Constants and Macros Assignment:

1. Write a function macro to find the smallest number in an array of integers

   Ans:

   ```
   #define FIND_MIN(arr, size) find_min(arr, size)

   int find_min(int arr[], int size) {
       int min = arr[0];
       for (int i = 1; i < size; i++) {
           if (arr[i] < min) {
               min = arr[i];
           }
       }
       return min;
   }
   ```

2. What are the differences between macros and constant. Can you replace a constant with a macro and vice versa? Give examples for your statements

   Ans:

   ```
   #define FIND_MIN(arr, size) find_min(arr, size)

   int find_min(int arr[], int size) {
       int min = arr[0];
       for (int i = 1; i < size; i++) {
           if (arr[i] < min) {
               min = arr[i];
           }
       }
       return min;
   }
   ```

3. Refer macro below

   ```
   #define MYPROD(x)        (x *x)
   ```

   WAP to invoke the above macro with inputs as below and display the result.

   a. MYPROD(2+1)
   b. MYPROD(6+1)

   Do you get the expected answers as 9 and 49 in case a. and case b.?

   If not modify the code to produce the expected results. in above case

   Ans:

   ```
   #include <stdio.h>


   #define MYPROD(x) ((x) * (x))
   ```

```
int main() {

    int result1 = MYPROD(2 + 1);

    int result2 = MYPROD(6 + 1);


    printf("Result of MYPROD(2+1): %d\n", result1);  // Expected Output: 9

    printf("Result of MYPROD(6+1): %d\n", result2);  // Expected Output: 49


    return 0;

}
```

4. Write macro definitions with arguments for calculation of area of a triangle and circle.
    a. Use macros for both constants as well as formula evaluations.
    b. Store these macro definitions in a header file and invoke the macros from the main function.

Ans:

```
#include <stdio.h>
#include "geometry.h"

int main() {
    double base = 5.0, height = 10.0, radius = 7.0;

    double area_triangle = AREA_OF_TRIANGLE(base, height);
    double area_circle = AREA_OF_CIRCLE(radius);

    printf("Area of Triangle: %.2f\n", area_triangle);  // Output: 25.00
    printf("Area of Circle: %.2f\n", area_circle);      // Output: 153.94

    return 0;
}
```

5. Define a macro name MYPRINT as below.

```
#define MYPRINT(x)          printf(x)
```

Use the above macro conditionally only if a macro CUST_PRINT is defined , otherwise not to be used.

For eg refer the code and comments below.

```
int main()
{
    MYPRINT("Hello World");  // will be displayed only when CUST_PRINT is
defined
    printf("Test");          // will be displayed always irrepective of CUST_PRINT

    return 0;
}
```

Add the code to demonstrate the above behaviour.

Ans:

```
#include <stdio.h>

// Comment out or remove this line to disable MYPRINT functionality
//#define CUST_PRINT

// Define MYPRINT macro
#ifdef CUST_PRINT
    #define MYPRINT(x)    printf(x)
#else
    #define MYPRINT(x)    // No operation if CUST_PRINT is not defined
#endif

int main()
{
    MYPRINT("Hello World\n");  // This will NOT be displayed if CUST_PRINT
is not defined
    printf("Test\n");          // This will be displayed always

    return 0;
}
```

6. Identify and use the macros to display
    a. file name
    b. function name
    c. line of code

    Show the usage with a code example

    Ans:

    __FILE__: This macro gives the name of the current source file.

    __FUNCTION__: This macro provides the name of the current function.

    __LINE__: This macro provides the current line number in the source code.

```
#include <stdio.h>
```

```
void exampleFunction() {
    // Display the file name, function name, and line number
    printf("File Name: %s\n", __FILE__);
    printf("Function Name: %s\n", __FUNCTION__);
    printf("Line Number: %d\n", __LINE__);
}

int main() {
    // Display the file name, function name, and line number from main
    printf("File Name: %s\n", __FILE__);
    printf("Function Name: %s\n", __FUNCTION__);
    printf("Line Number: %d\n", __LINE__);

    // Call another function to show the use of macros inside a different function
    exampleFunction();

    return 0;
}
```