**CS 200: Introduction to Programming**

Dr. Naveed Arshad

Muhammad Kamran Nishat

Assignment 1  
(Due Date: Tuesday, September 23, 2014)



This assignment is due on Tuesday, 23th September. The usual late submission policy of 10% per day deduction for up to 3 days applies. The assignment needs to be submitted on LMS under the ‘Assignments’ tab.

The course policy about plagiarism is as follows:

1. Students must not share actual program code with other students.
2. Students must be prepared to explain any program code they submit.
3. Students cannot copy code from the Internet.
4. Students must indicate with their submission any assistance they received.
5. All submissions are subject to automated plagiarism detection.

Students are strongly advised that any act of plagiarism will be reported to the Disciplinary Committee.

**1. Expense Calculator: [20points]**

Write a program to calculate the total expenses. It should take Quantity and price per item as input by the user. It should ask user if he/she wants to enter more items, stops when they type 0 or continue if they type 1. After every item it should display total (discount of 10% is offered if the expense is more than 5000).

**2. MaxMin: [20points]**

Write a program which takes sequence of integers as inputs until user input number ***-1.***

In the end prints Maximum and Minimum of the numbers inputted by user.

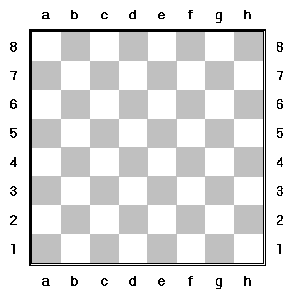
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The Bitwise operators supported by C language are listed in the following table. Assume variable A holds 60 (111100) and variable B holds 13 (001101), then:   |  |  |  | | --- | --- | --- | | **Operator** | **Description** | **Example** | | & | Binary AND Operator copies a bit to the result if it exists in both operands. | (A & B) will give 12 which is 0000 1100 | | | | Binary OR Operator copies a bit if it exists in either operand. | (A | B) will give 61 which is 0011 1101 | | ^ | Binary XOR Operator copies the bit if it is set in one operand but not both. | (A ^ B) will give 49 which is 0011 0001 | | ~ | Binary Ones Complement Operator is unary and has the effect of 'flipping' bits. | (~A ) will give -61 which is 1100 0011 in 2's complement form due to a signed binary number. | | << | Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand. | A << 2 will give 240 which is 1111 0000 | | >> | Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand. | A >> 2 will give 15 which is 0000 1111 | |

**3. Sierpinski Triangle: [60points]**

|  |  |
| --- | --- |
| Write a program to print a portion of the **Sierpinski triangle**, as follows. You will print a square grid using two different characters, perhaps an asterisk and a blank space. For example, the 8 × 8 grid would then look like the figure at the right. | \*\*\*\*\*\*\*\*  \* \* \* \*  \*\* \*\*  \* \*  \*\*\*\*  \* \*  \*\*  \* |

Suppose the columns and rows are numbered starting at zero; in the example they are numbered zero through seven. At row number ***r*** and column number ***c,*** compute the “bitwise-and” of the numbers using ***r&c***. If this value is zero print one character, say the asterisk, and otherwise print the other character. Your program should request the size of the grid, check that the supplied number is larger than zero, and print the grid.

BONUS PART



**4. Chess Moves: [100points]**

Ask user for the type of piece, **P**awns, **K**ing, **Q**ueen, **R**ooks, **B**ishops, **K**nights (e.g. P, K, Q or B). Then take its initial position as an input (e.g. e1 or f7). In the last ask for final position of the piece as input (e.g. e4 or g7). Using the chess rules determine if this move is legal for that selected piece or not.

(Hint: Use a pair of variables one of type ***char*** and other of type ***int*** each for initial and final position.)