Course: Programming Fundamentals – ENSF 337

Lab #: Lab 2

Instructor: M. Moussavi

Student Name: Derek Braun, 30040032

Lab Section: B01

Date Submitted: Sept. 26, 2018

Exercise D: More on scanf

Run #	Your Inputs	What is the value of n	What is the value of i	What is the value of d
1	12, 0.56	2	12	0.560000
2	5.12, 9.56	2	5	0.120000
3	12, ab	1	12	1234.500000
4	ab, 12	0	333	1234.500000
5	5ab, 9.56	1	5	1234.500000
6	13, 67	2	13	67.000000

Exercise E: Simple functions to do input error handling

```
/*
 * lab2exe_E.c
 * ENSF 337 - Lab 2 - Execise E
 * Completed By: Derek Braun, 30040032, B01
 *
 *  Functions to read numbers with input error detection.
 *
 *
 */

#include <stdio.h>
#include <stdlib.h>

int read_int(void);
/*
 *  REQUIRES
 *      User has been prompted to type in an int.
 *  PROMISES
 *      If an int can be read from the standard input stream, that
 *      int is the return value.  If not, an error message is printed
 *      and exit is called with an argument of 1.
 */

double read_double(void);
/*
 *  REQUIRES
 *      User has been prompted to type in a double.
 *  PROMISES
 *      If a double can be read from the standard input stream, that
 *      double is the return value.  If not, an error message is printed
 *      and 'exit' is called with an argument of 1.
 */
```

```
int main(void)
{
  int the_int;
  double the_double;

  printf("Doing tests of read_int and read_double functions ...\n\n");

  printf("Please try to enter an int: ");
  the_int = read_int();

  printf("read_int succeeded.  It read a value of %d.\n", the_int);

  printf("Please try to enter a double: ");
  the_double = read_double();

  printf("read_double succeeded.  It read a value of %g.\n", the_double);


  return 0;
}


int read_int(void)
{
        int pass;
        scanf("%d", &pass);
        int i = fgetc(stdin);

        while(i == 32){
                i = fgetc(stdin);
        }

        if(i != '\n'){
                printf("Unaccepted Entry\n");
                exit(1);
        }
        return pass;
}

double read_double(void)
{
        double pass;
        scanf("%lf", &pass);
        int i = fgetc(stdin);

        while(i == 32){
                i = fgetc(stdin);
        }

        if(i != '\n'){
                printf("Unaccepted Entry\n");
                exit(1);
        }
```

```
	return pass;
}
```

Output:

Test Case 1:

```
Doing tests of read_int and read_double functions ...

Please try to enter an int: -75
read_int succeeded.  It read a value of -75.
Please try to enter a double: 267.96
read_double succeeded.  It read a value of 267.96.
```

Test Case 2:

```
Doing tests of read_int and read_double functions ...

Please try to enter an int: 23.44
Unaccepted Entry
```

Test Case 3:

```
Doing tests of read_int and read_double functions ...

Please try to enter an int: 1
read_int succeeded.  It read a value of 1.
Please try to enter a double: 23.4a
Unaccepted Entry
```

Exercise F: Writing a Complete C Program with a Few Functions

```
/*
Completed By: Derek Braun, 30040032, B01

takes in input of distance and speed of a travelling vehicle.
prints the distance traveled, average speed and total time elapsed.
*/

#include <stdio.h>
#include <math.h>

void get_user_input(double *distance, double *speed);
void travel_time_hours_and_minutes(double distance, double speed, double *hours, double *minutes);
void display_info(double distance, double speed, double hour, double minutes);

int main(void){
	double dist_in;
```

```
	double speed_in;
	double total_hours;
	double total_minutes;

	get_user_input(&dist_in, &speed_in);
	travel_time_hours_and_minutes(dist_in, speed_in, &total_hours, &total_minutes);
	display_info(dist_in, speed_in, total_hours, total_minutes);

	return 0;
}

void get_user_input(double *distance, double *speed){
	printf("Please enter the travel distance in km: ");
	scanf("%lf", distance);
	printf("\nNow enter the vehicles, average speed (km/h): ");
	scanf("%lf", speed);
}

void travel_time_hours_and_minutes(double distance, double speed, double *hours, double *minutes){
	double intermediate_hours;
	intermediate_hours = 1/(speed/distance);

	*hours = floor(intermediate_hours);
	*minutes = 60*(intermediate_hours-floor(intermediate_hours));
}

void display_info(double distance, double speed, double hour, double minutes){
	printf("\nYou have traveled %.2lf (km) with a speed of %.2lf in %.2lf hour(s) and %.2lf
minutes(s)", distance, speed, hour, minutes);
}
```

Output:

Test Case 1:

Please enter the travel distance in km: 125.5

Now enter the vehicles, average speed (km/h): 65.5

You have traveled 125.50 (km) with a speed of 65.50 in 1.00 hour(s) and 54.96 minutes(s)

Test Case 2:

Please enter the travel distance in km: 5.44

Now enter the vehicles, average speed (km/h): 76.5

You have traveled 5.44 (km) with a speed of 76.50 in 0.00 hour(s) and 4.27 minutes(s)