Solution to Question 1

Subquestion 1

Clause 1

Function h takes 2 functions pointers, both of whom do not take any parameters, however do return an integer.

Function h returns a pointer to a function that does not take any parameters and returns an integer.

Clause 2

Function h simply compares the results of the 2 given function pointers and returns the function pointer whose result is greater in value.

That is achieved by evaluating the result of the 2 given function pointers and comparing them with the greater-than operator, multiplying the function pointer with the value we've gotten from that comparison (either `true` or `false`) with the address of the function that we want to check if its result is greater in value.

This procedure is done twice and the sum of those expressions would be the function pointer whose result is bigger in value. Since in the C language the Boolean value `true` is 1 and `false` is 0, if one of those expressions would be smaller than the other, it will be multiplied 0 and therefore will cancel out, while the other will be multiplied by 1 and left unmodified — returning the function pointer that returns the greater result.

Footnote: Assuming that the type `int` is the equivalent size of the type `int (*)()` is unacceptable. This is prone to segmentation faults on 64-bit systems.

Clause 3

The value of the variable res would be 2.

Subquestion 2

The output is 21587. By converting the 2-dimensional index into 1-dimensional index, with respect to the size of the type `short` (formula below), we are printing the concatenated (binary-level) ASCII values of the characters “S” and “T”.

$$2_{bytes}(3_{columns} \cdot 2_{row} + 3_{column}) = 18_{index}$$