0 0 0 1 1 0 0 1 0 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1	15	15	✓
✓	0 0	25	25	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **6**

Correct

Mark 0.33 out of 1.00

Cho 2 mảng toàn cục B và T dùng để lưu bản đồ mìn và trạng thái đóng/mở của các ô trong trò chơi Minesweeper.

```
#define M 5
#define N 5
#define MINE -1
/*Mảng B lưu bảng đồ mìn*/
int B[M][N];
/*Mảng T lưu trạng thái các ô*/
int T[M][N];
```

Hãy viết [hàm](#) **open_cell_1(r, c)** để mở ô r, c theo quy tắc 1: chỉ quan tâm đến ô này, không quan tâm đến các ô khác.

Khuôn dạng (prototype)

- Tên [hàm](#): open_cell_1
- Tham số:
 - r: int
 - c: int
- Kiểu trả về: void

Thân [hàm](#):

- Kiểm tra trạng thái ô (r, c) và đặt trạng thái nó là mở.

Chú ý

- Dữ liệu đầu vào luôn hợp lệ.
- Chỉ viết [HÀM](#), không viết TOÀN BỘ CHƯƠNG TRÌNH

Gợi ý

- Xem slides

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 void open_cell_1(int r, int c){
2     if(T[r][c] == 0){
3         T[r][c] = 1;
4     }
5 }
```

	Input	Expected	Got	
✓	<div>0 0 0 0 1</div> <div>1 0 0 1 0</div> <div>0 0 0 0 0</div> <div>0 0 1 0 0</div> <div>1 0 0 0 1</div> <div>0 0</div>	<div>1 0 0 0 1</div> <div>1 0 0 1 0</div> <div>0 0 0 0 0</div> <div>0 0 1 0 0</div> <div>1 0 0 0 1</div>	<div>1 0 0 0 1</div> <div>1 0 0 1 0</div> <div>0 0 0 0 0</div> <div>0 0 1 0 0</div> <div>1 0 0 0 1</div>	✓
✓	<div>1 0 0 0 1</div> <div>1 0 0 1 0</div> <div>0 0 0 0 0</div> <div>0 0 1 0 0</div> <div>1 0 0 0 1</div> <div>0 4</div>	<div>1 0 0 0 1</div> <div>1 0 0 1 0</div> <div>0 0 0 0 0</div> <div>0 0 1 0 0</div> <div>1 0 0 0 1</div>	<div>1 0 0 0 1</div> <div>1 0 0 1 0</div> <div>0 0 0 0 0</div> <div>0 0 1 0 0</div> <div>1 0 0 0 1</div>	✓

	Input	Expected	Got	
✓	<div><div>00000</div><div>00000</div><div>00000</div><div>00000</div><div>00000</div><div>34</div></div>	<div><div>00000</div><div>00000</div><div>00000</div><div>00001</div><div>00000</div></div>	<div><div>00000</div><div>00000</div><div>00000</div><div>00001</div><div>00000</div></div>	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.33/1.00**.

Question **7**

Correct

Mark 0.67 out of 1.00

Cho 2 mảng toàn cục B và T dùng để lưu bản đồ mìn và trạng thái đóng/mở của các ô trong trò chơi Minesweeper.

```
#define M 5
#define N 5
#define MINE -1
/*Mảng B lưu bảng đồ mìn*/
int B[M][N];
/*Mảng T lưu trạng thái các ô*/
int T[M][N];
```

Hãy viết [hàm](#) **open_cell_2(r, c)** để mở ô r, c theo quy tắc 2:

- Mở ô (r, c).
- Nếu B[r][c] == 0 (có nghĩa là 8 ô xung quanh nó không có mìn) thì mở luôn 8 ô xung quanh nó.

Khuôn dạng (prototype)

- Tên [hàm](#): open_cell_2
- Tham số:
 - r: int
 - c: int
- Kiểu trả về: void

Thân hàm:

- Mở ô (r, c) và xét để mở 8 ô xung quanh nó.

Chú ý

- Dữ liệu đầu vào luôn hợp lệ.
- Chỉ viết [HÀM](#), không viết TOÀN BỘ CHƯƠNG TRÌNH

Gợi ý

- Xem slides

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 void open_cell_2(int r, int c) {
2     int i, j;
3     T[r][c] = 1;
4     if (B[r][c] == 0) {
5         for (i = -1; i < 2; i++) {
6             for (j = -1; j < 2; j++) {
7                 if (i != 0 || j != 0) {
8                     if (r + i >= 0 && r + i < M && c + j >= 0 && c + j < N) {
9                         T[r + i][c + j] = 1;
10                    }
11                }
12            }
13        }
14    }
15 }
```

	Input	Expected	Got	
✓	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>1 1 0 0 0</div> <div>1 1 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div></div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>1 1 0 0 0</div> <div>1 1 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div></div>	✓

	Input	Expected	Got	
✓	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>2 2</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div>	✓
✓	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>3 3</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 1 1 1</div> <div>0 0 1 1 1</div> <div>0 0 1 1 1</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 1 1 1</div> <div>0 0 1 1 1</div> <div>0 0 1 1 1</div>	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

Question **8**

Correct

Mark 1.00 out of 1.00

Cho 2 mảng toàn cục B và T dùng để lưu bản đồ mìn và trạng thái đóng/mở của các ô trong trò chơi MineSweeper.

```
#define M 5
#define N 5
#define MINE -1
/*Mảng B lưu bảng đồ mìn*/
int B[M][N];
/*Mảng T lưu trạng thái các ô*/
int T[M][N];
```

Hãy viết [hàm ĐỆ QUY](#) **open_cell_3(r, c)** để mở ô r, c theo quy tắc 3:

- Nếu ô r, c) đã mở => return
- Ngược lại:
 - Mở ô (r, c)
 - Nếu B[r][c] == 0 (có nghĩa là 8 ô xung quanh nó không có mìn) gọi đệ quy mở 8 ô xung quanh nó theo quy tắc 3.

Khuôn dạng (prototype)

- Tên [hàm](#): open_cell_3
- Tham số:
 - r: int
 - c: int
- Kiểu trả về: void

Thân [hàm](#):

- Mở ô (r, c) và gọi đệ quy để mở 8 ô xung quanh nó.

Chú ý

- Dữ liệu đầu vào luôn hợp lệ.
- Chỉ viết [HÀM](#), không viết TOÀN BỘ CHƯƠNG TRÌNH

Gợi ý

- Xem slides

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 void open_cell_3(int r,int c){
2     int i, j;
3     if(T[r][c]==1);
4     else{
5         T[r][c] = 1;
6         if (B[r][c] == 0) {
7             for ( i = -1; i < 2; i++) {
8                 for ( j = -1; j < 2; j++) {
9                     if (i != 0 || j != 0) {
10                        if (r + i >= 0 && r + i < M && c + j >= 0 && c + j < N) {
11                            if(T[r + i][c + j] == 0){
12                                open_cell_3(r+i,c+j);
13                            }
14                        }
15                    }
16                }
17            }
18        }
19    }
20 }
```

	Input	Expected	Got	
--	-------	----------	-----	--

	Input	Expected	Got	
✓	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>1 1 0 0 0</div> <div>1 1 0 0 0</div> <div>1 1 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>1 1 0 0 0</div> <div>1 1 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div>	✓
✓	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>2 2</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div>	✓
✓	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>0 0 0 0 0</div> <div>3 3</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 1 1</div> <div>0 0 0 1 1</div> <div>0 0 1 1 1</div> <div>0 0 1 1 1</div> <div>0 0 1 1 1</div>	<div>0 1 1 1 0</div> <div>0 1 -1 1 0</div> <div>1 2 2 1 0</div> <div>1 -1 1 0 0</div> <div>1 1 1 0 0</div> <div>0 0 0 1 1</div> <div>0 0 0 1 1</div> <div>0 0 1 1 1</div> <div>0 0 1 1 1</div> <div>0 0 1 1 1</div>	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Thu](#)

Jump to...

[Thao tác với tập tin văn bản](#)