home > topics > c / c++ > questions > c help magic square

C Help Magic Square

winnerpl

P: n/a

Hey guys I'm trying to get a magic square but I'm stuck with it printing out only 1's and 0's in random places. Hope you experts can can provide some input on what I'm doing wrong.

```
#include <stdio.h>
#define MAX 20

int x[MAX][MAX];

int size_request(void); //asks the user to enter the size x size square.
void fill_square(int[][], int); //fills up the magic square with the
appropriate numbers.
void print_square(int[], int); //prints out the magic square.

int main(void)
{

int roww;
int size = size_request();
x[size][size];

fill_square(x, size);

for(roww = 0; roww < size; roww++)
print_square(x[roww], size);

return 0;
}

int size_request(void)
{
int size;
do
```

Question stats

- viewed: 3565
- replies: 4
- date asked: Nov 14 '05

Follow this discussion

```
{
printf("Enter a positive, odd integer in the range from 1 to 15:
");
scanf("%d", &size);
} while((size % 2 != 1) ||(size > 15) || (size < 1));

printf("\n");

return size;
}

void fill_square(int x[][MAX], int size)
{
int count;
int row = size - 1;
int col = row / 2;
int maxnum = size * size;

for(count = 1; count < maxnum; count++)
{
x [row][col] = 1;

if(x [row-2][col+1] == 0)
{
row = row - 2;
col = col + 1;
if (col == size)
col = col - size;
else if (row < 0)
row = row + size;
}
else
{
row --;
if (row < 0)
row = row + size;
}
}
return;
}

void print_square(int x[], int size)
{
int colw;

for(colw = 0; colw < size; colw++)
printf("%4d", x[colw] * x[colw]);

printf("\n");
return;
}
```

Thanks in advance guys.

Nov 14 '05 #1

Post Reply

Share this Question

4 Replies

Jack Klein

P: n/a

On Sun, 31 Oct 2004 21:07:29 -0500, "winnerpl" <wi******@hotmail.com> wrote in comp.lang.c:

> Hey guys I'm trying to get a magic square but I'm stuck with it printing out only 1's and 0's in random places. Hope you experts can can provide some input on what I'm doing wrong.
>
> OK, first of all you are writing a program in some language that sort of resembles C, but is not actually C.
>
> ```
> #include <stdio.h>
> #define MAX 20
>
> int x[MAX][MAX];
>
> int size_request(void); //asks the user to enter the size x size square.
> void fill_square(int[][], int); //fills up the magic square with the
> appropriate numbers.
> ```
>
> Note, don't ever use // comments in C or C++ code posted to usenet. Line wrapping by your news software has moved "appropriate numbers." to the next line, where it is nothing but a syntax error when somebody tries to compiler your code. Use /* */ comments, and line breaking can't mess them up.
>
> In any case, the prototype above is not valid C, it is a syntax error. There is no such thing as a function array parameter with more than one empty '[ ]' pair. If your compiler accepts this, it is either not a C compiler or you are not invoking it to operate in standard C conforming mode.
>
> ```
> void print_square(int[], int); //prints out the magic square.
>
> int main(void)
> {
> ```

```
int roww;
int size = size_request();
x[size][size];
```

What do you think that this line does? You have already defined the array at file scope. It doesn't change the size of the array. All it does is evaluate the value of the int in the array at x[size][size], which has the value 0, and then throws away that 0 value.

```
fill_square(x, size);

for(roww = 0; roww < size; roww++)
print_square(x[roww], size);

return 0;
}

int size_request(void)
{
int size;
do
{
printf("Enter a positive, odd integer in the range from 1 to 15:
");
scanf("%d", &size);
} while((size % 2 != 1) ||(size > 15) || (size < 1));

printf("\n");

return size;
}

void fill_square(int x[][MAX], int size)
{
int count;
int row = size - 1;
int col = row / 2;
int maxnum = size * size;
```

Let's see, if 'size' is 1, a value accepted by your size_request() function, then 'row', 'col' are both 0 and 'maxnum' is 1.

```
for(count = 1; count < maxnum; count++)
```

In that case, this loop has a real problem, it will never execute at all because 'count' starts at 1 and is never less than 'maxnum' if 'size' is 0.

```
{
x [row][col] = 1;
```

The line above is the only line that ever writes a value into the array. The only value it ever writes is 1, so it is hard to see how the array can have any values other than 0 (what it was initialized with) or 1 in it. Perhaps you meant to write:

```
x [row][col] = count;
if(x [row-2][col+1] == 0)
```

Using 'row-2' as a subscript will produce undefined behavior if 'size' is 1, although since this code won't execute when 'size' is 1, I don't suppose that matters much.

```
{
```

> ```
> row = row - 2;
> Why not 'row -= 2'?
> col = col + 1;
> Why not 'col += 1' or just '++col'?
> if (col == size)
> col = col - size;
> The line above could be replaced with 'col = 0'.
> else if (row < 0)
> row = row + size;
> }
> else
> {
> row --;
> if (row < 0)
> row = row + size;
> }
> }
> return;
> }
>
> void print_square(int x[], int size)
> {
> int colw;
>
> for(colw = 0; colw < size; colw++)
> printf("%4d", x[colw] * x[colw]);
>
> printf("\n");
> return;
> }
> ```
>
> Thanks in advance guys.

Don't they teach people how to debug anymore? Have you single stepped through your 'fill_square()' function to see what it does?

I made the following changes to your code:

1. Change prototype to 'void fill_square(int[MAX][MAX], int);' and changed the function definition to match.

2. Eliminated the do-nothing 'x[size][size];' statement from main().

3. Changed 'x [row][col] = 1;' in fill_square() to 'x [row][col] = count;'.

4. Added line right after that:

```
printf("x [%d][%d] = %d\n", row, col, count);
```

Now when I run your program and enter '3' for size, I get this output:

```
Enter a positive, odd integer in the range from 1 to 15: 3
```

```
x [2][1] = 1
x [0][2] = 2
x [-2][0] = 3
x [-1][1] = 4
x [0][2] = 5
x [-2][0] = 6
x [-1][1] = 7
x [0][2] = 8
0 0 64
0 0 0
0 1 0
```

....so it appears that you are using the expression [row-2] when row is 0, and you are accessing memory outside the array. You need to work on your subscript expressions.

--
Jack Klein
Home: http://JK-Technology.Com
FAQs for
comp.lang.c http://www.eskimo.com/~scs/C-faq/top.html
comp.lang.c++ http://www.parashift.com/c++-faq-lite/
alt.comp.lang.learn.c-c++
http://www.contrib.andrew.cmu.edu/~a...FAQ-acllc.html

Nov 14 '05 #2

Peter Nilsson

P: n/a

"Jack Klein" <ja*******@spamcop.net> wrote:

> > void fill_square(int[][], int);
> > ... the prototype above is not valid C,
>
> Agreed.
> it is a syntax error.
> I guess you could argue your case here, given that the (outer) array element's type can be deduced as incomplete from the syntax alone. [But a part of me thinks you're stretching it with this statement. :-]
> There is no such thing as a function array parameter with more than one empty '[ ]' pair. If your compiler accepts this, it is either not a C compiler or you are not invoking it to operate in standard C conforming mode.

```
% dir /B decl.*
decl.c

% type decl.c
void blah(int [][]);

% gcc -c -ansi -pedantic decl.c
```

```
decl.c:1: warning: array type has incomplete element type

% dir /B decl.*
decl.c
decl.o

%
```

A conforming implementation _is_ allowed to accept the code, so long as it emits the required diagnostic.

--
Peter

Nov 14 '05 #3

Tim Rentsch

P: n/a

Jack Klein <ja*******@spamcop.net> writes:

> On Sun, 31 Oct 2004 21:07:29 -0500, "winnerpl" <wi******@hotmail.com> wrote in comp.lang.c:
>
>> void fill_square(int[][], int);
>
>> [some snippage]
>
> In any case, the prototype above is not valid C, it is a syntax error. There is no such thing as a function array parameter with more than one empty '[ ]' pair. If your compiler accepts this, it is either not a C compiler or you are not invoking it to operate in standard C conforming mode.

Both a prototype and a function definition with an 'int[][]' type were accepted by 'gcc -(ansi|std=c99) -pedantic' -- some warnings, but no errors, and certainly no syntax errors. Is gcc in error here? Or does the standard actually allow this?

Incidentally, the meaning assigned seems to be the same as though it were like this:

```
int f( int (*)[] ); /* or int f( int [][] ); */

int f( int (*x)[] ){ /* or int f( int x[][] ){ */
return (*x)[0];
}

extern int a[];

int main(){
return f( &a ) == 0;
```

}

Of course, I'm making no claim that code like this is sensible.
Only asking if it's legal.

Nov 14 '05 #4

Flash Gordon

P: n/a

On 01 Nov 2004 20:41:54 -0800
Tim Rentsch <tx*@alumnus.caltech.edu> wrote:

> Jack Klein <ja*******@spamcop.net> writes:
>
>> On Sun, 31 Oct 2004 21:07:29 -0500, "winnerpl"
>> <wi******@hotmail.com> wrote in comp.lang.c:
>>
>>> void fill_square(int[][], int);
>>
>>> [some snippage]
>>
>> In any case, the prototype above is not valid C, it is a syntax error. There is no such thing as a function array parameter with more than one empty '[ ]' pair. If your compiler accepts this, it is either not a C compiler or you are not invoking it to operate in standard C conforming mode.
>
> Both a prototype and a function definition with an 'int[][]' type were accepted by 'gcc -(ansi|std=c99) -pedantic' -- some warnings, but no errors, and certainly no syntax errors. Is gcc in error here?
> Or does the standard actually allow this?

<snip>

The C standard does not discriminate between warnings and diagnostics, and once a compiler has produced the required diagnostic it is allowed to produce a program. So yes, gcc is allowed to do this since it has produced a diagnostic.
--
Flash Gordon
Sometimes I think shooting would be far too good for some people.
Although my email address says spam, it is real and I read it.

Nov 14 '05 #5

This discussion thread is closed

Replies have been disabled for this discussion.

Start new discussion

Similar topics

magic square matrix
magic square
grid output pattern for magic square in c
magic square - topographical model ...
magic square topograhical model
How can i open the magic square in the text file????
Magic Square program run amuck!
magic square~ what should i do
program in c to generate a magic square of size n*n, where n is odd

Browse more C / C++ Questions on Bytes

About Bytes | Advertise on Bytes | Contact Us
Sitemap | C / C++ Answers Sitemap | C / C++ Insights Sitemap

Follow us to get the Latest Bytes Updates