CSE 220
Fall 2015 Midterm Exam

Name: ____________________

Student ID: ___ ___ ___ ___ ___ ___ ___

Total Time: 50 minutes

Total points: 50

Total # Questions: 5

1 Extra Credit Question.

Rules:

- The exam is closed-book, closed-note, except for one double sided A4 paper.
- 1pt = 1 minute expected for answering the question.
- If you have a question about any problems, raise your hand until the proctor can help you.
1- (9pts) Convert the following numbers into the asked base format. \((\text{number})_b\) means that the number is in the base ‘b’. You also need to show your work process. (each ‘__’ is for one digit)

a) \((87)_{10}\) into base 2: \((010101011)_2\)

b) \((\text{D3A})_{16}\) into base 10: \((3386)_{10}\)

c) \((176)_8\) into base 16: \((7E)_{16}\)
2- (10pts) Assume that we have an integer variable called ‘x’ (means ‘int x;’). First we want to let user give us an integer value to x. Then using bitwise operations, how can we check its third and sixth bits are ‘1’ or not? Complete the following code to complete the task.

```c
#include<stdio.h>
int main() {
    int x;
    printf("Please enter an integer number: ");
    scanf("%d", &x);

    mask1 = 0x24;
    y   =    x & mask1;

    mask2 = 0x24;
    z = y ^ mask2; // xor (exclusive or)

    if( z == 0 ) {
        printf("Both third and sixth bits are 1 \n");
    } else {
        printf("Both third and sixth bits are NOT ‘1’ \n");
    }

    return 0;
}
```
3- (8 pts) The following code is finding a variable that has the maximum value among 5 variables which are all positive (greater than 0). There are couple of errors in it. Please find the errors and correct them.

```c
#include<stdio.h>
int main()
{
    // initializing it with the 0 so all other positive inputs will be greater than this.
    int max = 0;
    int inp;
    for ( int i = 1 ; i <= 5 ; i ++ ) {
        scanf("%d", &inp);
        if (max < inp) {
            max = inp;
        }
    }
    printf(" the maximum is : %d", &max max);

    return 0;
}
```
4- (8 pts) Please explain that what task is the following code is doing. (Following code is completely correct, just explain what the code is doing.)

```c
#include<stdio.h>
int main() {

    int n;
    printf("Please enter a integer number: ");
    scanf(" %d", & n);

    int result = 1;
    int i;
    for ( i = 1 ; i <= n ; i++) {
        result = result * i ;
    }

    printf("%d \n", result);

    return 0;
}
```

This code will get an integer input from the user and calculate the factorial of the input, which is \( n! \), and prints the output. For example if the input is 5, it will display 120 ( \( = 5*4*3*2*1 \)).
5- (15 pts) Write a code that gets a positive integer from user. Then based on the remainder of it divided by 12, prints which month of the year is the input. If the user’s input is a negative integer number, you should display an error message and exit the program. (you can assume that user enters an integer)

```c
#include<stdio.h>
int main() {
    int n;
    printf("Plz enter an integer: ");
    scanf("%d", &n);
    if ( n < 0) {
        printf("The input should be positive number\n");
        return 0;
    }
    int rem = n % 12;
    switch (rem) {
        case 4:
            printf("Jan\n");
            break;
        case 5:
            printf("Feb\n");
            break;
        case 6:
            printf("Mar\n");
            break;
        case 7:
            printf("Apr\n");
            break;
        case 8:
            printf("May\n");
            break;
        case 9:
            printf("Jun\n");
            break;
        case 10:
            printf("Jul\n");
            break;
        case 11:
            printf("Aug\n");
            break;
        case 0:
            printf("Sep\n");
            break;
        case 1:
            printf("Oct\n");
            break;
        case 2:
            printf("Nov\n");
            break;
        case 3:
            printf("Dec\n");
            break;
    }
    return 0;
}
```

<table>
<thead>
<tr>
<th>Input</th>
<th>What to display</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Jan</td>
</tr>
<tr>
<td>5</td>
<td>Feb</td>
</tr>
<tr>
<td>6</td>
<td>Mar</td>
</tr>
<tr>
<td>7</td>
<td>Apr</td>
</tr>
<tr>
<td>8</td>
<td>May</td>
</tr>
<tr>
<td>9</td>
<td>Jun</td>
</tr>
<tr>
<td>10</td>
<td>Jul</td>
</tr>
<tr>
<td>11</td>
<td>Aug</td>
</tr>
<tr>
<td>0</td>
<td>Sep</td>
</tr>
<tr>
<td>1</td>
<td>Oct</td>
</tr>
<tr>
<td>2</td>
<td>Nov</td>
</tr>
<tr>
<td>3</td>
<td>Dec</td>
</tr>
</tbody>
</table>
Extra Credit:

6- (10 pts) Write a code that gets a positive integer from user. Then, using a ‘while()’ or ‘for()’ loop, your code should print all the prime numbers less that the input. (Hint: you need to see if the input number is dividable by any of the integer numbers less than itself or not.)

```c
#include<stdio.h>
int main()
{
    int i;
    int num;
    int count = 0;
    int j;
    printf("Please enter a number: ");
    scanf("%d", &num);
    for (j = num; j >= 1; j--)
    {
        for (i = 1; i <= j; i++)
        {
            if (j % i == 0)
            {
                count = count + 1;
            }
        }
        if (count == 2)
        {
            printf(" %d is the prime number\n", j);
        }
        count = 0;
    }
    return 0;
}
```