

## Questions

### (a) Explain in your own words what recursion is.

Recursion is when a function which contains a call to itself.

### (b) Create a range-based loop using the auto keyword to loop through the values of array a of type std::array.

```
std::array<std::string, 4> a{"This", "is", "an", "array."};
for (auto i : a) {std::cout<<i<< ' ' ;}
```

The auto keyword infers the variable type from its initialiser. This is used here to create a copy (i) of an array element. In this case variable i becomes an std::string, if a was an array of int, then i would be int.

### (c) When would you use the const keyword?

When the value of a variable should not be changed during the whole program.

### (d) What is the advantage of the std::array type in comparison to built-in arrays (e.g. int a[]).

- 1) For assigning, copying, sorting and moving, built-in arrays can not do these directly, but Std::array has functions for these. So Std::array is more convenient.
- 2) Built-in arrays have to use pointers when the array is passed through a function. This causes some inconveniences. For example, a built-in arrays can not be returned. Only its pointer can be returned. Std::array can be returned safely.
- 3) Some useful information such as boundary can not be memorized in built-in arrays, but can be stored in Std::array.

### (e) Find the error, and explain why it is wrong:

- (1) It will look for a local header file called array, but you want it to look at the system header file called std::array. The correct way would be #include <array>.
- (2) if (a=1) should be if (a==1), because this step is to check whether or not a equals 1, not assign 1 to a. Did not define the data-type of the variable b.
- (3) The loop tries to put a value in b[10], but b is 10 long and therefore does not include index 10.
- (4) A const value can not be changed.

## Design decisions

For traversing the maze, the right-hand-algorithm is used. This requires us to know the facing direction of the traverser. The direction is stored as an integer for each cardinal direction (North: 0, East: 1, South: 2, West: 3). It is implemented in this way to facilitate turning: turning left is a subtraction and turning right is an addition. Another reason for choosing an integer is to be able to directly implement it in a switch statement. The direction dictated the tile checked by the right hand and the tile in front of the traverser. To have the current coordinates of these tiles the orientate() function is created. If according to the right-hand-algorithm is allowed to walk forward, the walkOn() function is called. Once the other edge is reached, it can be assumed the exit is found.