Question One

Find the bug in the following code so that it runs correctly

Answer:

```
#include <stdio.h>

int x = 1;

main(){
   if(x==1)
      printf ("x equals 1");
   else
      printf ("x does not equal 1");
   return 0;
}
```

Question Two

Write a header file for a function do_it() that takes three type char arguments and returns a type float to the calling program

Answer: (do_it.h)

```
#ifndef DO_IT_H
#define DO_IT_H

float do_it(char a, char b, char c);

#endif // DO_IT_H
```

Write a header file for a function print_a_number() that takes a single type int argument and returns a type float to the calling program

Answer: (print_a_number.h)

```
#ifndef PRINT_A_NUMBER_H
#define PRINT_A_NUMBER_H

void print_a_number(int x);

#endif // PRINT_A_NUMBER_H
```

What's wrong with the following program

Answer:

The print_msg() function is not supposed to take any arguments but the main function calls it with a string argument.

Question Three

Write a declaration for an array that will hold 50 type long values

Answer:

```
long values[50];
```

Show a statement that assigns the value 123,456 to the 50th element in the array from the above equation

Answer:

```
long values[49] = 123456;
```

What is the value of x when the foolowing statement is complete

Answer:
x is equal to 100 when the statement is complete

What is the value of ctr when the following statement is complete

Answer:
The value of ctr is 11 when the statement is complete

Write a while statement to count from 1 to 100 by 3s

Answer

```
#include <stdio.h>

main(){

   int count = 1;

   while(count <= 100){
      printf("%d ", count);
      count += 3;
   }
}
```

What is wrong with the following code fragment (MAXVALUES is not the problem !)

Answer:
The code under the for loop is always executed once, outside of the for loop, because of the semicolon terminating the for loop.

Question Four

Write a function named addarrays() that accepts two arrays that are of the same size. The function should add each element in the arrays together and place them in a third array

Answer:

```
void addarrays(int array1[], int array2[], int destination_array[], int SIZE){

   for(int i=0;i<SIZE;i++){
      destination_array[i] = array1[i] + array2[i];
   }
}
```

PTO

Modify the function you created in to return a pointer to the array containing the totals. Place this function in a program that also displays the values in all three arrays

Answer:

```
#include <stdio.h>


int *addarrays(int array1[], int array2[], int SIZE);

main(){
   int array1[] = {2,5,3,22,6};
   int array2[] = {13,143,11,10,121};

   int *array3 = addarrays(array1, array2, 5);
   for(int i=0;i<5;i++) {
      printf("%d \n", array3[i]);
   }
}


int *addarrays(int array1[], int array2[], int length){
   int *destination_array = malloc(length * sizeof(int));
   for(int i=0;i<length;i++){
      destination_array[i] = array1[i] + array2[i];
   }
   return destination_array;
}
```