

1310 INLAB 05

Note: Use **mkdir** command to create a new directory by name “lab5”. All the files for inlab 5 should be saved in this directory only. Also you need to save the source code for each checkpoint in a separate file. Names of the files should follow the format ‘timechk1.c’, ‘time-chk2.c’ etc. After you complete a checkpoint, make a copy of that file for the next checkpoint and modify the copied file.

You will start with a simple C program. You need to connect to ‘omega.uta.edu’ server for all C programs and use ‘vi’ or ‘pico’ editor for writing the programs.

At the command prompt, type the following:

vi time-chk1.c

or

pico time-chk1.c

```
/* time-chk1.c – There are four cities and 3 distance values. Distance 1 is distance between city 1 and city 2, distance 2 is distance between city 2 and city 3, and distance 3 is distance between city 3 and city 4. This program reads the 3 distance values and the average speed and calculates time to travel these distances. */
```

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    double dist1, dist2, dist3, /* distance values (in meters) */
           avgsp, /* variable to accept average speed */
           time1, time2, time3; /* variables to hold travel times */
```

```
    printf(“Enter distance between city 1 and city 2: ”);
    scanf(“%lf”, &dist1);
```

```
    printf(“Enter distance between city 2 and city 3: ”);
    scanf(“%lf”, &dist2);
```

```
    printf(“Enter distance between city 3 and city 4: ”);
    scanf(“%lf”, &dist3);
```

```
    printf(“Enter average speed: ”);
    scanf(“%lf”, &avgsp);
```

```
    time1 = dist1/avgsp;
    printf(“Time to travel from city 1 to city 2 is %4.2f\n”, time1);
```

```
    time2 = dist2/avgsp;
    printf(“Time to travel from city 2 to city 3 is %4.2f\n”, time2);
```

```
    time3 = dist3/avgsp;
    printf(“Time to travel from city 3 to city 4 is %4.2f\n”, time3);
```

```
    return 0;
```

```
}
```

Save the program and exit 'vi' or 'pico'. Answer the following questions:

1. What is the output of this program?
2. What are the inputs to this program?

Write answers to above questions on a sheet of paper. After this, compile the program using the command:

```
gcc time-chk1.c
```

Now execute the program using:

```
a.out
```

If the program runs properly, signal the grader.

Checkpoint 1

In the program that you just compiled, make the following changes to the printf() function used to output the values of time1, time2, and time3:

- While printing time1, change the placeholder to %13.3 from %4.2f
- While printing time2, change the placeholder to %10.2c from %4.2f
- While printing time3, change the placeholder to %8.3d from %4.2f

Compile and execute the program and observe the changes in the output. If there are any errors either during the compilation or runtime, correct them (hint: place holders) to get the output values in a required format. Signal the grader to show your results and explain the reason for errors, if any.

Checkpoint 2

Remove all the occurrences of time1, time2, and time3 from the above program. At the beginning of main() function declare two variable 'start_time' and 'cumulative_time' of type double. Put a printf() and scanf() statement to read the value of variable 'start_time'. Replace the calculation for time1 with the following one –

```
cumulative_time = start_time + dist1/avgsp;
```

Similarly, replace the calculation for time2 and time3 with the statements to calculate the value of cumulative_time such that the new travel time is added to the existing value of cumulative_time.

Print the value of cumulative_time by using %f in the printf() statement at the end of above calculation. Save the program and compile it. If it compiles without any errors, run it. Signal the grader when you get the right output.

Checkpoint 3

At the command prompt, type the following:

```
vi grades-chk4.c
```

or

```
pico grades-chk4.c
```

```
/* grades.c: This program calculates the grade of a student in five subject tests*/
```

```
#include <stdio.h>
```

```
int main(void )
```

```

{
    double score1, score2, score3, score4, score5, average;
    printf("Enter the scores in 5 subject tests: ");
    scanf("%d %d %d %d %d", &score1, &score2, &score3, &score4, &score5);
    if( (score1 = 0) && (score2 = 0) && (score3 = 0) && (score4 = 0)&& (score5 = 0))
    {
        printf("Invalid input value. Exiting program\n");
        return 0;
    }
    average = (score1+score2+score3+score4+score5)/5;
} /* main */

```

Some logical errors are deliberately introduced in this program. You need to find those errors and correct them before you try to compile and execute the program. Here is the hint for correcting the errors –

- Placeholders in the scanf statement
- Comparison operator in the ‘if’ statement

After you fix above two errors, save and compile the program and signal the grader if it works fine.

Checkpoint 4

Add the following lines of code at the end of the above program:

```

    if (average >= 90 && average <100)
        printf("Grade is A");
    if (average >= 80 && average <90)
        printf("Grade is B");

```

When this is done, save and exit from ‘vi’ or ‘pico’. Compile the program again and make sure it is error-free. Signal the grader when you’ve finished this checkpoint.

Checkpoint 5

Complete the above program using the information given below:

- A student gets a grade of ‘C’ if his average is greater than or equal to 70 and less than 80
- A student gets a grade of ‘D’ if his average is greater than or equal to 60 and less than 70
- A student gets a grade of ‘F’ otherwise.

Complete this, and then save and exit the ‘vi’ editor, compile the program using ‘gcc’. If the program compiles without any errors, execute the program using the ‘a.out’ executable to get the output. If the program has compilation errors, you need to fix them before executing the program. Signal the grader when you have finished.

Checkpoint 6

Add the code in the above program to calculate the final grade. The final grade will depend on the average of scores in five subjects and the project score. The information for doing this is provided below:

You need to declare a variable named 'project' to store the value of project score and another variable named 'final_grade' to store the final grade. Add printf() and scanf() statement to read the project score. After you find out the average of five test scores, write the code to calculate the final_grade such that average of subject scores and the project score will each count 50% of the final_grade.

You also need to modify all the 'if' statements, which print the grade of the student. Use 'final_grade' instead of 'average' in these 'if' statements as given below:

Grade	final_grade
A	≥ 90
B	80 - 89
C	70 - 79
D	60 - 69
F	<60

When you've completed this, compile and execute the program as before and with the output on the screen, signal the grader to check your work.

Checkpoint 7

Include a program header at the top of your program. The header should contain the following:

1. Your name
2. Your lab instructor's name
3. Your lab section
4. Duration of the lab
5. Lab due date
6. Brief description of object of the lab

You need to encapsulate the header within the /* and */ comment marks. Save the file and exit 'vi' or 'pico'. Compile the program. If it compiles without error, signal the grader.

Checkpoint 8