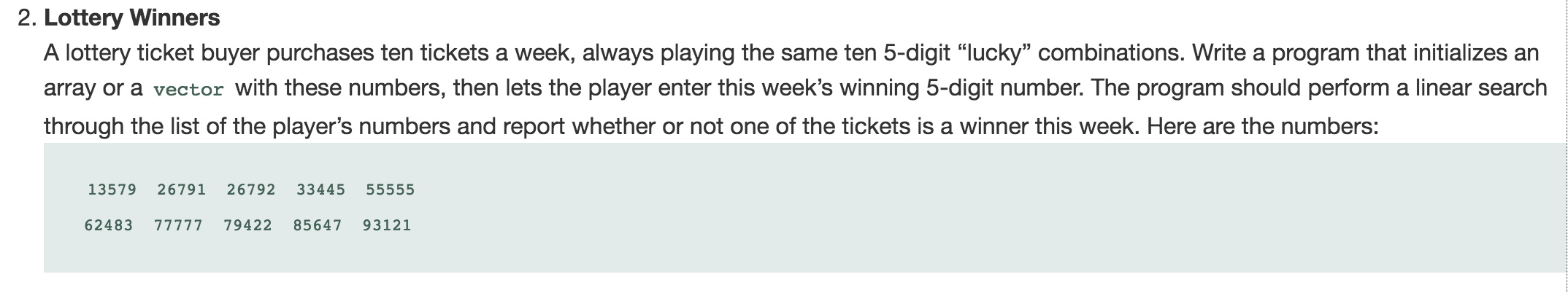
COMSC-110 Lab-7 Due: July-19-2020

Format:

1- After each problem statement, copy and paste the source code.

2- After the source code, paste the screen shot of the result.

3- Submit either word or pdf file (ONLY ONE FILE – NO MAC PAGE FILES)



Answer:

#include <iostream>

#include <cmath>

using namespace std;

int main() {

int luckyNumb[10] = {13579, 26791,26795,33445,55555,62483,77777,79422,85647,93121};

int winningNumber;

cout << "Enter the winning number:" << endl;

cin >> winningNumber;

while (static\_cast<int>(log10(winningNumber)+1) < 5){

cout << "There should be 5 digit, enter the number again:" << endl;

cin >> winningNumber;

}

for(int temp : luckyNumb){

if(temp == winningNumber){

cout << "You are a winner!" << endl;

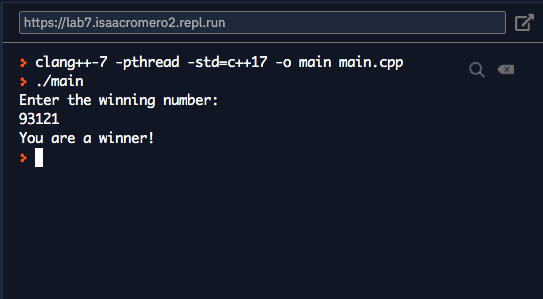
return 0;

}

}

cout << "Sorry, no wins today!" << endl;

}





Answer:

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

void swap(string&, string&);

void selectionSort(string[], int);

void display (string[] , int);

int main()

{

  const int NUM\_NAMES = 20;

  string names[NUM\_NAMES] = {"Collins, Bill", "Smith, Bart", "Allen, Jim",

"Griffin, Jim", "Stamey, Marty", "Rose, Geri","Taylor, Terri", "Johnson, Jill", "Allison, Jeff","Looney, Joe", "Wolfe, Bill", "James, Jean","Weaver, Jim", "Pore, Bob", "Rutherford, Greg","Javens, Renee", "Harrison, Rose", "Setzer, Cathy", "Pike, Gordon", "Holland, Beth" };

cout << "Unorder\n" << endl;

display(names,NUM\_NAMES);

cout << "\nOrder \n" << endl;

//  // Insert your code to complete this program

selectionSort(names,NUM\_NAMES);

display(names,NUM\_NAMES);

  return 0;

}

void display(string arr[], int SIZE){

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

if (i % 4 == 0 && i != 0) cout << endl;

cout << left << setw(15) << arr[i] + "\t";

}

}

void selectionSort(string arr[], const int SIZE){

for (int i = SIZE-1; i > 0; i--){

string \*max = &arr[i];

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

if(arr[j] > \*max) max = &arr[j];

}

swap(arr[i], \*max);

}

}

void swap(string &s1, string &s2){

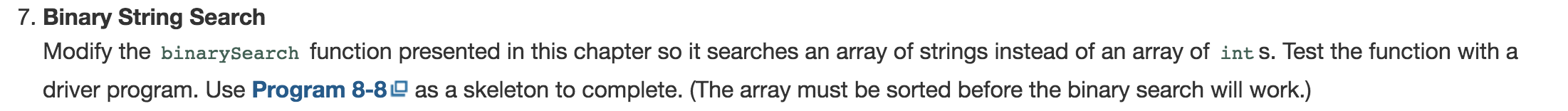
string temp = s1;

s1 = s2;

s2 = temp;

}





Answer:

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

void swap(string&, string&);

void selectionSort(string[], int);

void display (string[] , int);

int binarySearch(string [], const int, string);

int main()

{

  const int NUM\_NAMES = 20;

  string names[NUM\_NAMES] = {"Collins, Bill", "Smith, Bart", "Allen, Jim",

"Griffin, Jim", "Stamey, Marty", "Rose, Geri",

"Taylor, Terri", "Johnson, Jill", "Allison, Jeff",

"Looney, Joe", "Wolfe, Bill", "James, Jean",

"Weaver, Jim", "Pore, Bob", "Rutherford, Greg",

"Javens, Renee", "Harrison, Rose", "Setzer, Cathy",

"Pike, Gordon", "Holland, Beth" };

// Insert your code to complete this program

string target;

cout << "What are we searching for? (eg Collins, Bill)" << endl;

getline(cin, target,'\n');

selectionSort(names,NUM\_NAMES);

display(names,20);

if (binarySearch(names,NUM\_NAMES, target) > 0 ){

cout << "\nFound it" << endl;

}else cout << "\nnot here" << endl;

  return 0;

}

int binarySearch(string names[] , const int SIZE, string target){

int left = 0;

int right = SIZE -1;

int middle;

while (left < right){

middle = (left+right)/2;

if( names[middle]==target) return middle; // return the last position where the string was found

else if(target > names[middle])left = middle+1;

else right = middle-1;

}

return -1; // if loops ends then string was not found

}

void display(string arr[], int SIZE){

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

if (i % 4 == 0 && i != 0) cout << endl;

cout << left << setw(15) << arr[i] + "\t";

}

}

void selectionSort(string arr[], const int SIZE){

for (int i = SIZE-1; i > 0; i--){

string \*max = &arr[i];

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

if(arr[j] > \*max) max = &arr[j];

}

swap(arr[i], \*max);

}

}

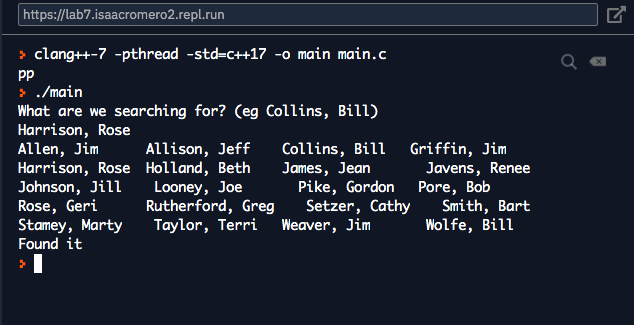
void swap(string &s1, string &s2){

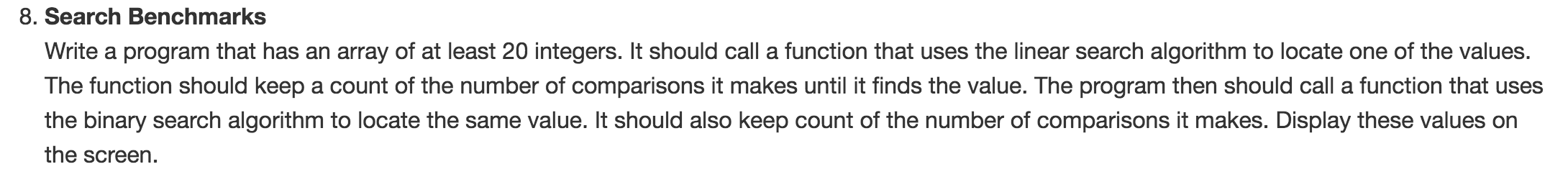
string temp = s1;

s1 = s2;

s2 = temp;

}





Answer:

#include <iostream>

#include <iomanip>

#include <stdlib.h>

#include <ctime>

using namespace std;

void swap(int&, int&);

void selectionSort(int[], int);

void display (int[] , int);

int binarySearch(int [], const int, int);

int linearSearch(int[],const int , int);

void fillArray (int[], const int);

int main()

{

srand(time(NULL));

  const int NUMS = 20;

  int nums[NUMS] = {0};

fillArray(nums,NUMS);

display(nums, NUMS);

cout << endl;

// Insert your code to complete this program

int target;

cout << "What number are you looking for (1-100)" << endl;

cin >> target;

while( target < 0 && target > 100){

cout << "The array only has range 1 -100. Enter a number between 1-100:" << endl;

cin >> target;

}

if (linearSearch(nums,NUMS, target) > 0 ){

cout << "Found it" << endl;

}

selectionSort(nums, NUMS);

display(nums,NUMS);

if(binarySearch(nums, NUMS, target) > 0){

cout << "Found it!" << endl;

}

  return 0;

}

void fillArray(int num[], const int SIZE){

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

num[i] = rand()%100 +1;

}

}

int linearSearch(int num[] , const int SIZE, int target){

int counter = 0;

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

if(num[i] == target){

counter++;

cout << "Number of comparasons: " << counter << endl;

return i; // returning the position of target

}

else{

counter++;

}

}

return -1;

}

int binarySearch(int names[] , const int SIZE, int target){

int counter = 0;

int left = 0;

int right = SIZE -1;

int middle;

while (left < right){

middle = (left+right)/2;

if( names[middle]==target){

counter++;

cout << "\nComparisons: " << counter << endl;

return middle; // return the last position where the number was found

}

else if(target > names[middle]){

counter+=2;

left = middle+1;

}else{

counter+=2;

right = middle-1;

}

}

return -1; // if loops ends then number was not found

}

void display(int arr[], int SIZE){

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

if (i % 4 == 0 && i != 0) cout << endl;

cout << arr[i] << "\t";

}

}

void selectionSort(int arr[], const int SIZE){

for (int i = SIZE-1; i > 0; i--){

int \*max = &arr[i];

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

if(arr[j] > \*max) max = &arr[j];

}

swap(arr[i], \*max);

}

}

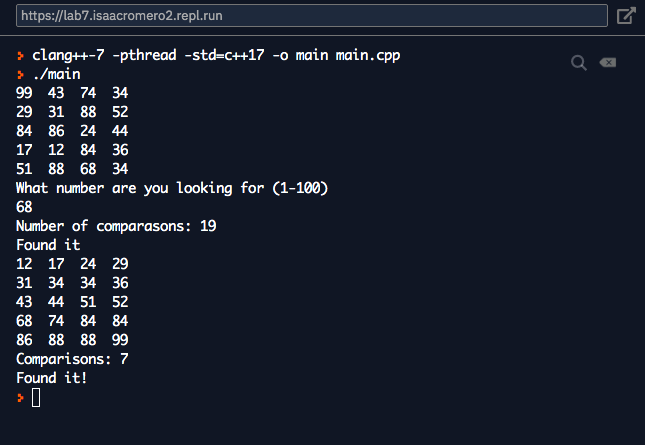
void swap(int &s1, int &s2){

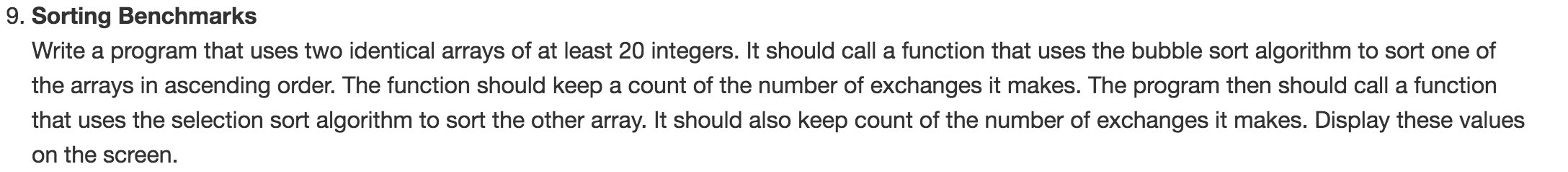
int temp = s1;

s1 = s2;

s2 = temp;

}





#include <iostream>

#include <iomanip>

#include <stdlib.h>

#include <ctime>

using namespace std;

void swap(int&, int&);

void display (int[] , int);

void fillArray (int[], const int);

void selectionSort(int[], int);

void bubbleSort(int[], const int);

int main()

{

srand(time(NULL));

  const int NUMS = 20;

  int nums[NUMS] = {0};

int copynums[NUMS]={0};

fillArray(nums,NUMS);

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

copynums[i] = nums[i];

}

cout << "Array:" << endl;

display(nums, NUMS);

// Insert your code to complete this program

cout << "Selection Sort: " << endl;

selectionSort(nums, NUMS);

display(nums,NUMS);

cout << "Bubble Sort: " << endl;

bubbleSort(copynums, NUMS);

display(copynums, NUMS);

  return 0;

}

void fillArray(int num[], const int SIZE){

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

num[i] = rand()%100 +1;

}

}

void display(int arr[], int SIZE){

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

if (i % 5 == 0 && i != 0) cout << endl;

cout << arr[i] << "\t";

}

cout << endl;

}

void bubbleSort(int arr[], const int SIZE){

int counter = 0;

int pause;

for (int i = SIZE -1 ; i > 0; i--){

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

if(arr[j] > arr[j+1]){

swap(arr[j], arr[j+1]);

counter++;

}

}

}

cout << "Number of Swaps: " << counter << endl;

}

void selectionSort(int arr[], const int SIZE){

int counter = 0;

for (int i = SIZE-1; i > 0; i--){

int \*max = &arr[i];

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

if(arr[j] > \*max) max = &arr[j];

}

swap(arr[i], \*max);

counter++;

}

cout << "Number of Swaps: " << counter << endl;

}

void swap(int &s1, int &s2){

int temp = s1;

s1 = s2;

s2 = temp;

}

