ANS: False. The algorithm is the hardest of solving a problem.

b) A sentinel value must be a value that cannot be confused with a legitimate data value.

ANS: True.

c) Flowlines indicate the actions to be performed.

ANS: False. Flowlines indicate the order in which steps are performed.

d) Conditions written inside decision symbols always contain arithmetic operators (i.e., +, -, *, /, and %).

ANS: False. They normally contain conditional operators.

e) In top-down, stepwise refinement, each refinement is a complete representation of the algorithm.

ANS: True.

For Exercises 3.17 to 3.21, perform each of these steps:

1. Read the problem statement.
2. Formulate the algorithm using pseudocode and top-down, stepwise refinement.
3. Write a C program.
4. Test, debug, and execute the C program.

3.17 Drivers are concerned with the mileage obtained by their automobiles. One driver has kept track of several tankfuls of gasoline by recording miles driven and gallons used for each tankful. Develop a program that will input the miles driven and gallons used for each tankful. The program should calculate and display the miles per gallon obtained for each tankful. After processing all input information, the program should calculate and print the combined miles per gallon obtained for all tankfuls. Here is a sample input/output dialog:.

```
Enter the gallons used (-1 to end): 12.8
Enter the miles driven: 287
The miles / gallon for this tank was 22.421875

Enter the gallons used (-1 to end): 10.3
Enter the miles driven: 200
The miles / gallon for this tank was 19.417475

Enter the gallons used (-1 to end): 5
Enter the miles driven: 120
The miles / gallon for this tank was 24.000000

Enter the gallons used (-1 to end): -1

The overall average miles/gallon was 21.601423
```


ANS:

2) ***Top:***

Determine the average miles/gallon for each tank of gas, and the overall miles/gallon for an arbitrary number of tanks of gas

First refinement:

Initialize variables

Input the gallons used and the miles driven, and calculate and print the miles/gallon for each tank of gas. Keep track of the total miles and the total gallons.

Calculate and print the overall average miles/gallon.

Second refinement:

Initialize totalGallons to zero.
Initialize totalMiles to zero.

Input the gallons used for the first tank.

3)

```
/* Exercise 3.18 Solution */
#include <stdio.h>

int main()
{
   int accountNumber; /* current account's number */
   double balance;    /* current account's starting balance */
   double charges;    /* current account's total charges */
   double credits;    /* current account's total credits */
   double limit;      /* current account's credit limit */

   /* get account number */
   printf( "\nEnter account number ( -1 to end): " );
   scanf( "%d", &accountNumber );

   /* loop until sentinel value read from user */
   while ( accountNumber != -1 ) {
      printf( "Enter beginning balance: " );
      scanf( "%lf", &balance );

      printf( "Enter total charges: " );
      scanf( "%lf", &charges );

      printf( "Enter total credits: " );
      scanf( "%lf", &credits );

      printf( "Enter credit limit: " );
      scanf( "%lf", &limit );

      balance += charges - credits; /* calculate balance */

      /* if balance is over limit, display account number
         with credit limit and balance to two digits of precision */
      if ( balance > limit ) {
         printf( "%s%d\n%s%.2f\n%s%.2f\n%s\n",
            "Account:      ", accountNumber, "Credit limit: ", limit,
            "Balance:      ", balance, "Credit Limit Exceeded." );
      } /* end if */

      /* prompt for next account */
      printf( "\nEnter account number ( -1 to end ): " );
      scanf( "%d", &accountNumber );
   } /* end while */

   return 0; /* indicate successful termination */

} /* end main */
```

3.20 The simple interest on a loan is calculated by the formula

```
interest = principal * rate * days / 365;
```

The preceding formula assumes that `rate` is the annual interest rate, and therefore includes the division by 365 (days). Develop a program that will input `principal`, `rate` and `days` for several loans, and will calculate and display the simple interest for each loan, using the preceding formula. Here is a sample input/output dialog:

```
Enter loan principal ( -1 to end): 1000.00
Enter interest rate: .1
Enter term of the loan in days: 365
The interest charge is $100.00

Enter loan principal ( -1 to end ): 1000.00
Enter interest rate: .08375
Enter term of the loan in days: 224
The interest charge is $51.40

Enter loan principal ( -1 to end ): 10000.00
Enter interest rate: .09
Enter term of the loan in days: 1460
The interest charge is $3600.00

Enter loan principal ( -1 to end ): -1
```


ANS:

2) ***Top:***

For an arbitrary number of loans determine the simple interest for each loan.

First refinement:

Input the principal of the loan, the interest rate and the term of the loan, calculate and print the simple interest for the loan, and process the next loan.

Second refinement:

input the first loan principal in dollars.
While the sentinel value (-1) has not been entered for the loan principal
Input the interest rate
Input the term of the loan in days
Calculate the simple interest for the loan
Print the simple interest for the loan
Input the loan principal for the next loan

3)

```
1  /* Exercise 3.20 Solution */
2  #include <stdio.h>
3
4  int main()
5  {
6     double principal; /* loan principal */
7     double rate;      /* interest rate */
8     double interest;  /* interest charge */
9     int term;         /* length of loan in days */
10
11    /* get loan principal */
12    printf( "Enter loan principal ( -1 to end): " );
13    scanf( "%lf", &principal );
```

ANS:

```
if ( y == 8 ) {
   if ( x == 5 )
      printf( "@@@@@\n" );
}
else {
   printf( "#####\n" );
   printf( "$$$$$\n" );
   printf( "&&&&&\n" );
}
```

3.33 Write a program that reads in the side of a square and then prints that square out of asterisks. Your program should work for squares of all side sizes between 1 and 20. For example, if your program reads a size of 4, it should print

```
****
****
****
****
```

ANS:

```
1  /* Exercise 3.33 Solution */
2  #include<stdio.h>
3
4  int main()
5  {
6     int side;      /* side of square */
7     int temp;      /* temporary integer */
8     int asterisk;  /* asterisk counter */
9
10    printf( "Enter the square side: " ); /* get size of square */
11    scanf( "%d", &side );
12
13    temp = side;
14
15    /* loop through rows of square */
16    while ( side-- > 0 ) {
17       asterisk = temp;
18
19       /* loop through columns of square */
20       while ( asterisk-- > 0 ) {
21          printf( "*" );
22       } /* end inner while */
23
24       putchar( '\n' );
25    } /* end outer while */
26
27    return 0; /* indicate successful termination */
28
29 } /* end main */
```

3.35 A palindrome is a number or a text phrase that reads the same backwards as forwards. For example, each of the following five-digit integers are palindromes: 12321, 55555, 45554 and 11611. Write a program that reads in a five-digit integer and determines whether or not it is a palindrome. [*Hint*: Use the division and remainder operators to separate the number into its individual digits.]

ANS:

```
/* Exercise 3.35 Solution */
#include<stdio.h>

int main()
{
   int number;      /* input number */
   int temp1;       /* first temporary integer */
   int temp2;       /* second temporary integer */
   int firstDigit;  /* first digit of input */
   int secondDigit; /* second digit of input */
   int fourthDigit; /* fourth digit of input */
   int fifthDigit;  /* fifth digit of input */

   printf( "Enter a five-digit number: " ); /* get number */
   scanf( "%d", &number );

   temp1 = number;

   /* determine first digit by integer division by 10000 */
   firstDigit = temp1 / 10000;
   temp2 = temp1 % 10000;

   /* determine second digit by integer division by 1000 */
   secondDigit = temp2 / 1000;
   temp1 = temp2 % 1000;

   temp2 = temp1 % 100;

   /* determine fourth digit by integer division by 10 */
   fourthDigit = temp2 / 10;
   temp1 = temp2 % 10;

   fifthDigit = temp1;

   /* if first and fifth digits are equal */
   if ( firstDigit == fifthDigit ) {

      /* if second and fourth digits are equal */
      if ( secondDigit == fourthDigit ) {

         /* number is a palindrome */
         printf( "%d is a palindrome\n", number );
      } /* end if */
      else { /* number is not a palindrome */
         printf( "%d is not a palindrome\n", number );
      } /* end else */

   } /* end if */
   else { /* number is not a palindrome */
      printf( "%d is not a palindrome\n", number );
   } /* end else */

   return 0; /* indicate successful termination */

} /* end main */
```

```
Enter a 5-digit number: 11727
The number 11727 has 2 seven(s) in it
```

3.40 Write a program that displays the following checkerboard pattern

```
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
```

Your program must use only three output statements, one of each of the following forms:

```
printf( "* " );
printf( " " );
printf( "\n" );
```

ANS:


```
1  /* Exercise 3.40 Solution */
2  #include <stdio.h>
3
4  int main()
5  {
6     int side = 8; /* side counter */
7     int row;      /* row counter */
8     int mod;      /* remainder */
9
10    /* loop 8 times */
11    while ( side >= 1 ) {
12       row = 8; /* reset row counter */
13       mod = side % 2;
14
15       /* loop 8 times */
16       while ( row >= 1 ) {
17
18          /* if odd row, begin with a space */
19          if ( mod != 0 ) {
20             printf( " " );
21             mod = 0;
22          } /* end if */
23
24          printf( "* " );
25          --row;
26       } /* end while */
27
28       printf( "\n" ); /* go to next line */
29       --side;
30    } /* end while */
31
32    return 0; /* indicate successful termination */
33
34 } /* end main */
```