ITERATION ACTIVITY SHEET/WORKSHEET

TRACING

Give the screen output.

```
int main()
{ int x,y,z,i,j;
  x = 3; y = 1; z = 4;
  for (i = 1; i <= 11; i++)
  {  for (j = x; j >= 1; j--)
        printf(" ");
     printf("w");
     if (y)
       do
       {
          printf("w");
          y++;
          if (y > z)
             y = 0;
       } while (y);
     else
       for (j = 1; j <= z; j++)
       {
         if (((i-1)%3==0)&&(i<10)&& ((j-1)%3 ==0)&&(j !=z)&& (j > 1))
               printf("0");
         else if ((i == 8)&&(j/5 == 1)&& (j <7))
               printf("0");
         else printf(" ");
       }
     printf("w\n");
     if (i < 4)
     {   x--;
        z += 2;
     }
     else if (i >= 8)
          { x++;
          z -= 2;
          if (x > 2)
             y = 1;
          }
  }
   return 0;
}
```

ANALYSIS

1. The following codes will print “Hello” 10 times: [True or False]

```
for (i = -5; i < 5; i++)
    printf("Hello\n");
```

2. It is possible to have a logically correct algorithm that contains an infinite loop. [True or False]

3. The following is an infinite loop: [True or False]

```
while (!0) printf("Hello\n");
```

4. It is possible to have a syntactically correct **do-while** loop which looks like:

```
int n=1;
do ; while (n < 5);
```

5. The following is an infinite loop: [True or False]

```
a = 5; b = 5;
while (a <= b)
    a++; b++;
```

6. In C language, it is possible to have a **while** loop inside the body of a **for** loop which in turn is inside the body of a **do-while** loop. [True or False]

PROGRAMMING PROBLEMS

a. Write a program that accepts as input a binary number and displays its corresponding decimal equivalent. Declare your integer variables as long. A long declaration accommodates 8 digits for a number.
Example:

Enter a binary number: 1101
Decimal equivalent: 13

b. Write a function that computes for the result given a **positive n value**. Analyze and study well the examples below.

Example 1:
Enter start: 2
Enter end : 5
Enter n : 3
Result : -244
(computed from $2! - 3^3 + 4! - 3^5$)

Example 2:
Enter start: 4
Enter end : 6
Enter n : 2
Result : 712
(computed from $4! - 2^5 + 6!$)

Example 3:
Enter start: 3
Enter end : 6
Enter n : 2
Result : 720
(computed from $2^3 + 4! - 2^5 + 6!$)

Example 4:
Enter start: 5
Enter end : 7
Enter n : 1
Result : 720
(computed from $1^5 + 6! - 1^7$)

c. Complete the following program outline:

```
int Divisor( int nNum ) {
      int nDiv = 1;

      while (nNum > 9)
      {     nNum /= 10;
            nDiv *= 10;
      }
      return nDiv;
    }
```

```
      void RemoveLeftDigit (int *nNum )
     {  /* write your code that will update the value of nNum to one that with
           the leftmost digit removed.  Use the function Divisor.
            Examples: *nNum = 435237, *nNum should be 35237
                      *nNum = 7841, *nNum should be 841
                      *nNum = 9, *nNum should be 0
        */
     }

int main () {
/* Write your code that would get a number from the user and then, display all numbers
from 1 to the leftmost digit of that number.  Also display the number without the
leftmost digit. You are required to use the functions that are defined in this problem
(and you are free to introduce additional functions as needed).

      Example 1: Enter number: 435237
                 Output: 1 2 3 4
                 Number: 35237

      Example 2: Enter number: 7841
                 Output: 1 2 3 4 5 6 7
                 Number: 841

      Example 3: Enter number: 9
                 Output: 1 2 3 4 5 6 7 8 9
                 Number: 0
*/
      return 0;
}
```

d. Write a C function that will display the pattern below. The size of the pattern is highly dependent on the given n (supplied by the user). Study the given examples well. Do you notice the kind of pattern you are to create?

Example 1: Enter n : 4

```
XXXXXXX
OXXXXXO
OOXXXOO
OOOXOOO
OOXXXOO
OXXXXXO
XXXXXXX
```

Example 2: Enter n : 3

```
XXXXX
OXXXO
OOXOO
OXXXO
XXXXX
```

e. You were hired by ABC Bank. Your task is to write a C program to compute the interest to be applied on time deposits. The user is required to input two values, specifically: the balance, and the interest rate per annum (in percentage). The program will then compute and output the interest amount, and the new balance. Example interactions are shown below. The items in bold face represent the inputs from the user, the rest are outputs of the program. Your program should display the same outputs as the examples:

Example #1:
Input the balance: **100000.00**
Input the interest rate per annum (in percentage): **3.5**
The interest amount is: 3500.00
The new balance is: 103500.00
Example #2:
Input the balance: **250000.00**
Input the interest rate per annum (in percentage): **5**
The interest amount is: 12500.00
The new balance is: 262500.00