The Pumpkin Patch

AUTHOR

Version 1.00

09/17/2014

Table of Contents

The Great Pumpkin Patch Problem

It's almost Halloween and Linus is setting out to the garden to wait for the Great Pumpkin. Unfortunately, due to diversification, there are lots of other gourds in the garden this year. This program will determine how many patches of pumpkins there are and how big they are.

Author:

Saharath Kleips

Version:

1.00

Consider a 10 x 10 garden with zucchini (z), yellow squash (y), spaghetti squash (s), and pumpkins (p):

pzzzzzzzzp

pyypzzzzzy

ppppssssyy

ssspssssyy

sssspssyy

ssssppsspy

zzzzzzsspy

zzzzzzsspy

yyyypzsspy

yyyypppppy

This garden has four patches of pumpkins: one at the top left corner covering 8 squares, one in the top right corner covering 1 square, one in the center covering 4 squares, and one near the buttom right covering 10 squares. Note: in order for a square to be a part of a patch, it must connect with another square in that patch along an edge, not just a corner.

Description has been modified from Professor Frederick C. Harris' PA03-PC1 assignment (University of Nevada, Reno - CS302).

File Index

File List

Here is a list of all documented files with brief descriptions:

File Documentation

pumpkin.cpp File Reference

```
#include <stdlib.h>
#include <iostream>
```

Functions

int calculateSize (char **garden, int i, int j)

Calculates the size of a patch recursively checking if the right, bottom, left, and then up patches are also pumpkins.

void quickSort (int elements[], int leftBound, int rightBound)

Preforms a quick sort from least to greatest on an array of integers using the middle as a pivot point.

int main ()

Main function that controls user input, console output, and program loops.

Variables

int row = 0

The max row size of the garden.

int column = 0

The max column size of the garden.

Detailed Description

Definition in file pumpkin.cpp.

Function Documentation

int calculateSize (char ** *garden*, int *i*, int *j*)

Calculates the size of a patch recursively checking if the right, bottom, left, and then up patches are also pumpkins.

Precondition:

garden must not be empty.

Parameters:

garden	is a two dimensional array of characters representing gourds.
i	is the row index of the pumpkin.
j	is the column index of the pumpkin.

The size of the patch.

Definition at line 143 of file pumpkin.cpp.

void quickSort (int *elements*[], int *leftBound*, int *rightBound*)

Preforms a quick sort from least to greatest on an array of integers using the middle as a pivot point.

Precondition:

elements must not be empty.

Parameters:

Returns:

elements	is the list of integers to be sorted.
leftBound	is the left most index to be sorted.
rightBound	is the right most index to be sorted.

elements is a sorted from least to greatest.

Definition at line 175 of file pumpkin.cpp.

Variable Documentation

int column = 0

The max column size of the garden.

Definition at line 45 of file pumpkin.cpp.

int row = 0

The max row size of the garden.

Definition at line 43 of file pumpkin.cpp.

Index

INDE