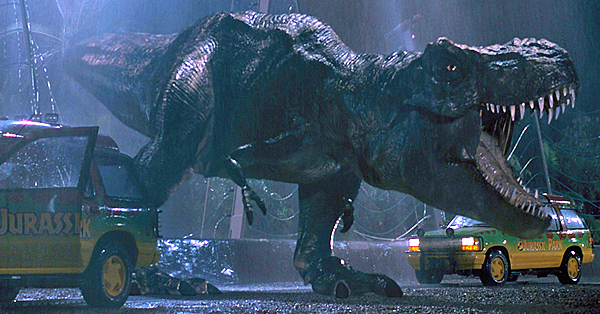
PROGRAM 5 / CSC 1300

# Jurassic Park

 **http://resizing.flixster.com/qld6HGa6NBb5IfhCziiWEvZ91sc=/700x380/dkpu1ddg7pbsk.cloudfront.net/site/10/27/24/10272456\_ori.jpg**

## Assignment Date: Monday, November 18, 2019

## Due Date: Monday, December 2, 2019 (by 11:59pm) NO LATE SUBMISSIONS!!

### What to Turn In:

* Zip & upload to PROGRAM 5 folder in ilearn dropbox
* **prog5.cpp**, **prog5.h**, **functions.cpp**
* **dinosaurs.txt** – this should be a text file with at least FIVE dinosaurs that you created with your program (data should be separated by #).

### Description:

Your assignment is to write a program for a dinosaur zookeeper (think Jurassic Park). The zoo can hold up to 100 Dinosaurs. Information about each Dinosaur needs will be kept and manipulated in this program. The user should be able to load dinosaur information from any file he or she chooses, add dinosaurs manually, delete a dinosaur, print dinosaur information to either a file or to the screen, or print a cost analysis of each dinosaur and the total cost to house and take care of these dinosaurs.

### What’s New in Program 5:

* Structures (structure array, nested structures)

## PROGRAM SPECIFICATIONS

### Structures: (define all structures in Prog5.h)

You will need to create two structures. One is called **Cost**. Cost will have the following members:

* The number of hours it takes to take care of a specific Dinosaur.
* The cost (per hour) of taking care of this Dinosaur.
* The cost of food to feed this Dinosaur for one week.
* The cost of materials/supplies (grooming, medical) for this Dinosaur for one week

The second structure is called **Dinosaurs**. Dinosaurs will have the following members:

* The name of the Dinosaur
* The description of the Dinosaur
* The average length of the Dinosaur (in feet)
* The average height of the Dinosaur (in feet)
* The location of the Dinosaur (example: its origin or where they are commonly found)
* If the Dinosaur is dangerous (Boolean variable)
* A variable to hold the Cost structure members

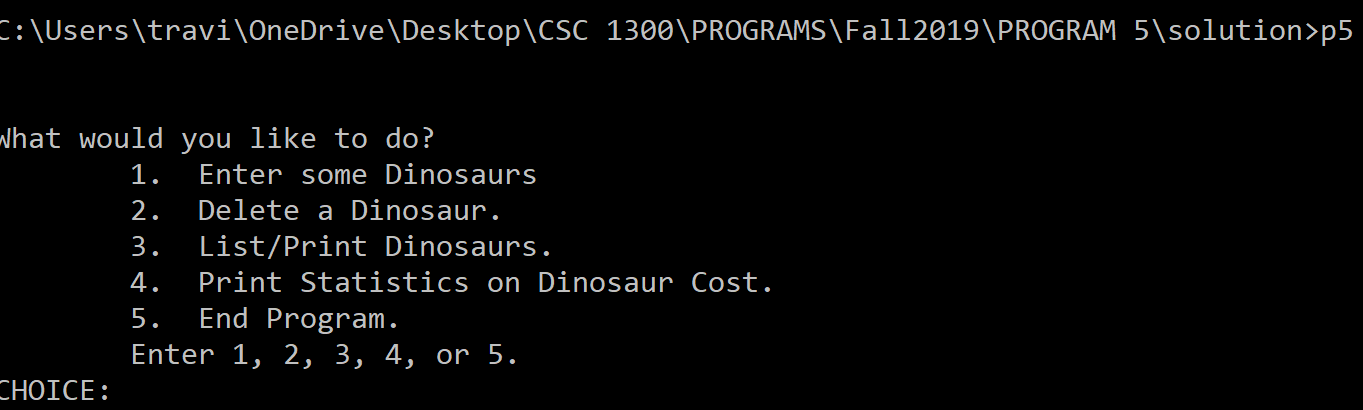
### The Main Function: (define main function in Prog5.cpp)

You will need to create an array of 100 elements of the Dinosaurs data type. You will also need to create a variable that will keep up with the current number of Dinosaurs.

The main function will display a menu of five options:

1. Enter some Dinosaurs.
2. Delete a Dinosaur
3. List/Print Dinosaurs.
4. Print Dinosaur Costs.
5. End Program.

Make sure you always validate all user choices in the whole program before proceeding to do what the user wants. Each menu choice will call a function that you will create.



If the user chooses option 1, then your program will call the enterDinosaur function. If the user chooses option 2, then your program will call the deleteDinosaur function. If the user chooses option 3, then your program will call the printDinosaurs function. If the user chooses option 4, then your program will call the printStatistics function. If the user chooses option 5, then your program will ask the user if they wish to save their dinosaur list to a file. If they choose yes, then your program should call the saveDinosaursToFile function and then end. If they choose no, then your program should just end.

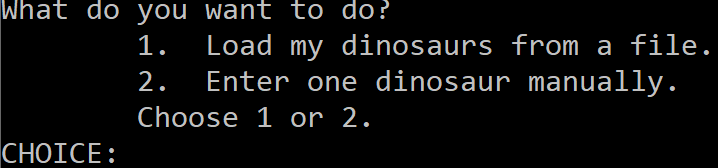
### enterDinosaurs function (define all other functions in functions.cpp)

The enterDinosaurs() function takes two parameters: the number of Dinosaurs currently loaded in the Dinosaurs array and the Dinosaurs array. The function will return the new number of Dinosaurs. Before trying to add any Dinosaurs, this function should first check to make sure the number of Dinosaurs isn’t already 100. Because if it is, then your program should not add any Dinosaurs, but should instead tell the user that their zoo is at full capacity and that they are not able to add Dinosaurs. Then the function should end.

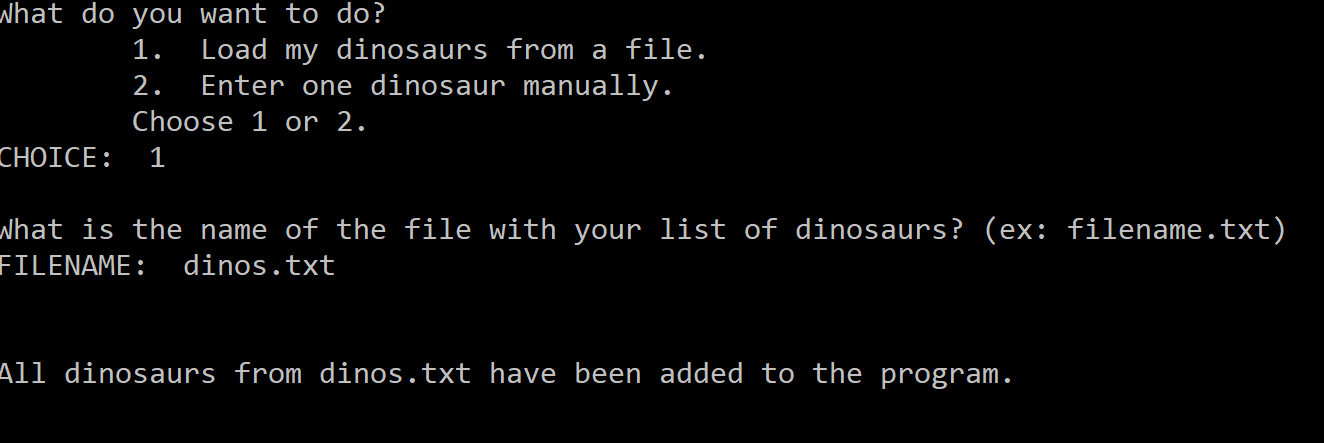
Otherwise, your program will display a menu asking the user if they would like to do one of the following:

1. Load my Dinosaurs from a file.
2. Enter one Dinosaur manually.

Validate the user’s choice.



If the user chooses option 1, then ask the user what the name of the file is that they would like to load the Dinosaurs from. Then open their file. Check if the file could open before reading from it.



Read each Dinosaur from the file and place them in the Dinosaurs array, making sure that you increment the number of Dinosaurs each time a Dinosaur is added. When you are reading from the file, everything read in will have to be read in as a string. Some of the Dinosaur members are strings, so that won’t be a problem. However, some of the Dinosaur members are floats. So, this function will have to call the convertToFloat() function (provided later) in order to convert the string to a float and then placed in the Dinosaurs array. Then return the number of Dinosaurs at the end of the function.

If the user chooses option 2, then you will want to start a loop so that the user can add multiple Dinosaurs manually without returning to the main menu. Ask the user for the following: (make sure you place each bit of information in the correct place in the Dinosaurs array)

* NAME:
* DESCRIPTION:
* AVERAGE LENGTH (in feet):
* AVERAGE HEIGHT (in feet):
* LOCATION:
* IS IT A DANGEROUS DINOSAUR? (y or n)
* How many hours do you spend caring for the [insert Dinosaur name here]?  
  NUM HOURS:
* What is the cost per hour for caring for the [insert Dinosaur name here]?  
  COST PER HOUR:
* How much money do you spend on food for the [insert Dinosaur name here]?  
  FOOD COST:
* How much money do you spend on grooming and medical supplies for the [insert Dinosaur name here]?  
  SUPPLY COST:

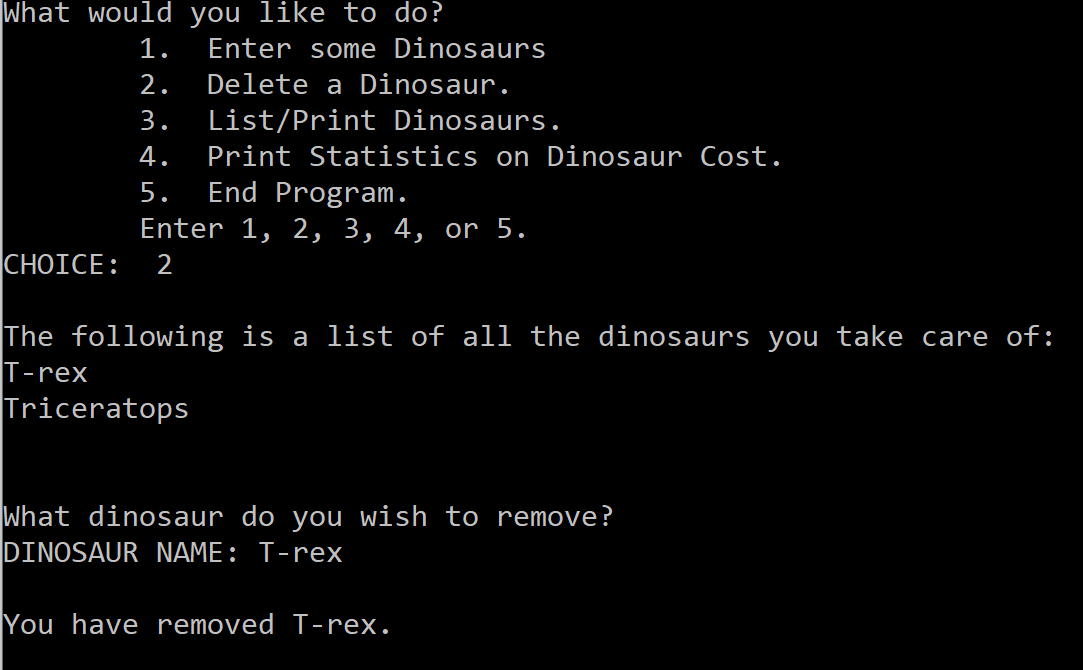
Then, increment the number of Dinosaurs by one. Then, ask the user if they want to add another Dinosaur. If they do, then repeat this option. If not, then just return the number of Dinosaurs.

### deleteDinosaur function

The deleteDinosaur() function has two parameters: the current number of Dinosaurs in the Dinosaurs array and the Dinosaurs array. This function returns the new number of Dinosaurs.

First, this function will say “The following is a list of all the Dinosaurs you take care of: “ and then it will say the name of each Dinosaur. Then, your program will ask the user  
What Dinosaur do you wish to remove?  
DINOSAUR NAME:

Your program should then read in the name and place it in a variable. Then, this function will call the moveArrayElements function, which will take care of removing the Dinosaur. The moveArrayElements function returns a Boolean value to tell if the Dinosaur was removed or not. Check to see if the Dinosaur was removed. If it was, then decrement the number of Dinosaurs and print out “You have removed [insert Dinosaur name here].” Then return the new number of Dinosaurs.



### moveArrayElements function

The moveArrayElements() function has the following parameters: a string with the name of the Dinosaur that the user wishes to remove, the current number of Dinosaurs in the Dinosaur array, and the Dinosaurs array. This function returns a Boolean variable that is true if the Dinosaur was removed and false if it was not removed.

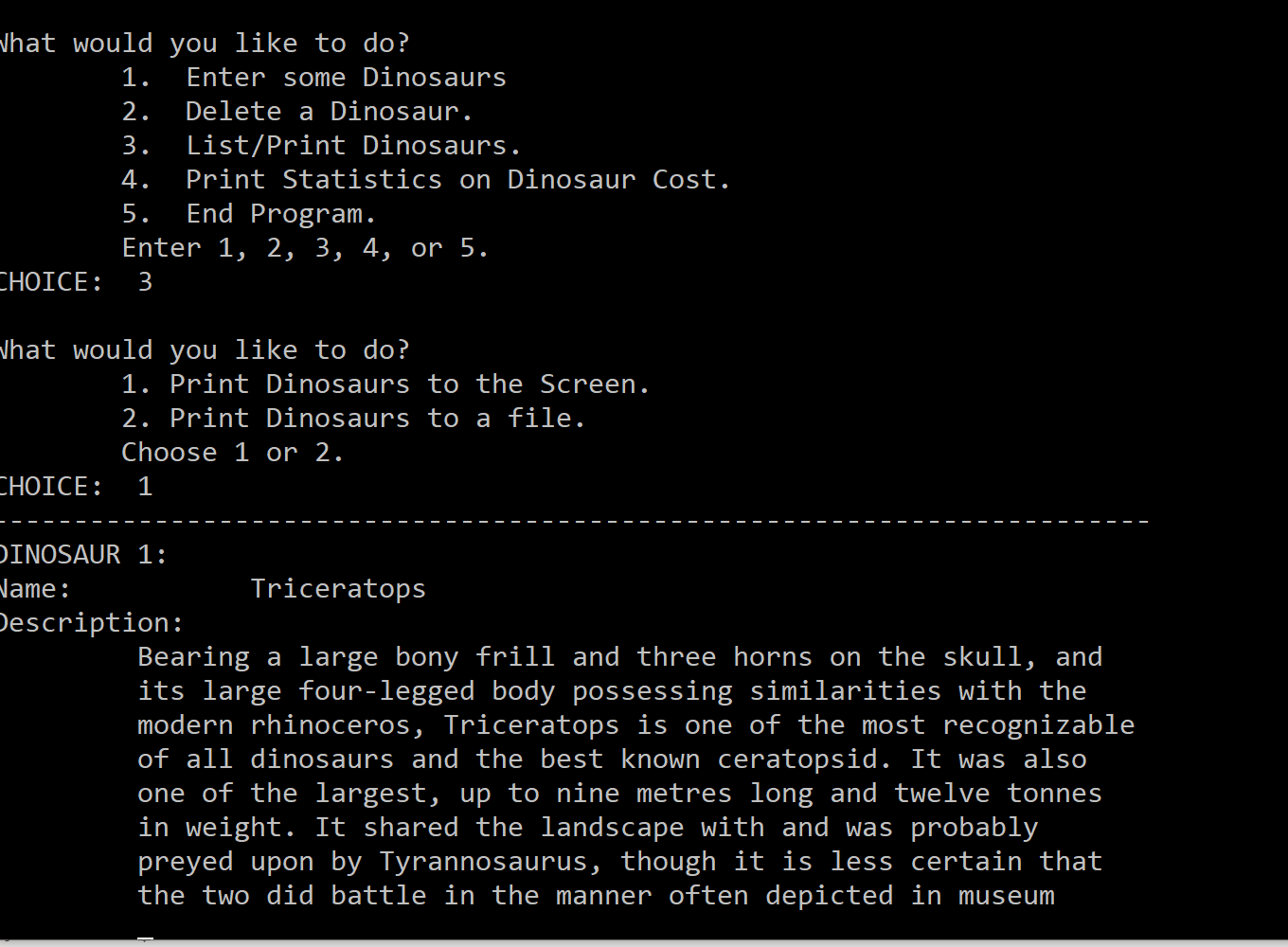
This function should first find the subscript number of the Dinosaur that needs to be removed. Once that is found, then you know there is a Dinosaur in the Dinosaurs array by that name and that your program will be able to remove it. If your program cannot find the Dinosaur in the array, then you return false from this function. Otherwise, this function should now overwrite the element with the Dinosaur to delete (x) with the next element in the array (x+1), moving each element after the deleted element to the left. Then return true that the Dinosaur was found & removed.

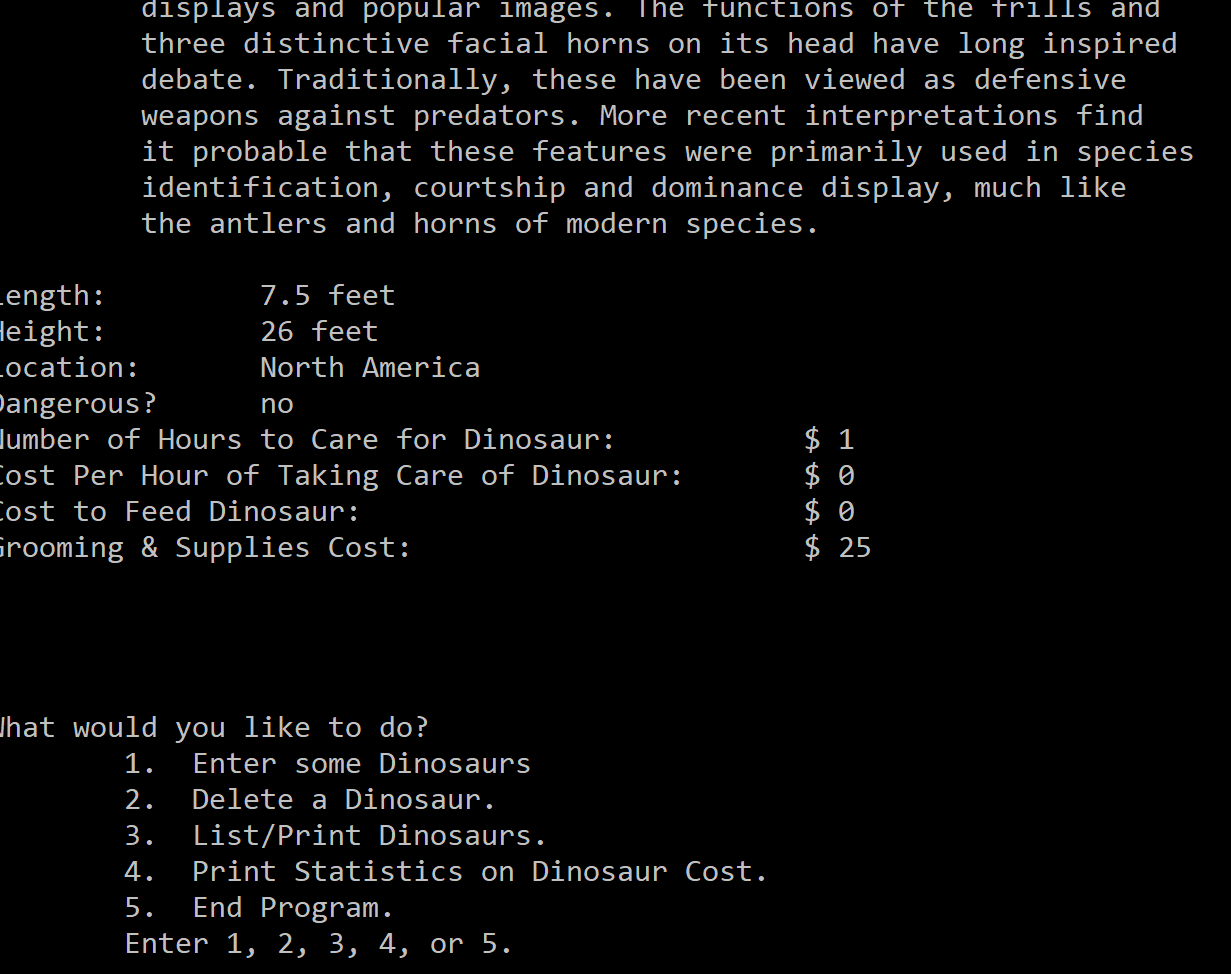
### printDinosaurs function

The printDinosaurs() function is a void function and contains the following parameters: number of Dinosaurs currently in the Dinosaurs array and the Dinosaurs array. The printDinosaurs() function will first display a menu to the user containing the following options:

* Print Dinosaurs to the Screen.
* Print Dinosaurs to a File.

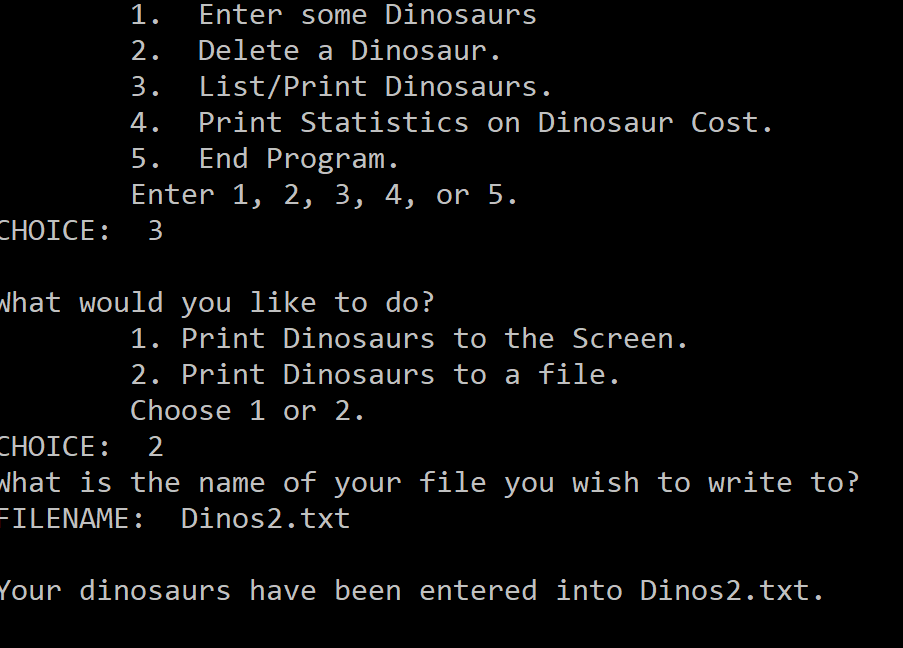
Validate the user’s choice. If the user chooses option 1, then you will print out all the Dinosaurs in the Dinosaurs array to the screen in the following format:

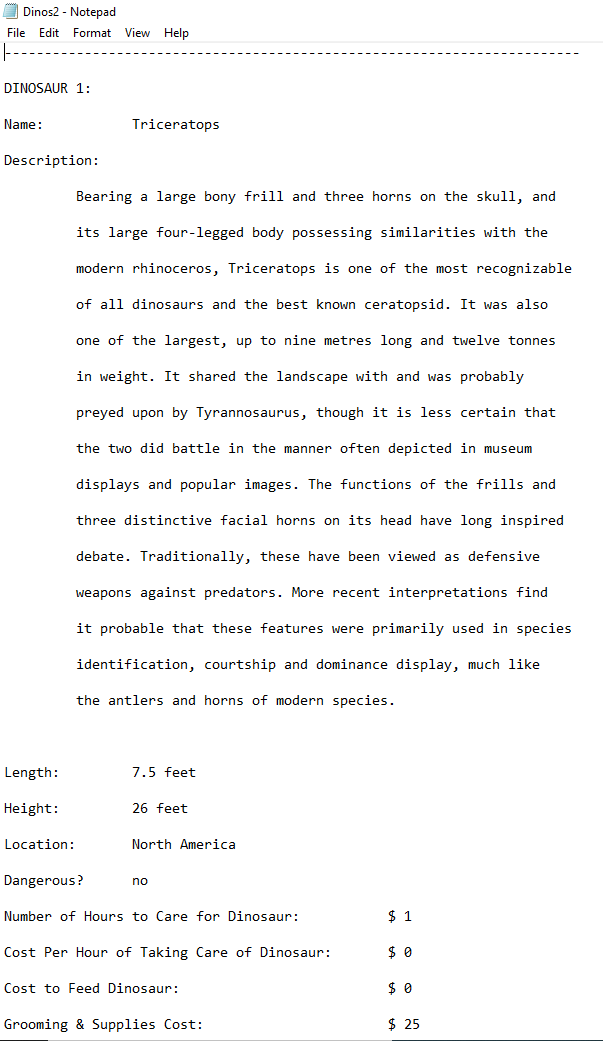




Notice that the description does word wrapping. In other words, each word is not split up – they are kept together. You get a bonus of 5 points if you can do word wrapping for your descriptions.

If the user selects option 2, then your program should ask the user what is the name of the file that they wish to print the Dinosaurs to. Then, open this file. You do not have to check for errors for this file because it probably will not yet exist. Then, write all the information to the file like you did to the screen if the user had chosen option 1. Example below:





Then, your program should write “Your Dinosaurs have been written to [insert filename here].”

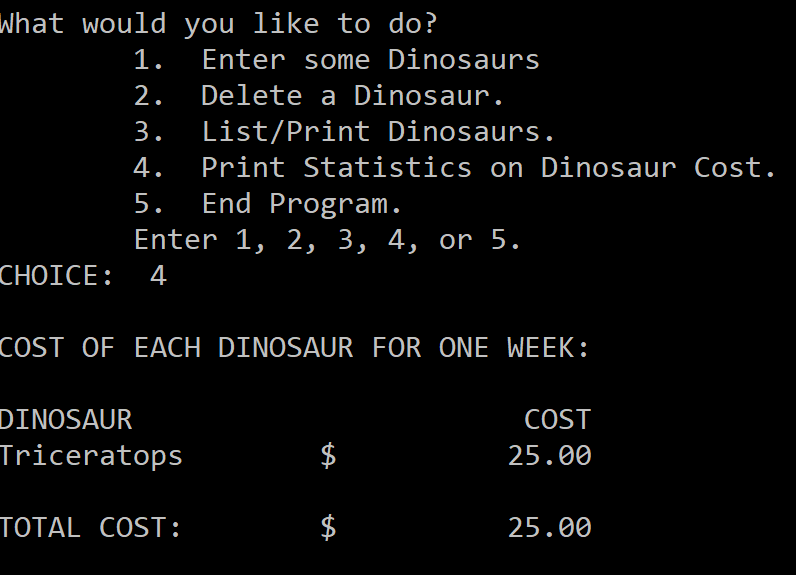
### printStatistics function

The printStatistics() function is a void function and it contains two parameters: the current number of Dinosaurs in the Dinosaur array and the Dinosaurs array.

This function should print out the total cost of each Dinosaur in a table. To figure out the cost of a Dinosaur, use the formula:

Cost = numHours \* costPerHour + foodCost + materialCost

Then, this function should also keep a running total of the total cost of all the Dinosaurs.



### saveDinosaursToFile function

The saveDinosaursToFile() function is a void function and it contains two parameters: the current number of Dinosaurs in the Dinosaurs array and the Dinosaurs array. The function should ask the user what the name of the file that they wish to save their Dinosaurs to. The function should then open that file and write out all the Dinosaur data in the following order:

* Name
* Description
* Avg. Length
* Avg. Height
* Location
* Dangerous
* numHours
* costPerHour
* foodCost
* MaterialCost

There should be no newlines or endlines in the file. All data should be separated by a pound sign (#) instead of a space.

After printing all data from the Dinosaurs array to the file, this function should print out a message to standard output saying “Your Dinosaurs were successfully saved to the [insert filename here] file.”

### convertToFloat function

The convertToFloat() function will need to be used when reading data from a file and inputting the data into the Dinosaurs array (so this function will be called from the enterDinosaurs function). I am providing this function for you. You just need to insert it into your program so you can use it.

float convertToFloat(string s)

{

istringstream i(s);

float x;

if (!(i >> x))

x = 0;

return x;

}

You will need to insert the following header file to use the convertToFloat function:

#include <sstream>