SENG1000 – C/C++ PROGRAMMING

MAJOR ASSIGNMENT 3A -THE AMAZING RACE ARRAY VARIANT

OVERVIEW

The Amazing Race is a television show in which contestants frequently have to fly from one city to another as quickly as possible, without concern for cost. Write a program that calculates flight times between cities in this scenario.

This is a difficult assignment that typically requires knowledge beyond what is normally covered in this course. You can earn a bonus (detailed below) on this assignment. Since it is assumed that you have advanced knowledge, Carlo will not provide you with assistance on how to do this assignment.

Do not attempt this unless you feel very confident that you can do it. If completing Major Assignment 3 takes you more than one hour, you definitely should not attempt this assignment.

GENERAL COURSE OBJECTIVES ADDRESSED IN THIS ASSIGNMENT

- Most objectives in the course apply to this assignment, as it is quite advanced.

ACADEMIC INTEGRITY AND LATE PENALTIES

- Link to Academic Integrity Information
- Link to Late Policy

EVALUATION

- The evaluation of this assignment will be done as detailed in the Marking lecture from Week 2.

PREPARATION

- You must have **significant** programming experience. This is a very difficult assignment.

REQUIREMENTS

MAIN OBJECTIVE

- Write a program that will find the best route (i.e. least amount of time from starting time specified by the user to final arrival) from one city to another. If you successfully do this, you could get a maximum mark of 200%.
- If you cannot write a program that finds the best route, you can write one that finds any route. If you successfully do this, you could get a maximum mark of 120%.
- If you submit this assignment and you do it poorly, you would probably be better off handing in Major Assignment 3 instead.
- If you submit both Major Assignment 3 and 3a, Major 3a will be marked and Major 3 will be ignored.

FLIGHT SCHEDULES

The flight schedules are as follows:

- Departing Toronto (city 1):

Destination	Departure	Flying Time	Departure	Flying Time	Departure	Flying Time
Atlanta	6:25a	2:20	9:10a	4:50	12:30p	4:15
	4:10p	6:10	8:00p	2:15		
Denver	7:30a	3:35	3:00p	6:00		
Chicago	6:40a	1:20	every 60 minutes from 7:40a to 2:40p	1:35	every 60 minutes from 3:30p to 7:30p	1:45
	9:00p	1:30	10:00p	1:15		

  - What this means is that there are five flights leaving Toronto for Atlanta, leaving at 6:25 a.m., 9:10 a.m., 12:30 p.m., 4:10 p.m., and 8:00 p.m. They have radically different lengths, since they might stop in other unknown cities as well. The shortest flight it the 8 p.m. flight, taking 2 hours and 15 minutes. The longest flight is the 4:10 p.m. flight, taking 6 hours and 10 minutes.

- Departing Atlanta (city 2):

Destination	Departure	Flying Time	Departure	Flying Time	Departure	Flying Time
Toronto	7:10a	2:10	10:30a	4:10	3:00p	3:50
	5:10p	6:10	9:00p	2:20		
Austin	9:00a	2:10	3:30p	2:50	8:00p	2:30
Denver	6:00a	3:00	1:20p	5:00	5:10p	2:50
Chicago	6:50a	2:10	every 60 minutes from 7:50a to 2:50p	3:00	every 60 minutes from 3:50p to 7:50p	2:30
	8:30p	2:10				

- Departing Austin (city 3):

Destination	Departure	Flying Time	Departure	Flying Time	Departure	Flying Time
Atlanta	9:10a	2:20	3:00p	2:20	9:30p	2:30
Denver	10:30a	2:20	6:20p	2:20		
Santa Fe	5:00p	0:55				

- Departing Santa Fe (city 4):

Destination	Departure	Flying Time	Departure	Flying Time	Departure	Flying Time
Austin	3:00p	0:45				

- Departing Denver (city 5):

Destination	Departure	Flying Time	Departure	Flying Time	Departure	Flying Time
Toronto	6:30a	4:10	10:30a	5:20	2:00p	5:00

Atlanta	6:00a	3:10	1:00p	3:20	3:00p	3:50
Austin	12:00p	2:00	3:00p	2:20		
Chicago	7:00a	2:20	every two hours from 8:00a to 4:00p	2:50	6:30p	2:40

- Departing Chicago (city 6):

Destination	**Departure**	**Flying Time**	**Departure**	**Flying Time**	**Departure**	**Flying Time**
Toronto	7:40a	1:10	every 60 minutes from 9:10a to 5:10p	2:30	7:10p	2:00
	9:10p	2:10				
Atlanta	6:50a	2:10	every 60 minutes from 8:00a to 8:00p	2:40	9:50p	3:00
Denver	9:00a	2:10	every two hours from 11:30a to 5:30p	2:20	9:00p	2:50
Buffalo	11:00a	2:00	1:10p	1:50	3:00p	2:30
	6:00p	2:10				

- Departing Buffalo (city 7):

Destination	**Departure**	**Flying Time**	**Departure**	**Flying Time**	**Departure**	**Flying Time**
Chicago	9:40a	1:40	11:10a	1:50	5:50p	2:30
	8:10p	2:20				

USER INPUT:

- The user must provide the starting and ending cities as a number. The numbers corresponding to each city are shown above. Do not change the numbers for the cities.
- The user will also specify a starting time. The time must be entered in 24 hour time without a colon (e.g. "412", "1120", "1500", "2359").
    - The user will always enter a valid time (i.e. I won't try to mess you up by testing with a time of 4399).
- When you get input from the user, the order must be starting city, ending city, starting time, **all on one line** separated by spaces.
- As always, prompt the user first.
- It is recommended, but not required, that you use getNum() from Major Assignment 2.
    - Do not use scanf().

- You are not responsible for handling excessively long input lines (users will not enter more than 40 characters) or numbers out of the range of an int variable.
- Do not have extra user input (including "Press any key to continue") beyond getting the above information repeatedly.
- Quit the program only upon receiving invalid input.

OUTPUT

- The result must be calculated and displayed (**in correct hh:mm format**; not hours only or hours separated from minutes by a space) when appropriate. The result must include all layovers between cities.
    - The calculation includes waiting in the first airport for your first plane.
- When printing the output, list each leg of the plan (with times and timezones included).
- Do not clear the screen. It is appropriate to display blank lines at the end of your loop to space the output out.
- Spelling and consistency in your output is important. Also, make good use of blank lines to separate unrelated parts of your output.
- Sample Output (not necessarily the fastest route):

```
Flying from Chicago to Austin.
Starting from Chicago at 4:02 p.m. CST
Leaving Chicago at 5:00 p.m. CST for Atlanta.
Arriving in Atlanta at 8:40 p.m EST.
Leaving Atlanta at 9:00 a.m. EST next day for Austin.
Arriving in Austin at 10:10 a.m CST.
Total travel time: 18:08
```

COMMENTING AND STYLE:

- You must have a header comment similar to that found in the SET Coding Standards or the Course Notes.
- You must have function comments for all functions except main().
- You must have inline comments that adhere to the two main principles for good inline comments (covered in the Commenting lecture and Code Complete).
- Adhere to the principles covered in the Style lecture.
- The above points will apply identically for subsequent assignments as well.

OTHER REQUIREMENTS:

- You must take time zones into account. It is your job to find out what the time zones are for each city. Assume standard time, not daylight time (this will only affect what you display as the time zone in the output). Cities are in the Eastern, Central, and Mountain time zones (EST, CST, and MST, respectively).
- If you wish to have your data in a separate .cpp or .h file, you can do so. You can name that file whatever you want.
- Use the SET Coding Standards (found on eConestoga) that are relevant.
- You must use good structured design principles to modularize your code into functions. Having most of your code in one function is very bad design and will pretty much guarantee that you won't get the bonus.

- The program must not use goto (this requirement holds for all subsequent assignments).
- All variables must be declared within functions (i.e. it must not use global variables (covered in the Scope and Style lecture)).
- As usual, constants can be created/declared globally.
- Appropriate programming style as discussed in lecture and in the Course Notes must be used.
- Do not have any input or output except as required by this assignment.
- It is assumed that you will adhere to all course requirements detailed in the Course Notes readings so far in the course. **This requirement holds for all subsequent assignments.**
- Do not clear the screen (this requirement is true for all assignments).

CHECKLIST REQUIREMENTS

- Create a requirements checklist. This should contain the specific requirements from this assignment as well as any relevant requirements that have been covered in lecture or that are found in the SET Coding Standards or SET Submission Standards. Do it in whatever form you wish. Hand in your completed checklist in PDF form as checklist.pdf. Not having this checklist will result in a cap of 80 on your mark.

FILE NAMING REQUIREMENTS

- You must call your source file m3a.cpp.
- You must call your checklist checklist.pdf.

SUBMISSION REQUIREMENTS

- Do not hand in any other source files besides those mentioned in the File Naming Requirements.
- Follow the instructions in the SET Submission Standards and the lecture on Submitting Assignments to submit your program. Submit both files in one submission to the correct Assignment folder.
- Once you have submitted your files, make sure that you've received the eConestoga e-mail confirming your submission. Do not submit that e-mail (simply keep it for your own records until you get your mark).

ADDITIONAL INFORMATION

- You can assume (unrealistically) that if you land in a city at exactly the same time that your connecting flight leaves, you can immediately get on that connecting flight (let's assume that the airline will hold the plane for you (you're such a special person!)).