Introduction to C - Programming Assignment #1

Due date: *September 9, 2011 – 11:59pm*

Objectives

1. To give students practice at typing in, compiling and running simple programs.
2. To learn how to read in input from the user.
3. To learn how to use assignment statements and arithmetic expressions to make calculations

Introduction: Arup's Getting Married!!!

Arup got engaged this summer and now has been inundated with much more work than he anticipated in planning his wedding. He is in a bind and has decided that some C Programming skills could actually help him. In the following assignment, you'll write three programs to help him make some decisions for his wedding planning. Hopefully, your programs will save him some time so that he can spend enough time teaching classes and still have time leftover for some UCF football!!!

Problem A: Rectangular Wedding Table Arrangements (table.c)

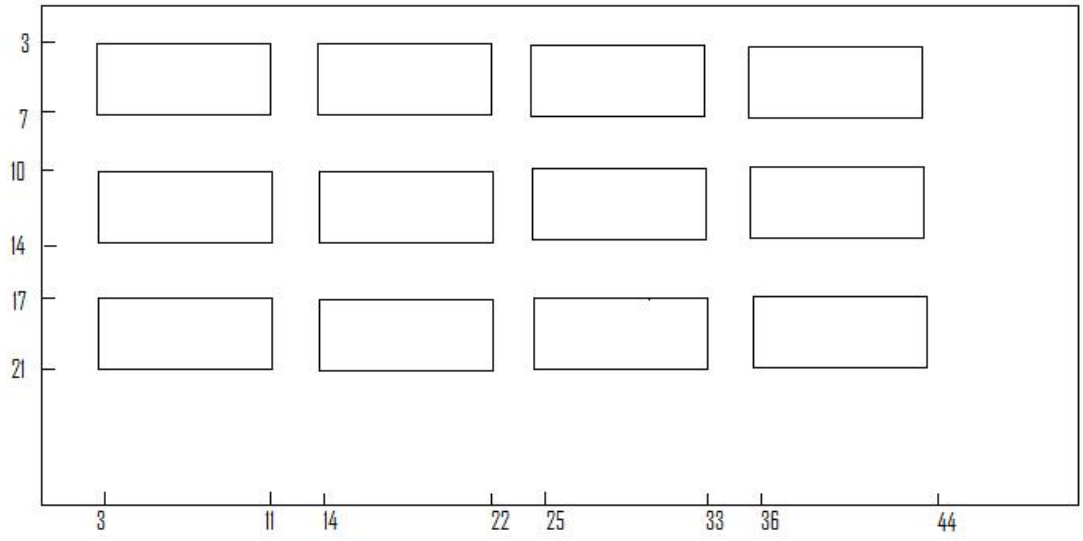
One of the first jobs everyone has when planning a wedding is picking a venue for the reception. In doing so, many venues show their seating arrangements for dinner to their clients. Depending on the dimensions of the tables and the distance between tables affects the number of people that can be seated. In this problem, you'll analyze placing rectangular tables in a rectangular room. Given the relevant dimensions, your job will be to determine the maximum number of people that can be seated in the room.

First, we assume that the room dimensions, length by width, are given, in feet. Next, we assume that each table is identical in size, and those dimensions are also provided in feet. We are also given the number of feet of space required between each table and between each table and wall. The last piece of information we need is the number of people that can be seated at each table.

Consider the following situation:

Room dimensions: 50' x 30'
Table dimensions: 8' x 4'
Space required: 3'
People per table: 10

The picture is given on the following page.



The labels on the left side of the diagram represent how far from the "top" of the room the beginning and end of each table is. Note that after the label 21, if we were to place another table, we'd have to skip three feet to 24 and place the table, which would end at 28 feet. But the problem with this is that we'd only have two feet remaining from the end of this fourth table to the bottom wall.

The labels on the bottom of the diagram represent how far from the left wall the beginning and ending of each table is. It should be fairly clear that a fifth table will not fit at the right end of the room.

Thus, for this given situation, 120 people (10 people at each of the 12 tables) can be seated for dinner.

Input Specification

1. The room dimensions (in feet) will be positive integers in feet less than 1000.
2. The table dimensions (in feet) will be positive integers in feet less than 1000.
3. The space required between tables and walls will be a positive integer less than 10.
4. The people per table will be a positive integer in between 3 and 20, inclusive.

Output Specification

Output the total number of people the given seating arrangement can support with a statement in the following format.

```
This arrangement seats X people.
```

where X is the number of people is question.

Output Sample

Below is one sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

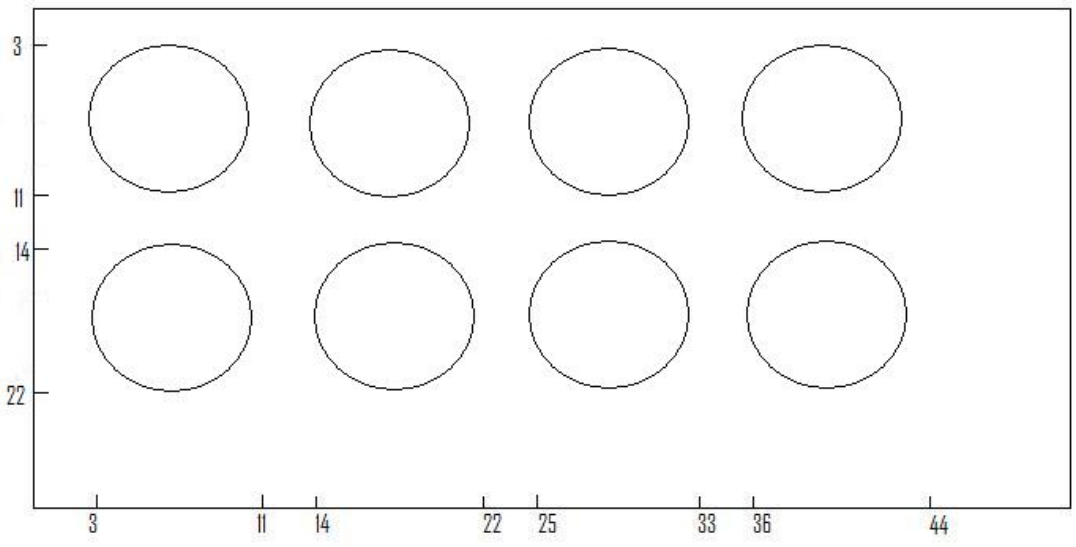
Sample Run #1

```
What are the length and width of the room (in feet)?
50 30
What are the length and width of each table (in feet)?
8 4
How much space is required between tables (in feet)?
3
How many people does each table seat?
10
This arrangement seats 120 people.
```

Problem B: Circular Table Arrangement (circle.c)

Unfortunately for Arup, his finacee thinks that rectangular tables are not attractive. She prefers circular tables. For this portion of the assignment, you must make your calculations based on circular tables. The only difference in the input is that instead of receiving a length and width for each table, you will only receive the radius of each table, in feet.

Here is an example for the same sized room, spacing and number of people per table as the example in part A with tables of radius 4:



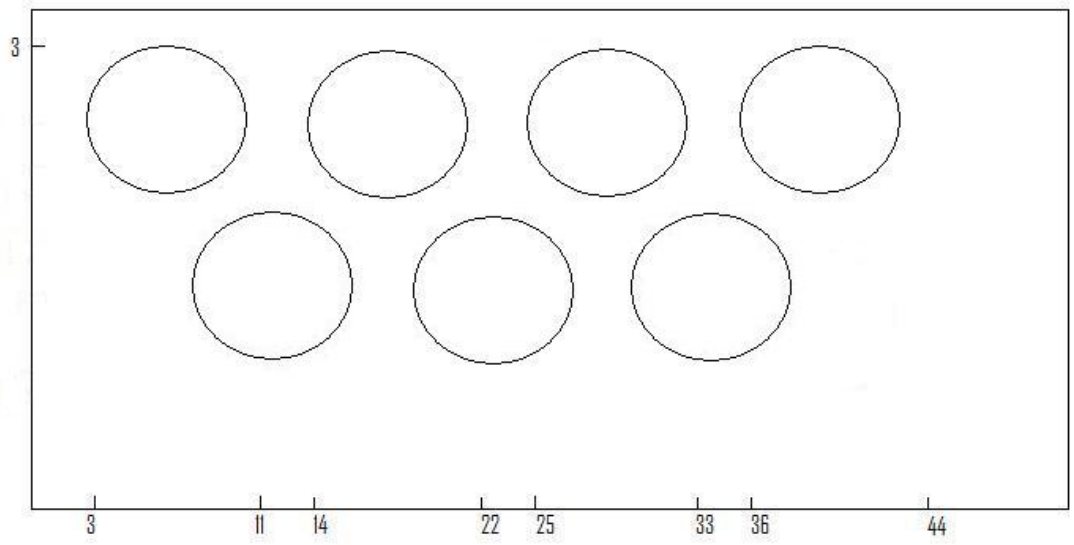
Input Specification
Same as problem A except that the radius will be a positive integer in between 1 and 500.

Output Specification
Same output specification as part A.

Output Samples
A sample of the program running is included below. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above.

Sample Run
```
What are the length and width of the room (in feet)?
50 30
What is the radius of each table (in feet)?
4
How much space is required between tables (in feet)?
3
How many people does each table seat?
10
This arrangement seats 80 people.
```

A Bit of Extra Credit (circle2.c)
A slightly better way (sometimes) to arrange circular tables is to "offset" each consecutive row in a "checkerboard" pattern as follows:



To form this design, note that the centers of two adjacent circles in the top row and the corresponding circle below it on the next row for an equilateral triangle. This design in

certain cases allows for more rows than the original circular design posed, which then allows for more people.

Rewrite your program to calculate the number of people that can be seated using this type of circular table arrangement.

Problem C: What's For Dinner (food.c)

At the dinner of the wedding reception, Arup and his fiancée have decided that they will offer guest the choice between chicken and steak for their entrees. Naturally, the chicken costs less than the steak, so Arup's secret desire is that everyone orders chicken so that he can pocket the extra money to buy some Dolphins tickets.

Your job will be to write a program that helps him figure out the minimum number of guests who have to order chicken for the total meal cost to be below a particular threshold.

For example, if the chicken costs $40/person, the steak costs $50/person, there are 200 guests, and the designated target value was $9059, then at least 95 people must order the chicken. To see this, note that if 95 people order the chicken, the total cost is:

95 x $40 + 105 x $50 = $9050, which is under the target.

If only 94 people order the chicken, the cost is:

94 x $40 + 106 x $50 = $9060, which exceeds our target of $9059.

Implementation Requirements

Loops have not yet been taught in the course. Do NOT use a loop to solve this problem. Doing so will result in the automatic loss of 20 points.

Input Specification

1. The cost of the chicken (in dollars) will be a positive integer less than 100.
2. The cost of the steak (in dollars) will be a positive integer greater than the cost of the chicken and less than 100.
3. The number of guest will be a positive integer greater than 10 and less than 1000.
4. The designated target value (in dollars) will be an integer greater than the cost of all guests ordering chicken and less than the cost of all guests ordering beef.

Output Specification

Output the minimum number of guests who much order chicken for the total cost of the meals to be less than or equal to the designated target.

```
You need at least X guests to order the chicken.
```

where X is the number of guests in question.

Output Samples

A sample output of running the program is included below. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above.

Sample Run #1

```
What is the cost of the chicken, in dollars?
40
What is the cost of the beef, in dollars?
50
How many guests will there be?
200
What is your spending limit for food, in dollars?
9059

You need at least 95 guests to order the chicken.
```

Deliverables

Three source files:

1) *table.c*, for your solution to problem A
2) *circle.c* for your solution to problem B
3) *food.c* for your solution to problem C

All files are to be submitted over WebCourses.

Restrictions

Although you may use other compilers, your program must compile and run using Code::Blocks. Each of your three programs should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

Grading Details

Your programs will be graded upon the following criteria:

1) Your correctness

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.

3) Compatibility to Code::Blocks. If your program does not compile in this environment, you will get a **sizable** deduction from your grade.