

## ECE 71 – Engineering Computations in C

Professor Kriehn – Fall 2016

**Code Due By:** Midnight on Tuesday, Nov 08, 2016

**Writeup Due By:** Class on Thursday, Nov 10, 2016

### HOMEWORK #24 – Fun with Pointers

Write a program that prompts the user to input 3 integer values into the computer. Write a function that re-orders the values in the three integer variables such that the values occur in ascending order.

#### Specifications:

Use the following function prototype for your sorting function:

```
void ptr_sort(int *a, int *b, int *c);
```

Use the following three variables to store your numbers in the `main()` function:

```
int x, y, z;
```

Use the following three pointers to point to the three variables in the `main()` function:

```
int *i, *j, *k;
```

After you have your integer variables and pointers declared, prompt the user to enter 3 numbers (1 number at a time). Scan in the numbers into the variables using your pointers, and call your `ptr_sort()` function – whose purpose is to sort the numbers so that `x` contains the smallest number, `y` contains the middle number, and `z` contains the largest number. Then print the results to the screen. You **MUST** use pointers to do ALL operations relating to scanning, sorting, and printing the numbers.

If you execute the program, the following information should be displayed:

```
~> hw23.o
Please enter the first number: 3
Please enter the second number: 1
Please enter the third number: 2
The sorted numbers are: 1, 2, and 3.
~> hw23.o
Please enter the first number: -15
Please enter the second number: -30
Please enter the third number: 212
The sorted numbers are: -30, -15, and 212.
~>
```

## HOMEWORK #25 – Airline Departure and Arrival Times

The following table shows the daily flights from one city to another:

Departure Time	Arrival Time
8:00 am	10:16 am
9:43 am	11:52 am
11:19 am	1:31 pm
12:47 pm	3:00 pm
2:00 pm	4:08 pm
3:45 pm	5:55 pm
7:00 pm	9:20 pm
9:45 pm	11:58 pm

Write a program that asks the user to enter a time (expressed in hours and minutes, using the 24-hour clock). The program then displays the departure and arrival times for the flight whose departure time is closest to that entered by the user, using standard time.

### Specifications:

The program should prompt the user to enter a time and pattern match against the ‘:’ character. After reading in the hours and minutes, calculate the time in minutes past midnight, and pass the time into a function that finds the flight whose departure time is closest to the `desired_time` (expressed in minutes since midnight). The function will store the departure and arrival times of the flight (also expressed in minutes since midnight) in variables pointed to by `departure_time` and `arrival_time`, respectively. Use the following function prototype:

```
void find_closest_flight(int desired_time, int *departure_time,
int *arrival_time);
```

Inside the function, define two constant integer arrays called `departures` and `arrivals` that store the departure times and arrival times of the airplane in minutes since midnight. Use a global constant to help convert the departure and arrival times to minutes past midnight. Use a `while()` loop and `if-else` statement to determine which element of the array contains the departure time closest to the `desired_time`. Then assign the appropriate departure and arrival times to the variables pointed to by the pointers `departure_time` and `arrival_time`, and exit the function.

Once the `find_closest_flight()` function returns back to the main function, use the information stored in the variables that were pointed to by `departure_time` and `arrival_time` in the function to back out the actual departure and arrival times. Since the times are in minutes past midnight, you must pull out the hour and minutes, convert from 24 time to standard time, and print out the appropriate results – with the appropriate number of digits.

For example, 13:15 is  $13 \times 60 + 15 = 795$  minutes since midnight, which is closer to 12:47 pm (767 minutes since midnight) than to any of the other departure times.

~> hw25.o

Enter a 24-hour time: 13:15

Closest Departure/Arrival Times

Departure Time: 12:47 pm

Arrival Time: 3:00 pm

~>

~> hw25.o

Enter a 24-hour time: 23:59

Departure Time: 9:45 pm

Arrival Time: 11:58 pm

~>

~> hw25.o

Enter a 24-hour time: 3:12

Departure Time: 8:00 am

Arrival Time: 10:16 am

~>