**Problem 2:**

**Animal.cpp**

#include <iostream>

#include <string>

using namespace std;

//string unsortedArray;

int numPeople;

int numberOfNames();

void fillArray(string tempArr[]);

void printArray(string tempArr[]);

void sortArray(string tempArr[], string sortedTempArr[]);

void writeFile(string tempArr[]);

struct Animal{

string name;

string gender;

int age;

int price;

string type;

};

Animal cheapest(string type, Animal a[], int size){ //assuming 'size' is the size of the array

string tempType;

int countToFirstType = 0;

int currentCheapest;

int priceToCompare;

int firstType = 0;

int animalNum;

for (int i = 0; i < size; i++){

tempType = a[i].type;

if (tempType == type){ //To check if the type of animals matches

if(firstType == 0){ //To ensure we set the first type's price

currentCheapest = a[i].price;

firstType =1;

animalNum = i;

}

else{ //only enters if the first type's price has been established

priceToCompare = a[i].price;

if(currentCheapest > priceToCompare){

currentCheapest = priceToCompare;

animalNum = i;

}

}

}

}

return a[animalNum];

}

bool livetogether(Animal a, Animal b){

if (a.type == b.type){

return true;

}

else{

return false;

}

}

int main(){

//declaring test data

Animal Dog;

Animal Dog2;

Animal Dog3;

Animal Dog4;

Animal Cat;

Animal Cat2;

Animal Cat3;

Animal Cat4;

//initializing test data

Cat.type = "cat";

Cat.price = 5300;

Cat2.type = "cat"; //Cheapest Cat is Cat 2

Cat2.price = 2000;

Cat2.name = "BigDog";

Cat3.type = "cat";

Cat3.price = 4500;

Cat4.type = "cat";

Cat4.price = 7000;

//////////////////////

Dog.type = "dog";

Dog.price = 10000;

Dog2.type = "dog";

Dog2.price = 12000;

Dog3.type = "dog"; //Cheapest Dog is Dog 3

Dog3.price = 9000;

Dog3.name = "Howard";

Dog4.type = "dog";

Dog4.price = 12000;

Animal arrAnimal[8];

arrAnimal[0] = Cat;

arrAnimal[1] = Cat2;

arrAnimal[2] = Cat3;

arrAnimal[3] = Cat4;

arrAnimal[4] = Dog;

arrAnimal[5] = Dog2;

arrAnimal[6] = Dog3;

arrAnimal[7] = Dog4;

Animal inexpensiveDog = cheapest("dog", arrAnimal, 8);

cout << inexpensiveDog.name << endl;

Animal inexpensiveCat = cheapest("cat", arrAnimal, 8);

cout << inexpensiveCat.name << endl;

bool canTheyLiveTogether = livetogether(Cat, Cat4);

cout << canTheyLiveTogether << endl;

bool theyCant = livetogether(Cat, Dog);

cout << theyCant << endl;

}