

Announcements

Next week class

Monday: as usual

Wednesday: quiz and discussion for Wed and Fri

Friday: all lab (My last scheduled day out of town)

Office hours

today (Friday): 2:30 - 3:45 (ends 15 minutes early)

Sunday: 2:00 - 4:00 pm

Monday, Wednesday (normal): 1:30-3:00

Tuesday: 3:15-5:00

Thursday, Friday: canceled

Reminder: Time change early Sunday morning

Set clocks back 1 hour ("Fall back; Spring ahead")

Note: MyroC defines structs for both Pixel and Picture

Since these are pre-defined in MyroC.h,

do NOT add definitions in your own programs,

just use `#include <MyroC.h>`

Supplemental Problem 3

Given tel_locs array, tel_locs[0][0] given as SW corner location "safe" if it is within 2 blocks of telephone moving north and west

Suppose int tel_locs [5][5] = {{0,0,0,0,0},
{0,0,0,0,0},
{0,0,1,0,0},
{0,0,0,0,0},
{0,0,0,0,0}};

Which locations are safe?

A. 00X00
0XX00
XXX00
00000
00000

B. 00X00
00XX0
00XXX
00000
00000

C. 00000
00000
XXX00
0XX00
00X00

D. 00000
00000
00XXX
00XX0
00X00

E. 00X00
0XXX0
XXXXX
0XXX0
00X00

Supplemental Problem 3

Given tel_locs array, tel_locs[0][0] given as SW corner location "safe" if it is within 2 blocks of telephone moving north and west

Suppose int tel_locs [5][5] = {{0,0,0,0,0},
 {0,0,0,0,1},
 {0,0,0,0,0},
 {0,0,0,0,0},
 {0,0,0,0,0}};

Which locations are safe?

A. 0000X
 0000X
 00000
 00000
 00000

B. 00000
 0000X
 0000X
 0000X
 00000

C. 0000X
 0000X
 X0000
 00000
 00000

D. X00XX
 X0000
 X0000
 X0000
 00000

E. some-
 thing
 else

Supplemental Problem 3

Given `tel_locs` array, `tel_locs[0][0]` given as SW corner
location "safe" if it is within 2 blocks of telephone

Consider the following code segment to print this grid:

```
for (int row = 0; row < north_south_size; row ++)  
{  
    for (int col = 0; col < east_west_size; col++)  
        printf ("%2d", tel_locs[row][col]);  
    printf ("\n");  
}
```

Consider the North/South and East/West orientation of the grid printed:

- | | | | | | | | |
|---|-------|----|-------|------|------|------|------|
| 1 | North | 2. | South | 3. | | 4. | |
| | | | | West | East | East | West |
| | South | | North | | | | |

Which relative directions describe the printed grid?

- A. 1 & 3 B. 2 & 3 C. 1 & 4 D. 2 & 4

Supplemental Problem 3

Given `tel_locs` array, `tel_locs[0][0]` given as SW corner
location "safe" if it is within 2 blocks of telephone

Consider the following code segment to print this grid:

```
for (int row = north_south_size - 1; row >= 0; row --)
{
    for (int col = east_west_size - 1; col >= 0; col --)
        printf ("%2d", tel_locs[row][col]);
    printf ("\n");
}
```

Consider the North/South and East/West orientation of the grid printed:

- | | | | | | | | |
|---|-------|----|-------|------|------|------|------|
| 1 | North | 2. | South | 3. | | 4. | |
| | | | | West | East | East | West |
| | South | | North | | | | |

Which relative directions describe the printed grid?

- A. 1 & 3 B. 2 & 3 C. 1 & 4 D. 2 & 4

Projects, Supplemental Problems: the following are required

For each function, need Comments, Pre/post conditions

Format reflects structure

Testing

both black box and white box testing

rationale behind given test cases

statement whether results are correct and how you know

summary statement about program correctness

Program under development without comments, formatting

==> no help possible (don't ask)

Submission without one of these ==> no grade

Definitions in MyroC

```
typedef struct
{
    unsigned char R;
    unsigned char G;
    unsigned char B;
} Pixel
```

```
typedef struct
{
    int height; //<= 266
    int width;  //<= 427
    Pixel pix_array[266][427];
} Picture;
```

Consider statements about
height and width:

1. setting height=266,
width=427 is always safe
2. any values for height, width
will work

- A. 1 and 2
- B. 1, not 2
- C. 2, not 1
- D. neither 1
nor 2

A white Pixel may be declared as

```
Pixel whitePix = {255, 255, 255}; // white pixel
```

Which, if any, of the following statements define a picture of the prescribed color?

A. `Picture picX = {200, 300, {{whitePix}}};` //white picture

B. `Picture pic2 = {200, 300, {{{255,255,255}}}};` //white picture

1. A and B 2. A, not B 3. B, not A 4. neither
A nor B

A black Pixel may be declared as

```
Pixel blackPix = {0, 0, 0}; // black pixel
```

Which, if any, of the following statements define a picture of the prescribed color?

- A. `Picture pic3 = {200, 300, {{{0}}}};` `//black picture`
B. `Picture pic4 = {200, 300, {{blackPix}}};` `//black picture`

1. A and B 2. A, not B 3. B, not A 4. neither
A nor B

Recall MyroC defines a Pixel:

```
typedef struct
{
    unsigned char R;
    unsigned char G;
    unsigned char B;
} Pixel
```

Suppose variable `pix` is declared as a `Pixel`, and suppose one wants to increase its red value by 10, except that it may not exceed 255 as the maximum red value. Which of the following approaches work reliably:

approach 1

```
pix.R += 10;
if (pix.R > 255)
    pix.R = 255;
```

approach 2

```
int value = pix.R+10;
if (value > 255)
    value = 255;
pix.R = value;
```

A. both 1
and 2

B. 1,
but not 2

C. 2,
but not 1

D. neither 1
nor 2

According to the lab (with a citation), converting a pixel to gray scale involves setting RGB values of a pixel to 30% of the red value + 59% of the green value + 11% of the blue value.

Which of the following code segments accomplish this?

approach X

```
pix.R = 0.3*pix.R + 0.59*pix.G + 0.11*pix.B;  
pix.G = 0.3*pix.R + 0.59*pix.G + 0.11*pix.B;  
pix.B = 0.3*pix.R + 0.59*pix.G + 0.11*pix.B;
```

approach 2

```
int value = 0.3*pix.R + 0.59*pix.G + 0.11*pix.B;  
pix.R = value;  
pix.G = value;  
pix.B = value;
```

A. both 1
and 2

B. 1,
but not 2

C. 2,
but not 1

D. neither
X1nor 2