# 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;

}

1. 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;

}

1. Refer macro below

#define MYPROD(x) (x \*x)

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

* 1. MYPROD(2+1)
  2. 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;

}

1. Write macro definitions with arguments for calculation of area of a triangle and circle.
   1. Use macros for both constants as well as formula evaluations.
   2. 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;

}

1. 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;

}

1. Identify and use the macros to display
   1. file name
   2. function name
   3. 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;

}