CSC9V4 Pract. 6: It's *not rude to point!

Introduction

Whole modules could be devoted to pointers and their underlying implementations. Though that may be a bit much for our purposes, time to get started!

Pen and Paper

Attempt the following **on your own first**, before consulting with others.

1. If `i` is a variable and `p` points to `i`, which of the following expressions are aliases for `i`?

```
(a)  *p        (c)  *&p          (e)  *i          (g)  *&i
(b)  &p        (d)  &*p          (f)  &i          (h)  &*i
```

2. If `i` is an `int` and `p` and `q` are pointers to `int`, which of the following assignments are legal?

```
(a)  p = i;        (d)  p = &q;        (g)  p = *q;
(b)  *p = &i;      (e)  p = *&q;       (h)  *p = q;
(c)  &p = q;       (f)  p = q;         (i)  *p = *q;
```

Back to the Source

Write the `sum_average();` function to match its function call, below in bold font. The function will compute the sum and the average of the elements of the array `b`. The program will then print the computed sum and average. You can write an implementation that uses pointers.

Although function definition can go at the bottom of the file, remember to 'declare' the function with a function header before main()!

Please, note that the proposed program may contain syntax or design errors. Test the program and check that it works as expected. If it doesn't, please correct the errors you will identify.

```c
#include <stdio.h>

#define N 5


/* Declare functions here. */


int main(void)
{
  int b[N], I, sum, avg;

  printf(“Enter %d numbers: “, N);
  for (I = 0; I < N; i++)
    scanf(“%d”, &b[i]);

  sum_average(b, N, &sum, &avg);  /* Function to implement. */

  printf("Sum: %d\n", sum);
  printf("Average: %d\n", avg);

  return 0;
}



/* Define functions here. */
```

Check Point.