Project 8

Generated by Doxygen 1.7.6.1

Mon Sep 15 2014 01:57:07

Contents

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 pumpkin.cpp File Reference

```
#include <stdexcept> #include <iostream>
```

Functions

- int findP (char garden[ ][MAX_SIZE], int row, int col, int rows, int cols)

    Function Prototypes.
- int main ()

Variables

- const int MAX_SIZE = 40

2.1.1 Detailed Description

Author

CatherinePollock

Date

9/16/14

This is the file that drives reading in, examining, and outputting information about a pumpkin patch.

2.1.2 Function Documentation

2.1.2.1 int **findP** (char *garden[ ][MAX_SIZE],* int *row,* int *col,* int *rows,* int *cols*)

Function Prototypes.

Function Implementation.

findP function

Determines if 'p' is at given location. If it is, the elements around are checked for a 'p' value. If no 'p' is found, the function returns 1, for one found instance. If one 'p' is found, the function recursively calls itself with the new location of a 'p'. If more than one 'p' is found around the original 'p', the function calls itself with all values surrounding it. With each call of the function and each finding of a 'p', 1 is added to the total found 'p's, and the value is changed to an uppercase P to flag that it has been counted.

Parameters

char	garden array, used to search through
int	row is given to indicate current row to be checked
int	col is given to indicate current column to be checked
int	rows to indicate max amount of rows
int	cols to indicate max amount of columns

Returns

　　int size of current pumpkin patch

declare variables and arrays

check if value at location is 'p'

change the value of 'p' to 'P' to flag that it has been counted.

check above for another 'p'

check below for another 'p'

check to left for another 'p'

check to right for another 'p'

if one 'p' was found, call this function with new location

if multiple 'p's were found, call function with all surrounding locations

2.1.2.2 int **main** ( )

Main Function

This program counts patches of pumpkins in a garden. First, the program gets a size for rows and columns from the user. Next, the user must enter in these values for their garden. Once they have all been entered, the program uses the function findP to search for patches. The size is given back to main after each patch is found, and then the sizes

of the patches for each garden is sorted using a bubble sort. Once they have been sorted, the results are printed to screen. variable decleration

determine size of garden

array decleration

reset variables

loop through rows to fill array

loop through columns

enter the char into array unless it is a new-line or space

reset counter and index values

if array has rows and columns loop through

find all instances of p and save to variable size

loop through each garden

print garden information

sort sizes for each gardem with bubble sort

2.1.3   Variable Documentation

2.1.3.1   const int **MAX_SIZE = 40**