**TASK5: what is the difference between heap and stack?**

**Stack:**

A stack is a region of memory that is used for temporary storage of data in a program. It is organized in a last-in, first-out (LIFO) manner, where the last item that is added to the stack is the first one to be removed. The stack is used for storing local variables, function arguments, and return addresses. The stack size is fixed and is usually limited, typically around a few megabytes.

Heap:

A heap is a region of memory that is used for dynamic memory allocation. It is used for allocating memory for objects or data structures that have a non-fixed size or a lifetime that extends beyond the scope of the current function. Memory allocated on the heap can be accessed by any part of the program as long as a reference to it exists. The heap is much larger than the stack, and the size is usually limited only by the amount of available memory in the system.

Main differences:

Allocation: Memory allocation on the stack is automatic, and the memory is allocated and deallocated in a LIFO manner, without any need for explicit management. On the other hand, memory allocation on the heap is done explicitly by the programmer, and the memory is allocated