CSE 241

PS # 2

04.03.2020

Q1)

Write a program that converts from 24-hour notation to 12-hour notation. For example, it should convert 14:25 to 2:25 P.M. The input is given as two integers. There should be at least three functions: one for input, one to do the conversion, and one for output. Record the A.M./P.M. information as a value of type char , 'A' for A.M. and 'P' for P.M. Thus, the function for doing the conversions will have a call-by-reference formal parameter of type char to record whether it is A.M. or P.M. (The function will have other parameters as well.) Include a loop that lets the user repeat this computation for new input values again and again until the user says he or she wants to end the program.

Q2)

You are a contestant on a game show and have won a shot at the grand prize. Before you are three doors. $1,000,000 in cash has randomly been placed behind one door. Behind the other two doors are the consolation prizes of dishwasher detergent. The game show host asks you to select a door, and you randomly pick one. However, before revealing the prize behind your door, the game show host reveals one of the other doors that contains a consolation prize. At this point, the game show host asks if you would like to stick with your original choice or to switch to the remaining door.

Write a function to simulate the game show problem. Your function should randomly select locations for the prizes, select a door at random chosen by the contestant, and then determine whether the contestant would win or lose by sticking with the original choice or switching to the remaining door. You may wish to create additional functions invoked by this function. N ext, modify your program so that it simulates playing 10,000 games. Count the number of times the contestant wins when switching versus staying. If you are the contestant, what choice should you make to optimize your chances of winning the cash, or does it not matter?

Q3)

You would like to know how fast you can run in miles per hour. Your treadmill will tell you your speed in terms of a pace (minutes and seconds per mile, such as “5:30 mile”) or in terms of kilometers per hour (kph). W rite an overloaded function called convertToMPH. The first definition should take as input two integers that represent the pace in minutes and seconds per mile and return the speed in mph as a double. The second definition should take as input one double that represents the speed in kph and return the speed in mph as a double. One mile is approximately 1.61 kilometers. Write a driver program to test your function.

Q4)

Consider a text file named scores.txt that contains player scores for a game. A possible sample is shown here where Ronaldo’s best score is 10400, Didier’s best score is 9800, etc.

Ronaldo

10400

Didier

9800

Pele

12300

Kaka

8400

Cristiano

8000

Write a function named **getHighScore** that takes a string reference parameter and an integer reference parameter. The function should scan through the file and set the reference parameters to the name of the player with the highest score and the corresponding score.

Q5)

The standard deviation of a list of numbers is a measure of how much the numbers deviate from the average. If the standard deviation is small, the numbers are clustered close to the average. If the standard deviation is large, the numbers are scattered far from the average. The standard deviation, S , of a list of N numbers xi is defined as follows,

$$S = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N}}$$

where x is the average of the N numbers x1 , x2 , …. Define a function that takes a partially filled array of numbers as its argument and returns the standard deviation of the numbers in the partially filled array. Since a partially filled array requires two arguments, the function will actually have two formal parameters: an array parameter and a formal parameter of type **int** that gives the number of array positions used. The numbers in the array will be of type double. Embed your function in a suitable test program.

Q6)

Generate a text-based histogram for a quiz given to a class of students. The quiz is graded on a scale from 0 to 5. Write a program that allows the user to enter grades for each student. As the grades are being entered, the program should count, using an array, the number of 0's, the number of 1's, the number of 2's, the number of 3's, the number of 4's, and the number of 5's. The program should be capable of handling an arbitrary number of student grades. Y ou can do this by making an array of size 6, where each array element is initialized to zero. Whenever a zero is entered, increment the value in the array at index 0. Whenever a one is entered, increment the value in the array at index 1, and so on, up to index 5 of the array. O utput the histogram count at the end. For example, if the input grades are 3, 0, 1, 3, 3, 5, 5, 4, 5, 4, then the program should output

1 grade(s) of 0

1 grade(s) of 1

0 grade(s) of 2

3 grade(s) of 3

2 grade(s) of 4

3 grade(s) of 5

A1)

```
#include <iostream>

using namespace std;

void input(int& hours24, int& minutes);

void convert(int& hours, char& AMPM);

void output(int hours, int minutes, char AMPM);

int main(){

	int hours, minutes;

	char AMPM, ans;

	do

	{

		input(hours, minutes);

		convert(hours, AMPM);

		output(hours, minutes, AMPM);

		cout << "Enter Y or y to continue, anything else quits."

		<< endl;

		cin >> ans;

	} while('Y'== ans || 'y' == ans);

	return 0;

}

void input(int& hours24, int& minutes){

	char colon;

	cout << "Enter 24 hour time in the format HH:MM "

	<< endl;
```

```
	cin >> hours24 >> colon >> minutes;
}
void convert(int& hours, char& AMPM){
	if(hours > 12) // definitely in the afternoon{
		hours = hours - 12;
		AMPM = 'P';
	}
	else if (12 == hours) // boundary afternoon hour
		AMPM = 'P'; // but hours is not changed.
	else if (0 == hours) // boundary morning hour{
		hours = hours + 12;
		AMPM = 'A';
	}
	else // (hours < 12) // definitely morning hour
		AMPM = 'A'; // hours is unchanged
}
void output(int hours, int minutes, char AMPM){
	cout << "Time in 12-hour format: " << endl
	<< hours << ":" << minutes << " "
	<< AMPM << 'M' << endl;
}
```

A2)

```
#include <iostream>
#include <cstdlib>
#include <time.h>
using namespace std;
```

```
const int NUM_GAMES = 10000;

int play_game_no_switch();

int play_game_switch();

int play_game_no_switch(){

	int prizedoor;

	int playerguess;

	prizedoor = rand() % 3;

	playerguess = rand() % 3;

	if (playerguess == prizedoor) return 1;

	return 0;

}

int play_game_switch(){

	int prizedoor;

	int playerguess;

	int revealdoor;

	int newguessdoor;

	int i;

	prizedoor = rand() % 3;

	// The player selects a door at random

	playerguess = rand() % 3;

	// The game show host reveals a door that doesn't have

	// the prize and isn't the same one guessed by the player.

	// Here we use a small loop to search for a door that satisfies

	// these conditions.

	for (i=0; i<3; i++){

		if ((i!=prizedoor) && (i!=playerguess)){
```

```
                revealdoor = i;
            }
        }
        // The player switches to the door that isn't the revealed door
        // and isn't the same one originally guessed by the player.
        for (i=0; i<3; i++){
            if ((i!=revealdoor) && (i!=playerguess)){
                newguessdoor = i;
            }
        }
        // If the new guess is the same as the prize we win, otherwise we lose
        if (newguessdoor == prizedoor) return 1;
        return 0;
}

int main(){
        // Variable declarations
        int numWinsSwitch = 0;
        int numWinsNoSwitch = 0;
        int i;

        srand(time(NULL));

        for (i=0; i<NUM_GAMES; i++){
            numWinsNoSwitch += play_game_no_switch();
            numWinsSwitch += play_game_switch();
        }
```

```
        cout << "In the simulation, when we DIDN'T switch we won " <<numWinsNoSwitch <<" times
and lost " <<

        NUM_GAMES - numWinsNoSwitch << " times, " <<" for a probability of " <<

        static_cast<double>(numWinsNoSwitch) / NUM_GAMES <<

        endl << endl;

        cout << "In the simulation, when we DID switch we won " <<

        numWinsSwitch <<" times and lost " <<

        NUM_GAMES - numWinsSwitch << " times, " <<" for a probability of " <<

        static_cast<double>(numWinsSwitch) / NUM_GAMES <<
        endl<<endl;


}
```

A3)

```
//This program tests an overloaded function to convert a pace

// in minutes and seconds to miles per hour, and also to convert

// kilometers per hour to miles per hour.

#include <iostream>

#include <cstdlib>

#include <time.h>

using namespace std;

// Function prototypes

double ConvertToMPH(int paceMin, int paceSec);

double ConvertToMPH(double kph);

// --------------------------------

// ----- ENTER YOUR CODE HERE -----

// --------------------------------

// ======================

// ConvertToMPH

// Converts a pace in minutes/seconds per mile into
```

```
// miles per hour.
// ======================
double ConvertToMPH(int paceMin, int paceSec)
{
int secspermile;
double mph;
// Convert pace to seconds per mile
secspermile = paceMin * 60 + paceSec;
// Convert to miles per hour. 1/secspermile is miles/second and then
// scale to one hour by multiplying by 3600 seconds/hour
mph = (1 / static_cast<double>(secspermile)) * 3600;
return mph;
}
// ======================
// ConvertToMPH
// Converts a pace in kilometers per mile into
// miles per hour.
// ======================
double ConvertToMPH(double kph)
{
double mph;
mph = kph / 1.61;
return mph;
}
// --------------------------------
// --------- END USER CODE --------
```

```
// --------------------------------

// ======================

// main function

// ======================

int main()

{

cout << "5:30 pace is " << ConvertToMPH(5,30) << " MPH." << endl;

cout << "7:30 pace is " << ConvertToMPH(7,30) << " MPH." << endl;

cout << "8:00 pace is " << ConvertToMPH(8,0) << " MPH." << endl;

cout << "10 kph is " << ConvertToMPH(10) << " MPH." << endl;

cout << "20 kph is " << ConvertToMPH(20) << " MPH." << endl;

cout << "120 kph is " << ConvertToMPH(120) << " MPH." << endl << endl;

}
```

A4)

```
#include <iostream>

#include <fstream>

#include <string>

using namespace std;

void getHighScore(string &name, int &highscore);

void getHighScore(string &name, int &highscore){

        string curname;

        int curscore;

        bool firstread = true;

        fstream inputStream;

        inputStream.open("input.txt");

        while (inputStream >> curname){
```

```
                inputStream >> curscore;
                // if it's the first score, set it as the high
                if (firstread){
                        firstread = false;
                        name = curname;
                        highscore = curscore;
                }
                else{
                        if (curscore > highscore){
                                name = curname;
                                highscore = curscore;
                        }
                }
        }
        inputStream.close();
}
int main( ){
        string highname;
        int highscore;
        getHighScore(highname, highscore);
        cout << "The high score is " << highscore << " by " << highname
        << endl;
        return 0;
}
```

A5)

```
#include <iostream>
#include <cmath>
```

```
using namespace std;

const int MAX_SIZE = 10;

// Pre: size <= declared size of the array argument

// Post: return the standard deviation first size array elements

// Note: The divisor is the size, not size - 1 as some

// formulae call for.

double stdDev(double s[], int size);

// Pre: size <= declared size of the array argument

// Post: return the mean of the first size array elements

double average(double s[], int size);

int main(){

	double s[MAX_SIZE];

	double stdDeviation, avg;

	char ans;

	do

	{

		cout << "Enter " << MAX_SIZE << " values, separated by\n"

		<< " white space, Terminated each with <cr>. \n"

		<< "I will compute the standard deviation.\n";

		for(int i = 0; i < MAX_SIZE; i++)

			cin >> s[i];

			stdDeviation = stdDev(s, MAX_SIZE);

			cout << "The Standard Deviation is: "

			<< stdDeviation << endl;

			cout << "Y/y continues, any other quits.\n";

			cin >> ans;
```

```
        } while(ans == 'y' || ans == 'Y');

        return 0;

}

double stdDev(double s[], int size){

        double sumSquares = 0;

        double avg = average(s, size);

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

                sumSquares += (s[i] - avg) * (s[i] - avg);

        return sqrt(sumSquares / size);

}

double average(double s[], int size){

        double sum = 0;

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

                sum += s[i];

        return sum / size;

}
```

A6)

```
#include <iostream>

#include <cstdlib>

using namespace std;

// ======================

// main function

// ======================

int main(){

        // Variable declarations

        int histogram[6];

        int score;
```

```
int i;
// Initialize histogram to 0
for (i=0; i<6; i++){
        histogram[i]=0;
}
// --------------------------------
// ----- ENTER YOUR CODE HERE -----
// --------------------------------
// Input grades until -1 is entered
cout << "Enter each grade and then -1 to stop." << endl;
do
{
        cin >> score;
        // If score in a valid range, increment the appropriate
        // entry in the histogram
        if ((score >=0) && (score <=5))
        {
        histogram[score]++;
        }
} while (score != -1);
// Output histogram
for (i=0; i<6; i++){
        cout << histogram[i] << " grade(s) of " << i << endl;
}
cout << endl;
// --------------------------------
```

```
        // --------- END USER CODE --------

        // --------------------------------

}
```