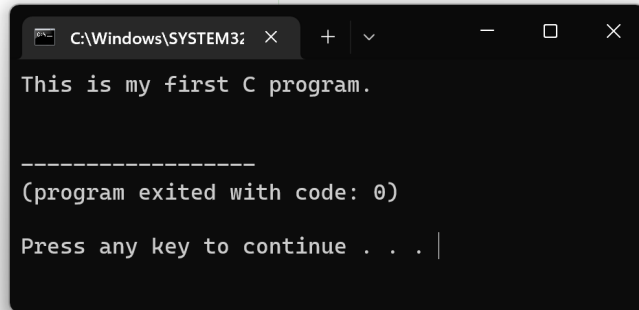Problem 1:

1. Set x equal to the first number in the list
2. If x is greater than the next number in the list, switch the two numbers.
3. Repeat step two until there is no more numbers left in the list
4. Repeat step 1 and set x equal to the new number that is now first in the list. Follow steps 2 and 3.
5. If the list remains the same after step 4, set the next number in the list equal to x and repeat steps 2-4
6. Repeat step 5 until there are no more numbers in the list
7. Repeat steps 2-5

Example:
1. 24, 11, 57, 17, 46
2. Step 1 and x = 24
3. Step 2, x is greater than 11, switch the numbers. new order of the same numbers: 11, 24, 57, 17, 46
4. Step 3: repeat step 2. x is less than 57, so the order of the list doesn`t change
5. Step 3: repeat step 2, x is greater than 17. so x and 17 will be switched. The new order is 11, 17, 57, 24, 46
6. Step 3: repeat step 2, x is less than 46, the order of the list remains the same.
7. No more numbers left in the list. Follow step 4. Set x = 11. X is less than 17, order doesn't change. x is less than 57, order doesn't change. X is less than 24, order doesn't change. X is less than 46, order doesn't change.
8. Step 5, x is now equal to 17. x is less than 57, order doesn't change. x is less than 24, order doesn't change, x is less than 46, order doesn't change.
9. Step 6, repeat step 5, set x equal to 57, x is greater than 24, switch x and 24. New order: 11, 17, 24, 57, 46.
10. X is greater than 46, switch the two. New order: 11, 17, 24, 46, 57.
11. Step 7, repeat steps 2-5, after repeating steps 2-5, the order doesn't change, which means that the task is complete.

Problem 2:

```
1
2   /* My first C program */
3   /*Harsh Solanki CPS188-162 Lab 1*/
4
5   #include <stdio.h>
6
7   int main (void)
8   {
9       printf ("This is my first C program.\n");
10      return (0);
11  }
12
```

```
C:\Windows\SYSTEM32   X    +   ∨              —   ☐   ✕

This is my first C program.


------------------
(program exited with code: 0)

Press any key to continue . . . |
```

Problem 3:

Algorithm:

1. Include stdio library
2. Include math library
3. Create three double data type variables, called a, b, and c. directly representing the a, b, and c values in the hypotenuse formula.
4. Use printf function to let the user know what to enter. "Please enter the a value: "
5. Use scanf function for the code to use the input value of a.
6. Use printf function to let the user know what to enter. "Please enter the b value: "
7. Use scanf function for the code to use the input value of b.
8. Set the c value to be equal to the square root of a*a + b*b
9. Use printf function to print the c value with a statement before it "The length of the hypotenuse is "

Code:
/* Harsh Solanki CPS188-162*/
include <stdio.h>
include <math.h>

int main (void)
{
double a, b, c;

printf("Enter the a value \n");
scanf("%lf", &a);
printf("Enter the b value \n");
scanf("%lf", &b);

c = sqrt(a*a+b*b);

printf("The Hypotenuse is %lf", c);
}

Lab 1 Problem 3.c ✖

```
1   /* Harsh Solanki CPS188-162*/
2
3   # include <stdio.h>
4   # include <math.h>
5
6   int main (void)
7   {
8   double a, b, c;
9
10  printf("Enter the a value \n");
11  scanf("%lf", &a);
12  printf("Enter the b value \n");
13  scanf("%lf", &b);
14
15  c = sqrt(a*a+b*b);
16
17  printf("The Hypotenuse is %lf", c);
18  }
19
```

C:\Windows\SYSTEM32\cmd.(✖   +   ⌄

```
Enter the a value
3
Enter the b value
4
The Hypotenuse is 5.000000

--------------------
(program exited with code: 0)

Press any key to continue . . .
```