1. (1 point) Give the result of the following exclusive-or operation:

\[ 0xa2 \oplus 0xb3 \]

Answer: 0x11

2. (2 points) What is the output of the following program? (which printf()'s results will be displayed?)

```c
#include <stdio.h>
int main() {
    int i = 5, j = 2, k = 4, value;
    if (i > j) {
        if (i > k) {
            printf("1: the greatest number is i : %d \n", i);
        } else {
            printf("2: the greatest number is k : %d \n", k);
        }
    } else {
        if (j > k) {
            printf("3: the greatest number is j : %d \n", j);
        } else {
            printf("4: the greatest number is k : %d \n", k);
        }
    }
}
```

Answer: 1: the greatest number is i : 5

3. (3 points) Following incomplete program will test if the 2nd bit is 1 and the 5th bit is 0 in a given word of 16 bits (i.e., 1 word = 2 bytes). If it is true print "true" else print "false". That is if one or both do not match you should print "false".

For example, if the word is 0x01AB, the 2nd bits is 1 and the 5th bit is 0. So the program should print "true".

<table>
<thead>
<tr>
<th>Bit position:</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Your task is to complete the program by assigning appropriate values to the masks and providing the correct bit-op (bitwise operator).

```c
#include <stdio.h>
```
int main() {
    unsigned short int i, j, mask1, mask2;

    printf("input a 16-bits word in hex\n");
    scanf("%x", &i);  // Use "x" for inputting a hexadecimal value

    mask1 = 0x0012;
    mask2 = 0x0002;
    j = i & mask1;
    j = j ^ mask2;

    /* j is 0 if there is a match of the bits in the same position, */
    /* j is 1 if there is a mismatch of bits in the same position. */
    if (j == 0)
        printf("true\n");
    else
        printf("false\n");
}

4. (4 points) Write a program that determines whether a given year is a leap year or not. Leap years are years in which February has 29 days. Your program will ask the user to enter a year. If the year entered is a leap year, your program will output the following statement:

Year year-entered-by-user is a leap year!

Otherwise, your program will output:

Year year-entered-by-user is NOT a leap year!

To compute whether a year is a leap year, following three criteria must be taken into consideration. A leap year satisfies the following:

(a) The year must be divisible by 4,
(b) The year must not be divisible by 100, unless:
(c) The year is also divisible by 400.

(If you want to test your program on a computer, you can test on the following years: Leap years: 2000, 2008, 2400; Not leap years: 1800, 1900, 2009, 2013, 2500)
#include <stdio.h>
int main() {
    int y;
    printf("type a year\n", y);
    scanf("%d", &y);

    /* Write down your code here */
    int isLeapYear = 0;
    if(!(y % 4)) {
        printf("Year divisible by 4\n");
        if(!(y % 100)) {
            printf("Year divisible by 100\n");
            if(!(y % 400)) {
                printf("Year divisible by 400: leap Year\n");
                isLeapYear = 1;
            } else {
                printf("Year divisible by 100 and not divisible by 400: not a leap Year\n");
            }
        } else {
            printf("Year divisible by 4 and not divisible by 100: leap-year\n");
            isLeapYear = 1;
        }
    } else {
        printf("Year not divisible by 4: not a leap year\n");
    }

    if(isLeapYear )
        printf("Year %d is a leap year\n", y);
    else
        printf("Year %d is NOT a leap year\n", y);
}
}