LAB 1 (Answers to Questions)

* Part 3

1. The outputs of uchar1 and schar1 are different event though they were assigned the same value because uchar1 is ‘unsigned’ while schar1 is signed. Therefore, when the hex number is converted into a decimal, the MSB for the signed one will indicate that the number is negative. However, the MSB for the unsigned one will just print the number without considering negative value.
2. Before the comparison, the value is converted from hex number to decimal and the output, which is bigger number is given to us.
3. The decimal value of schar 1 = -1 and schar2 = -2. Therefore, the output is -3. Same as the expectation.
4. The sum of uchar1 and uchar2 is 509. Same as the expectation.
5. The sum of schar1 and uchar1 is 254 because it’s 255 -1.

* Part 4

1. The internal data type for a “Boolean” is 1.

* Part 5

1. If you shift to the left by some number of places, then you get bigger number than the original number. On the other hand, if you shift to the right by some number of places, then you get smaller number than the original number.
2. When you shift it left more than 3 places, for example 4 places, then you will get 240. Because 15 is 1111 in binary and because of shift it becomes 11110000, which is 240.
3. When you shift it with more than 7 places, for example 8 places, then you will get 3840. The reason is the same as part 5 - #2. 1111 -> 111100000000.

* Part 6

1. I don’t have a clear understanding of pointer. Therefore, this might be wrong but when you print the pointer values themselves, the output will be the address of the register that the pointer is pointing at. Each element takes up to 4 bytes of memory. Therefore, ip and ip + 1 is not different by 1. It goes up 4 more than the previous memory location.

* Part 8

1. The value of the byte after the last character is 0 because it is the end of the string.
2. You get from HAL -> to IBM.
3. If you add integer 60 to the byte following the last character in the string than the value will be beyond what is written in ASCII table.