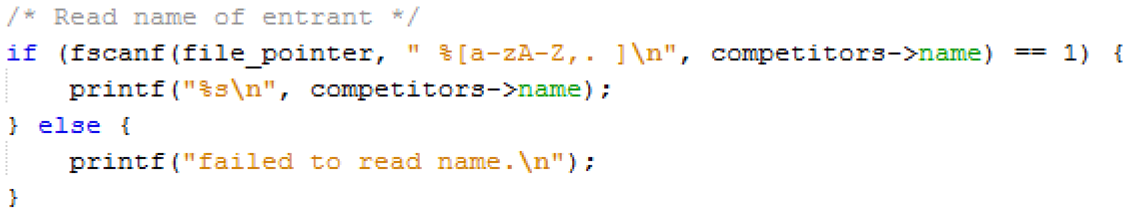
C Write up

Part1

When it comes to reading in the competitors from the file, I malloc the space for a competitor, and read in all the information until I reach the end of the file. When scanning in, to avoid a yellow underlined warning, I use a check that it has been scanned correctly shown below.



```
/* Read name of entrant */
if (fscanf(file_pointer, " %[a-zA-Z,. ]\n", competitors->name) == 1) {
    printf("%s\n", competitors->name);
} else {
    printf("failed to read name.\n");
}
```

This check avoids the following warning:
warning: ignoring return value of 'scanf', declared with attribute warn_unused_result [-Wunused-result]

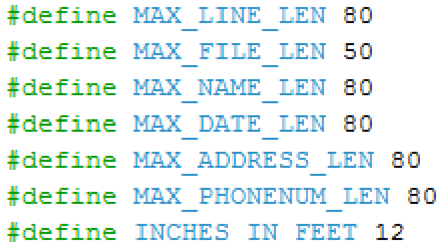
I tried very hard throughout my code, and put a lot of time and energy into, the removal of all errors and warnings brought up on the build host "build and complete". With my code having zero warnings, I am very happy with the quality of my checks in my code.

The Competitors are given a competitor number simply by incrementing an int every time a new struct is created. This means that as each competitor is read in they are automatically assigned a number 1 larger than the previous competitor.

The structs that I create from the file are instantly placed into the binary search tree once it has been filled with all information. I pass into the insert function the root node, and the current competitor that was just created. Inside of insert, I check if there is a root to the entire binary tree. If there is not, the first competitor automatically gets set as the root. If a root already exists, then a check is done to see whether the new node should be placed in the left or the right sub-tree. If the sizes are equal, it gets placed in the right sub tree as per the spec. I call a function called get_total_size on the new competitor and the root it is being compared to. This function gets the total length of their fruit and vegetables in inches, which is stored as a float as there may be a decimal number of inches. Once the comparison has been made, the node is inserted into the correct position. This of reading in competitors and placing them into the tree continues until all competitors have been read in.

Now the Layout for the table which the BST gets printed to it defined. I allow 20 characters for the name and competitor number, as the name can be somewhat long, and the heading of the column for competitor number itself is large. For the fruits lengths I allow 15 characters, which leaves nice spacing between them. Then the function print_results is called. This function is a recursive function, which prints out the nodes in the binary tree in-order. It works by going as far into the left tree as possible, until it reaches a node whose left node is null. If the left node is null, then this is the smallest node available, and so it is printed. It is printed in a function called simply “print”. This function first of all gets as Strings, or “char *”s the values for the length of the fruits and vegetables. These strings are fetched in the function “inches_to_feet_and_inches”. Here I malloc a char array of size 15, as that is the maximum length I have delegated for the fruit lengths. Doing some basic math, I convert the total length in inches into feet and inches, and with these values I use “sprint” to concatenate them into the char array. This then gets returned back to “print”. With these values, I not simply print using the same format for the table as described above, all the information wanted from the spec. After these have finished printing, you return back to print_results to continue searching the BST in order for nodes to print. If the node was printed, then that was because there was no left node, so now we look for a right node. If there is a right node, we call print_results with it, and this process continues. Once the right tree of a node is null then the recursively stacked function can start “unwinding” and will eventually unravel until you reach the root node, and continue down the right side of the BST. Once this has finished, the program has completed.

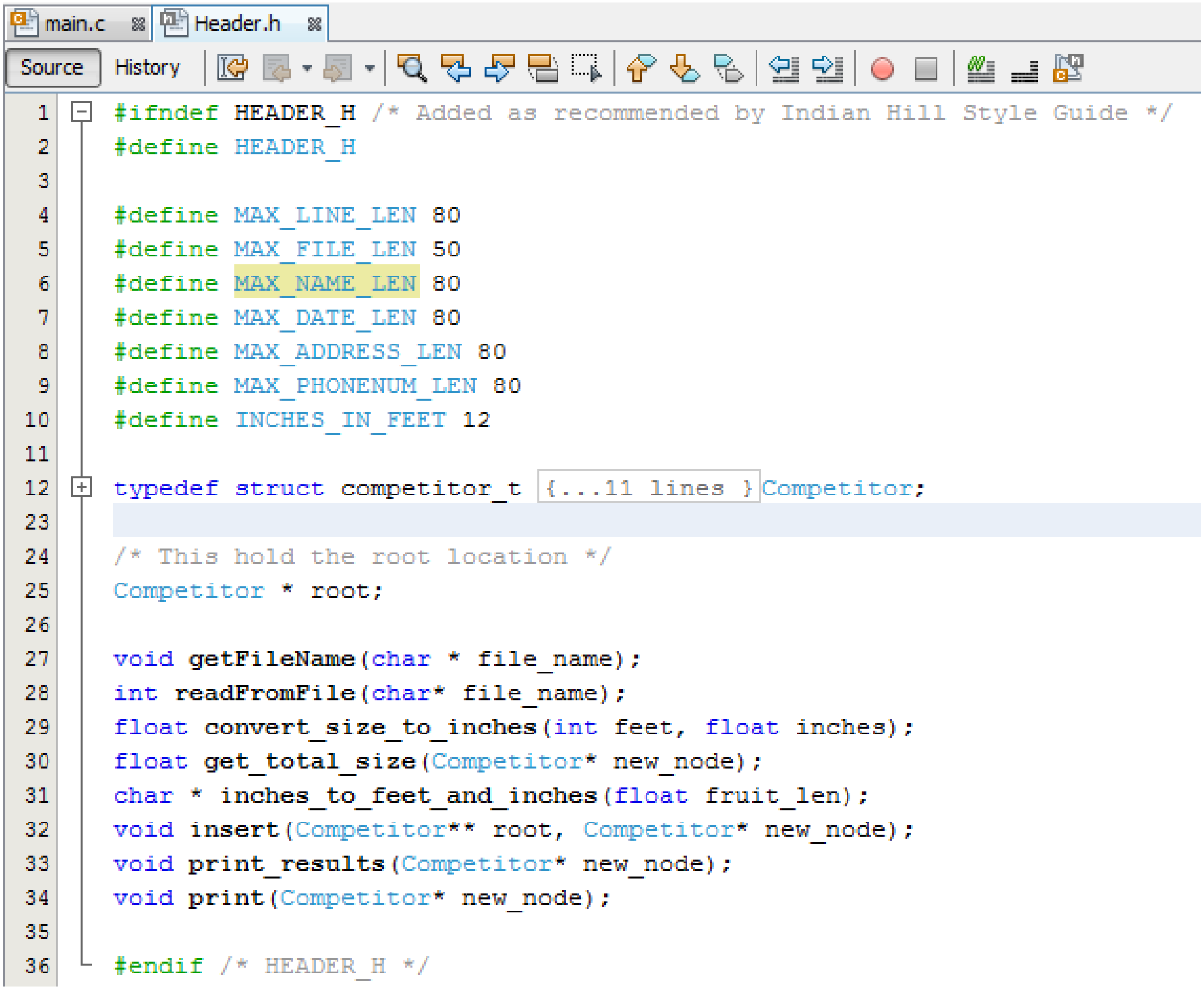
In my program I tried to use constants in all places it was appropriate. In total I had 7 constants, shown below



```
#define MAX_LINE_LEN 80
#define MAX_FILE_LEN 50
#define MAX_NAME_LEN 80
#define MAX_DATE_LEN 80
#define MAX_ADDRESS_LEN 80
#define MAX_PHONENUM_LEN 80
#define INCHES IN FEET 12
```

These mainly consist of struct variables being capped at 79, and thus I allow for 80 characters, because I need to take into account the null terminator C places at the end of strings. It is capped at 79 as the spec tells us that no line will be longer than 79 characters. It also has the number of inches in a foot, so that the code which involved moving between the units was self-documenting.

In my header file, I also added an unnecessary in this context, yet recommended check to see if you header file had already been defined, and if it had not to load it in. These are the top and bottom lines of the screenshot below. Although in such a simple program these checks may be considered overboard, it is certainly not a bad idea to include them, and I wished to show that I had made a strong attempt at using the Indian hill style guide when working on my project. It can also only instil good practice into my coding style, which is never a bad thing.



```c
#ifndef HEADER_H /* Added as recommended by Indian Hill Style Guide */
#define HEADER_H

#define MAX_LINE_LEN 80
#define MAX_FILE_LEN 50
#define MAX_NAME_LEN 80
#define MAX_DATE_LEN 80
#define MAX_ADDRESS_LEN 80
#define MAX_PHONENUM_LEN 80
#define INCHES_IN_FEET 12

typedef struct competitor_t {...11 lines }Competitor;

/* This hold the root location */
Competitor * root;

void getFileName(char * file_name);
int readFromFile(char* file_name);
float convert_size_to_inches(int feet, float inches);
float get_total_size(Competitor* new_node);
char * inches_to_feet_and_inches(float fruit_len);
void insert(Competitor** root, Competitor* new_node);
void print_results(Competitor* new_node);
void print(Competitor* new_node);

#endif /* HEADER_H */
```

As you can see, if the header is not defined, then we define the header. I end the “if” statement at the end of the header file.

The first screenshot [1.1] displays the system I was working on to code this project. I was using Windows7 64 bit. The version of NetBeans I used was NetBeans8.1.

As you can see from screenshot [1.1] I have zero errors or warnings when I build and clean, this was a core focus of mine throughout the project.

Part 2

I started off by adapting one of my current function to now take in an extra parameter, which is a pointer to a function. Adding this was almost trivial, after using the understanding gained from working on part 1. The function pointer points to a function that gets passed in, and declares the parameters that get passed in. Since both functions that I pass in take the same parameter, it was a very simple process.

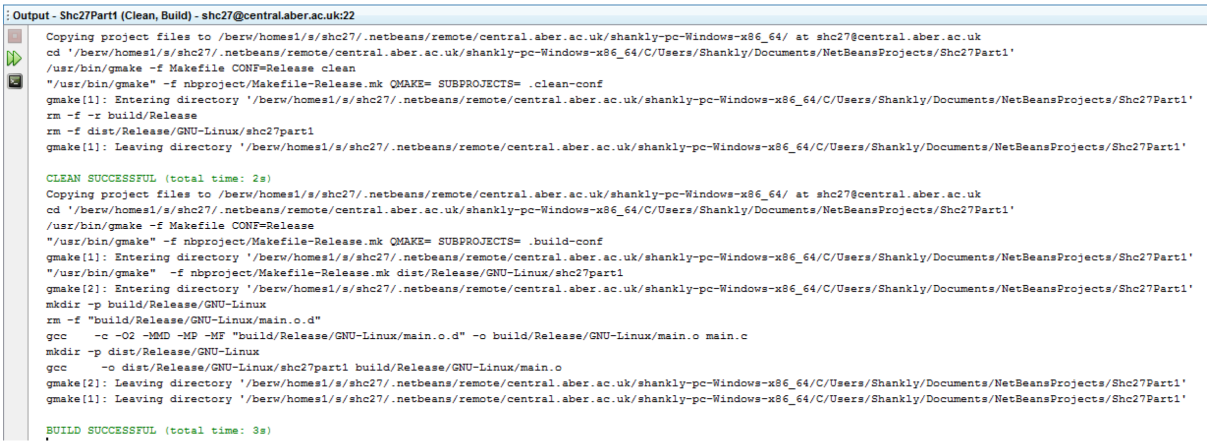
Within this function, which I called "search_through_BST", I recursively call itself as it searches through the BST in-order. When I pass in the print function from part 1, the program acts in the exact same manner. But to complete part 2, I had to create a 2nd print function for the Competitors contact details. This again was a trivial function, much simpler than the print function for part 1. I then pass into "search_through_BST" this new function, and the contact details for all competitors is printed out, in-order again. So the contact information is printed in the same order as the results table, making it easy follow and find individual comeptitors.

Disappointingly, I couldn't work out how to pass the event_name and event_date around my program nicely. This has unfortunately meant that to get the program functioning 100% correctly I made the variables event_name and event_date global variables. If I had been able to figure out an elegant way to pass them around I would have, and I acknowledge my use of global variables for these is unnecessary.

Screenshots

[1.1]

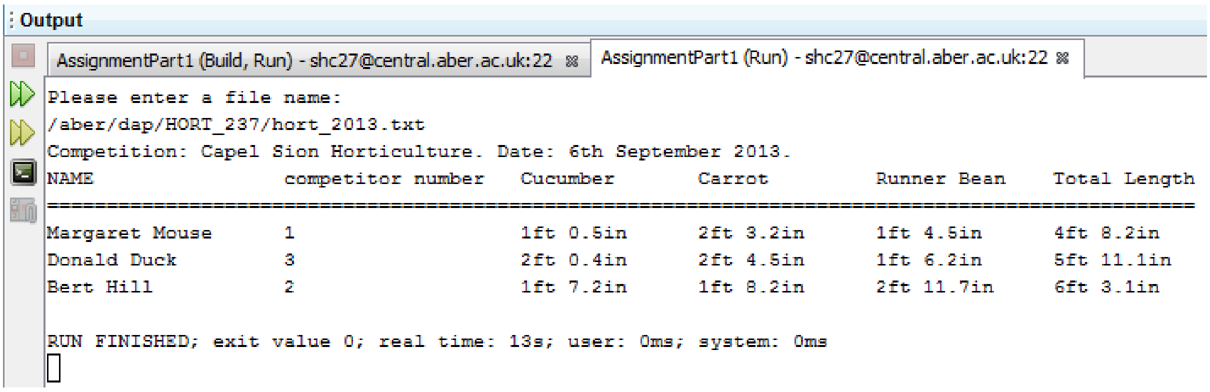
Clean and Build for part 1, using the build host


Output - Shc27Part1 (Clean, Build) - shc27@central.aber.ac.uk:22

```
Copying project files to /berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/ at shc27@central.aber.ac.uk
cd '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/Shc27Part1'
/usr/bin/gmake -f Makefile CONF=Release clean
"/usr/bin/gmake" -f nbproject/Makefile-Release.mk QMAKE= SUBPROJECTS= .clean-conf
gmake[1]: Entering directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/Shc27Part1'
rm -f -r build/Release
rm -f dist/Release/GNU-Linux/shc27part1
gmake[1]: Leaving directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/Shc27Part1'

CLEAN SUCCESSFUL (total time: 2s)
Copying project files to /berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/ at shc27@central.aber.ac.uk
cd '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/Shc27Part1'
/usr/bin/gmake -f Makefile CONF=Release
"/usr/bin/gmake" -f nbproject/Makefile-Release.mk QMAKE= SUBPROJECTS= .build-conf
gmake[1]: Entering directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/Shc27Part1'
"/usr/bin/gmake"  -f nbproject/Makefile-Release.mk dist/Release/GNU-Linux/shc27part1
gmake[2]: Entering directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/Shc27Part1'
mkdir -p build/Release/GNU-Linux
rm -f "build/Release/GNU-Linux/main.o.d"
gcc    -c -O2 -MMD -MP -MF "build/Release/GNU-Linux/main.o.d" -o build/Release/GNU-Linux/main.o main.c
mkdir -p dist/Release/GNU-Linux
gcc     -o dist/Release/GNU-Linux/shc27part1 build/Release/GNU-Linux/main.o
gmake[2]: Leaving directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/Shc27Part1'
gmake[1]: Leaving directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/Shc27Part1'

BUILD SUCCESSFUL (total time: 3s)
```

[1.2]

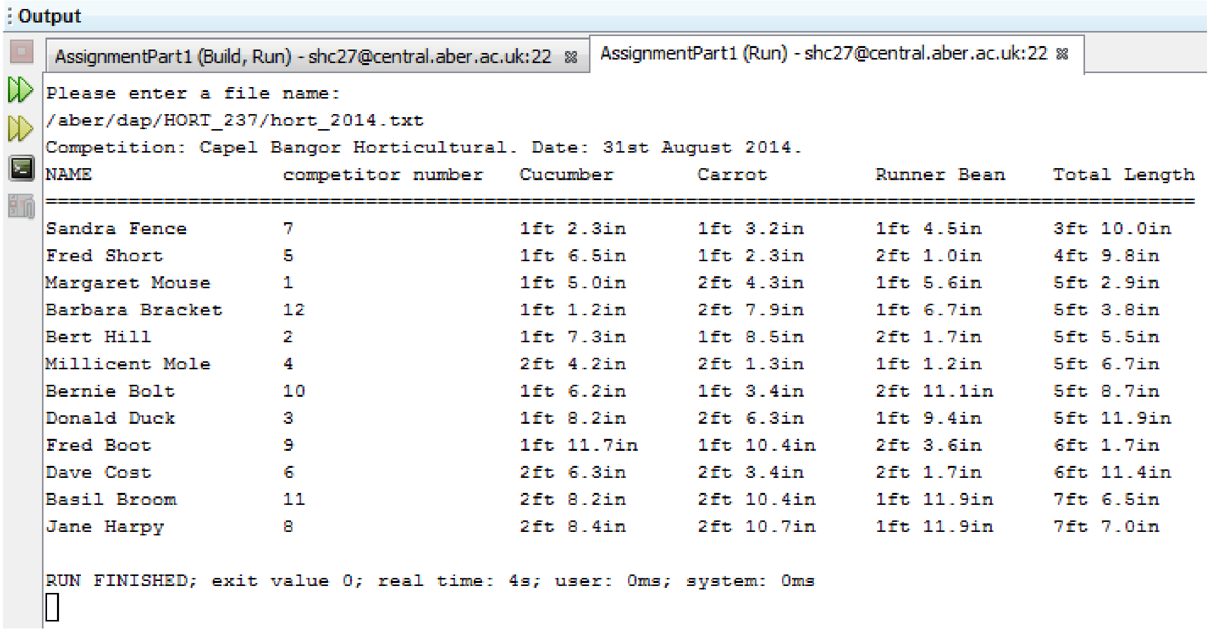
Output from the file: /aber/dap/HORT_237/hort_2013.txt


Output
AssignmentPart1 (Build, Run) - shc27@central.aber.ac.uk:22
AssignmentPart1 (Run) - shc27@central.aber.ac.uk:22

```
Please enter a file name:
/aber/dap/HORT_237/hort_2013.txt
Competition: Capel Sion Horticulture. Date: 6th September 2013.
NAME                competitor number   Cucumber       Carrot         Runner Bean    Total Length
=================================================================================================
Margaret Mouse      1                   1ft 0.5in      2ft 3.2in      1ft 4.5in      4ft 8.2in
Donald Duck         3                   2ft 0.4in      2ft 4.5in      1ft 6.2in      5ft 11.1in
Bert Hill           2                   1ft 7.2in      1ft 8.2in      2ft 11.7in     6ft 3.1in

RUN FINISHED; exit value 0; real time: 13s; user: 0ms; system: 0ms
```

[1.3]

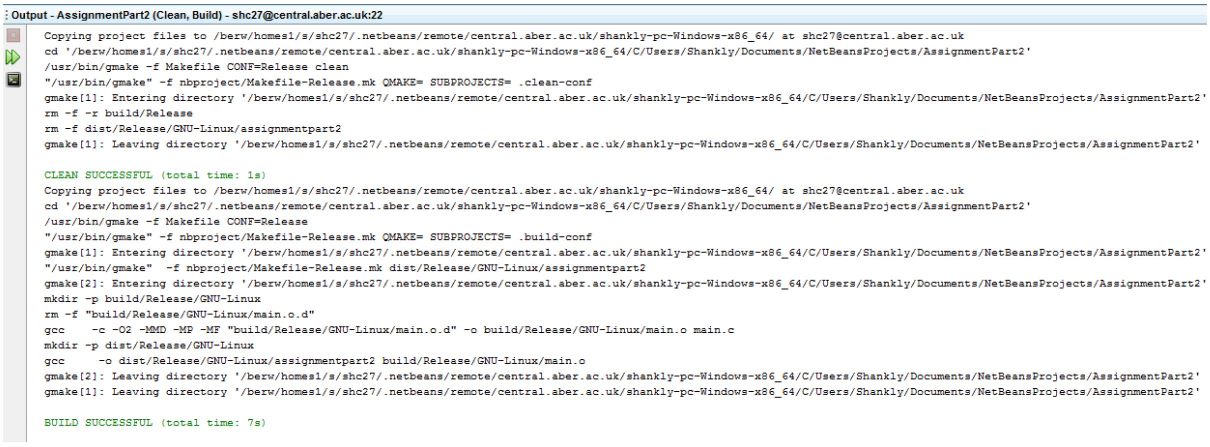
Output from the file: /aber/dap/HORT_237/hort_2014.txt


Output
AssignmentPart1 (Build, Run) - shc27@central.aber.ac.uk:22
AssignmentPart1 (Run) - shc27@central.aber.ac.uk:22

```
Please enter a file name:
/aber/dap/HORT_237/hort_2014.txt
Competition: Capel Bangor Horticultural. Date: 31st August 2014.
NAME                competitor number   Cucumber       Carrot         Runner Bean    Total Length
=================================================================================================
Sandra Fence        7                   1ft 2.3in      1ft 3.2in      1ft 4.5in      3ft 10.0in
Fred Short          5                   1ft 6.5in      1ft 2.3in      2ft 1.0in      4ft 9.8in
Margaret Mouse      1                   1ft 5.0in      2ft 4.3in      1ft 5.6in      5ft 2.9in
Barbara Bracket     12                  1ft 1.2in      2ft 7.9in      1ft 6.7in      5ft 3.8in
Bert Hill           2                   1ft 7.3in      1ft 8.5in      2ft 1.7in      5ft 5.5in
Millicent Mole      4                   2ft 4.2in      2ft 1.3in      1ft 1.2in      5ft 6.7in
Bernie Bolt         10                  1ft 6.2in      1ft 3.4in      2ft 11.1in     5ft 8.7in
Donald Duck         3                   1ft 8.2in      2ft 6.3in      1ft 9.4in      5ft 11.9in
Fred Boot           9                   1ft 11.7in     1ft 10.4in     2ft 3.6in      6ft 1.7in
Dave Cost           6                   2ft 6.3in      2ft 3.4in      2ft 1.7in      6ft 11.4in
Basil Broom         11                  2ft 8.2in      2ft 10.4in     1ft 11.9in     7ft 6.5in
Jane Harpy          8                   2ft 8.4in      2ft 10.7in     1ft 11.9in     7ft 7.0in

RUN FINISHED; exit value 0; real time: 4s; user: 0ms; system: 0ms
```

[2.1]

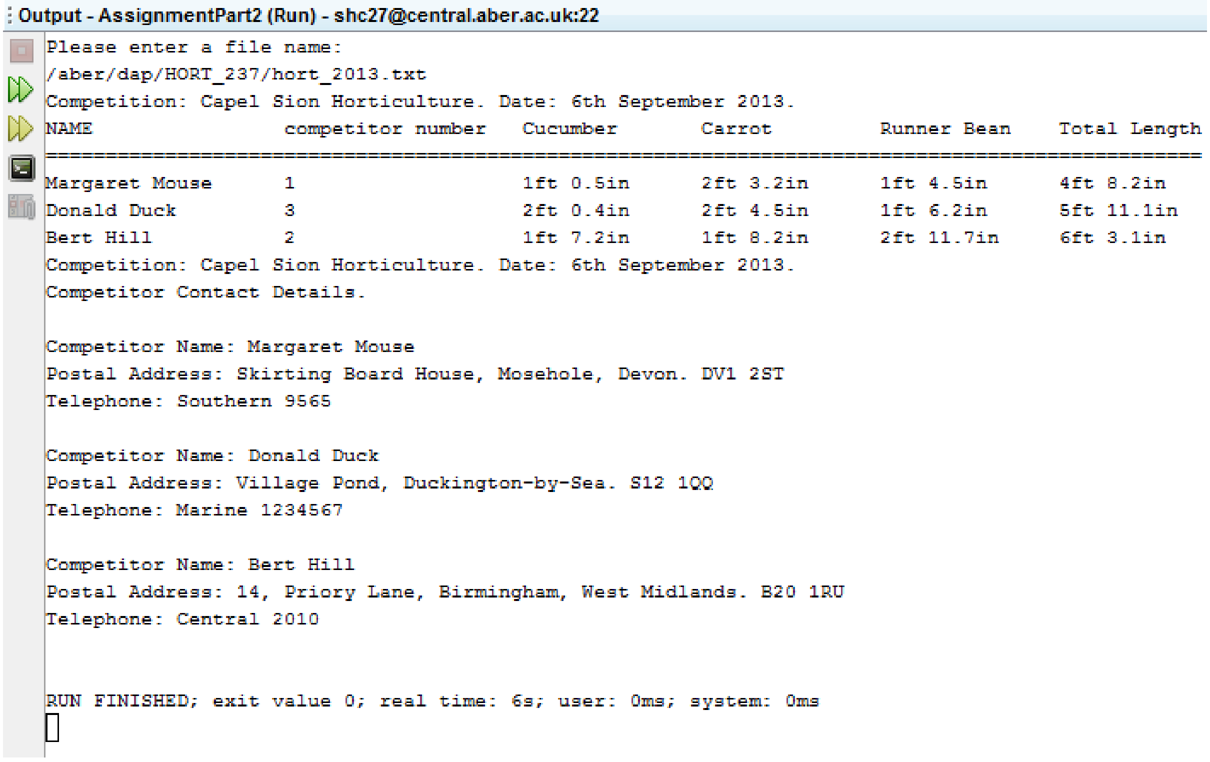
Clean and Build for part 2, using the build host


Output - AssignmentPart2 (Clean, Build) - shc27@central.aber.ac.uk:22

```
Copying project files to /berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/ at shc27@central.aber.ac.uk
cd '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/AssignmentPart2'
/usr/bin/gmake -f Makefile CONF=Release clean
"/usr/bin/gmake" -f nbproject/Makefile-Release.mk QMAKE= SUBPROJECTS= .clean-conf
gmake[1]: Entering directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/AssignmentPart2'
rm -f -r build/Release
rm -f dist/Release/GNU-Linux/assignmentpart2
gmake[1]: Leaving directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/AssignmentPart2'

CLEAN SUCCESSFUL (total time: 1s)
Copying project files to /berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/ at shc27@central.aber.ac.uk
cd '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/AssignmentPart2'
/usr/bin/gmake -f Makefile CONF=Release
"/usr/bin/gmake" -f nbproject/Makefile-Release.mk QMAKE= SUBPROJECTS= .build-conf
gmake[1]: Entering directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/AssignmentPart2'
"/usr/bin/gmake"  -f nbproject/Makefile-Release.mk dist/Release/GNU-Linux/assignmentpart2
gmake[2]: Entering directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/AssignmentPart2'
mkdir -p build/Release/GNU-Linux
rm -f "build/Release/GNU-Linux/main.o.d"
gcc    -c -O2 -MMD -MP -MF "build/Release/GNU-Linux/main.o.d" -o build/Release/GNU-Linux/main.o main.c
mkdir -p dist/Release/GNU-Linux
gcc     -o dist/Release/GNU-Linux/assignmentpart2 build/Release/GNU-Linux/main.o
gmake[2]: Leaving directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/AssignmentPart2'
gmake[1]: Leaving directory '/berw/homes1/s/shc27/.netbeans/remote/central.aber.ac.uk/shankly-pc-Windows-x86_64/C/Users/Shankly/Documents/NetBeansProjects/AssignmentPart2'

BUILD SUCCESSFUL (total time: 7s)
```

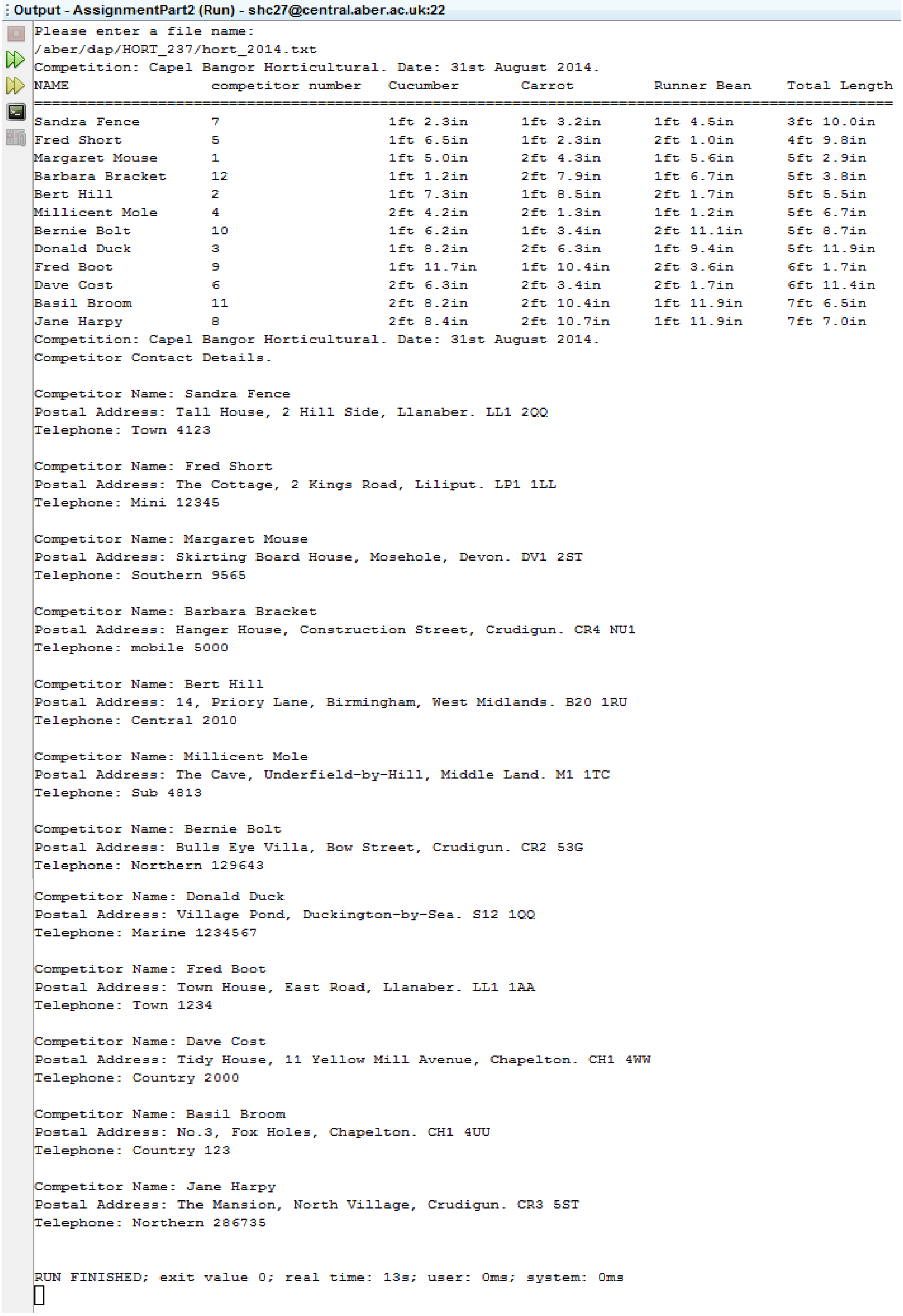
[2.2]

Output from the file: /aber/dap/HORT_237/hort_2013.txt


Output - AssignmentPart2 (Run) - shc27@central.aber.ac.uk:22

```
Please enter a file name:
/aber/dap/HORT_237/hort_2013.txt
Competition: Capel Sion Horticulture. Date: 6th September 2013.
NAME                competitor number   Cucumber       Carrot         Runner Bean    Total Length
=================================================================================================
Margaret Mouse      1                   1ft 0.5in      2ft 3.2in      1ft 4.5in      4ft 8.2in
Donald Duck         3                   2ft 0.4in      2ft 4.5in      1ft 6.2in      5ft 11.1in
Bert Hill           2                   1ft 7.2in      1ft 8.2in      2ft 11.7in     6ft 3.1in
Competition: Capel Sion Horticulture. Date: 6th September 2013.
Competitor Contact Details.

Competitor Name: Margaret Mouse
Postal Address: Skirting Board House, Mosehole, Devon. DV1 2ST
Telephone: Southern 9565

Competitor Name: Donald Duck
Postal Address: Village Pond, Duckington-by-Sea. S12 1QQ
Telephone: Marine 1234567

Competitor Name: Bert Hill
Postal Address: 14, Priory Lane, Birmingham, West Midlands. B20 1RU
Telephone: Central 2010


RUN FINISHED; exit value 0; real time: 6s; user: 0ms; system: 0ms
```

[2.3] Output from the file: /aber/dap/HORT_237/hort_2014.txt



Output - AssignmentPart2 (Run) - shc27@central.aber.ac.uk:22

```
Please enter a file name:
/aber/dap/HORT_237/hort_2014.txt
Competition: Capel Bangor Horticultural. Date: 31st August 2014.
NAME                competitor number   Cucumber       Carrot         Runner Bean    Total Length
=================================================================================================
Sandra Fence        7                   1ft 2.3in      1ft 3.2in      1ft 4.5in      3ft 10.0in
Fred Short          5                   1ft 6.5in      1ft 2.3in      2ft 1.0in      4ft 9.8in
Margaret Mouse      1                   1ft 5.0in      2ft 4.3in      1ft 5.6in      5ft 2.9in
Barbara Bracket     12                  1ft 1.2in      2ft 7.9in      1ft 6.7in      5ft 3.8in
Bert Hill           2                   1ft 7.3in      1ft 8.5in      2ft 1.7in      5ft 5.5in
Millicent Mole      4                   2ft 4.2in      2ft 1.3in      1ft 1.2in      5ft 6.7in
Bernie Bolt         10                  1ft 6.2in      1ft 3.4in      2ft 11.1in     5ft 8.7in
Donald Duck         3                   1ft 8.2in      2ft 6.3in      1ft 9.4in      5ft 11.9in
Fred Boot           9                   1ft 11.7in     1ft 10.4in     2ft 3.6in      6ft 1.7in
Dave Cost           6                   2ft 6.3in      2ft 3.4in      2ft 1.7in      6ft 11.4in
Basil Broom         11                  2ft 8.2in      2ft 10.4in     1ft 11.9in     7ft 6.5in
Jane Harpy          8                   2ft 8.4in      2ft 10.7in     1ft 11.9in     7ft 7.0in
Competition: Capel Bangor Horticultural. Date: 31st August 2014.
Competitor Contact Details.

Competitor Name: Sandra Fence
Postal Address: Tall House, 2 Hill Side, Llanaber. LL1 2QQ
Telephone: Town 4123

Competitor Name: Fred Short
Postal Address: The Cottage, 2 Kings Road, Liliput. LP1 1LL
Telephone: Mini 12345

Competitor Name: Margaret Mouse
Postal Address: Skirting Board House, Mosehole, Devon. DV1 2ST
Telephone: Southern 9565

Competitor Name: Barbara Bracket
Postal Address: Hanger House, Construction Street, Crudigun. CR4 NU1
Telephone: mobile 5000

Competitor Name: Bert Hill
Postal Address: 14, Priory Lane, Birmingham, West Midlands. B20 1RU
Telephone: Central 2010

Competitor Name: Millicent Mole
Postal Address: The Cave, Underfield-by-Hill, Middle Land. M1 1TC
Telephone: Sub 4813

Competitor Name: Bernie Bolt
Postal Address: Bulls Eye Villa, Bow Street, Crudigun. CR2 53G
Telephone: Northern 129643

Competitor Name: Donald Duck
Postal Address: Village Pond, Duckington-by-Sea. S12 1QQ
Telephone: Marine 1234567

Competitor Name: Fred Boot
Postal Address: Town House, East Road, Llanaber. LL1 1AA
Telephone: Town 1234

Competitor Name: Dave Cost
Postal Address: Tidy House, 11 Yellow Mill Avenue, Chapelton. CH1 4WW
Telephone: Country 2000

Competitor Name: Basil Broom
Postal Address: No.3, Fox Holes, Chapelton. CH1 4UU
Telephone: Country 123

Competitor Name: Jane Harpy
Postal Address: The Mansion, North Village, Crudigun. CR3 5ST
Telephone: Northern 286735


RUN FINISHED; exit value 0; real time: 13s; user: 0ms; system: 0ms
```

Bibliography

C Binary Tree with an Example C Code (Search, I. (2015). *C Binary Tree with an Example C Code (Search, Delete, Insert Nodes)*. [online] Thegeekstuff.com. Available at: http://www.thegeekstuff.com/2013/02/c-binary-tree/ [Accessed 10 Dec. 2015].

Cite This For Me, (2015). *Save Time and Improve your Marks with CiteThisForMe, The No. 1 Citation Tool*. [online] Available at: https://www.citethisforme.com/ [Accessed 10 Dec. 2015].

Cs.arizona.edu, (2015). *Indian Hill C Style Manual*. [online] Available at: https://www.cs.arizona.edu/~mccann/cstyle.html [Accessed 11 Dec. 2015].