1. (5 points) Find out all the mistakes in the following program, explain why they are wrong.

```c
#include <stdio.h>

int main() {
    int a, b;
    printf("Type in an integer:");
    scanf("%d", &a);
    printf("Type in an integer:");
    scanf("%d", &b);
    c = a % b;  // variable c is not defined.
    if(c = 0);  // it should be c == 0, and there should not be any ;
            // printf("%d is divisible by %d\n", a, b);
    else
            // printf("%d is not divisible by %d\n", a, b);
}
```

CSE 220: HOMEWORK 5

Due is on Sunday 11:59 PM, October 18, 2015
2. (5 points) Based on the operator precedence and associativity, give an equivalent fully parenthesized expression for the following expression:

\[ a = (b /= (((++c) \% (d--)) - e + ((f++) * g))) \]

3. (5 points) Give the output of the following program:

```c
#include <stdio.h>

int main(void) {
    int i, j, k;
    i = 3; j = 4; k = 5;
    printf("%d\n", i - 3 > 0 || ++i == j || j++ < k);
    printf ("%d %d %d\n", i, j, k);
}
```

output:
1
4 4 5

4. (10 points) The octal numeral system is the base-8 number system, and uses the digits 0 to 7. Octal numerals can be made from binary numerals by grouping consecutive binary digits into groups of three (starting from the right). For example, the binary representation for decimal 74 is 1001010, which can be grouped into (00)1 001 010 – so the octal representation is 112:

<table>
<thead>
<tr>
<th>Binary</th>
<th>Octal</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(1) Convert hexadecimal value 2A to octal representation. \(52_8\)
(2) Convert decimal value 37 to octal representation. \(45_8\)
(Note: you should not only give the final answer, but also show the details of each step)
5. (25 points) Complete the following program, such that: if the user types in a lowercase letter, convert it to uppercase and print out; if the user types in an uppercase letter, convert it to lowercase and print out; print out “It is not an alphabetic character” if the user’s input is neither an uppercase nor a lowercase letter. Here is the ASCII code table:

<table>
<thead>
<tr>
<th>int</th>
<th>char</th>
<th>int</th>
<th>char</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>0</td>
<td>90</td>
<td>Z</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>...</td>
<td>a</td>
</tr>
<tr>
<td>57</td>
<td>9</td>
<td>98</td>
<td>b</td>
</tr>
<tr>
<td>65</td>
<td>A</td>
<td>122</td>
<td>z</td>
</tr>
</tbody>
</table>

#include <stdio.h>

int main() {
    char ch;

    scanf("%c", &ch);

    /* Your code goes here */

    if (ch >= 'A' && ch <= 'Z'){
        ch += 32; // to lowercase
        printf("%c",ch);
    }
    else if (ch >= 'a' && ch <= 'z'){
        ch -= 32; // to uppercase
        printf("%c",ch);
    }
    else {
        printf("It is not an alphabetic character");
    }

    return 0; }

6. Write a c program that gets an integer input 'n' and prints all the even integers between 1 and n. (hint: you can use 'for' loop)

```c
#include <stdio.h>

int main() {
    int n;

    printf("Enter an integer: ");
    scanf("%d",&n);
    int i;
    for (i=1; i<=n; i++)
    {
        if ((i % 2) == 0)
            printf("%d\n",i);
    }
    return 0;
}
```

7. Write a c program that computes the sum of five non-zero integers using while statement.

```c
#include <stdio.h>

int main() {
    int n,i = 0,sum=0;
    while (i < 5){
        printf("Enter an integer to add: ");
```
8. Write a program that reads an integer n from the user input and computes the smallest prime number greater than n. (A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. For example, 2, 3, 5, 7, 11, 13, 17, 19, 23, and 29 are the first ten prime numbers.) We assume the user will only input integers that are greater than 1.

Hint:

You program should implement the following logic:

1. Read number n from user.
2. Increment n.
3. Check if new n is prime:
   4. If prime, print it and we are done;
   Else go to step 2 and repeat.

Furthermore, here is the logic of how to check if a number n is prime or not:

1. Initialize the divisor d to 2.
2. Check if d equals n:
   3. If equal, n is prime and we are done;
   Else go to the next step.
4. Check if n % d == 0:
5. If yes, n is NOT prime and we are done; Else increment d, then go to step 2 and repeat.

```c
#include <stdio.h>

int main() {
    int n, d;
    int isPrime=0;
    printf("Enter an integer n: ");
    scanf("%d",&n);
    n++;
    while (isPrime != 1){ //check if n is prime or not
        for ( d = 2 ; d <= n - 1 ; d++ ){
            if ( n%d == 0)
            {
                isPrime = 0;
            
            n++;
            
            break;
            }
        }
        if ( d == n ){
            isPrime = 1;
            printf("The smallest prime number greater than n: %d",n);
        } 
    }
    return 0;  }
```
Homework Submission

For the questions that you need to write a code, copy and paste your whole code as the answer.
The deliverable for this assignment is the following file:

    hw5.pdf
    Write your name on the top of all your pdf pages.

Be sure to use the specified file name (“hw5.pdf”) and format. To submit your homework, you should upload your file to handin system using the following link:

    http://secure.cse.msu.edu/handin