Nicholas Kidd

Program 2 Report

My struct to define each image's location information was called Image. My structure, which was a list of locations to reference my Image structs was called loc.

For the Image struct, I store four different things. I store the location name in a character array called locationName that is a character array of 100 characters taken from the argument line. I store the x coordinate that is taken from the argument line in an int named x. I store the y coordinate that is taken from the argument line in an int named y. Then I take the image number, denoted imageNumber, as an int that is taken from my variable index in my main function.

For my locationName, I chose a character array of 100 characters because I take the user's input and the arguments off the argument line as character arrays as well because this is a basic form of storing strings. The limitation is that if the location name is over 100 characters, then my locationName variable would not be able to store it and end in a segmentation fault. For both coordinates x and y, I used an int to store them because the coordinate values should not be outside of the range -2,147,483,648 <= x,y <= 2,147,483,648. In the data sets we were given, none of the integers go above 1000 or below -1000. This could be changed to a long long into if a larger coordinate system is needed. For now if the range is outside the range I gave above, it will be unable to store it. The imageNumber was an int, because we would not be tested with more than 2,147,483,648 data images. If there were more data images, I could have used an unsigned int or a long long unsigned int because this number will never be negative.

For the loc struct i store three different things. I store the the location name (the same as the first struct) in a char array of 100 characters called city. I store an integer array of indexes that refer to points on my Image list. This list is called indices and is made up of 1000 values. I set it to 1000 so if we had an entry of 1000 unique location names, it could store all 1000. The last value is an integer called idx. Idx holds the next available element to place an index in the indices integer array.

The limitations for city are the same as locationName for the Image struct. The integer array has a limitation of more than 1000 elements. I made the size of the array 1000 because we were taking in at most 1000 images and there is a good chance that there would not be more than

1000 of the same image. If there are, it is an easy fix as you could raise the size, it would just use more data. The idx has the same limitations of the imageNumber in struct. Once again, I could have used an unsigned int or a long long unsigned int but there was no need for the amount of data that would be present.

Other types of limitations for my program include the LocationList, which is an array containing all of my loc structs. I originally set the array to a size of argc but this was creating a segmentation fault resulting in the program outputting “core dumped”. The DataList I have, which is a list of all of my image structs has a size of argc but my program would not run at all if both were set to a size of argc. Another limitation present in the format of our output is for the image number to be 3 digits at all times. If the amount of images is over 1000, there is no way to output the image number with 3 digits without an overflow occurring.