EMPLOYEE MANAGEMENT SYSTEM

Arrays are represented in memory as contiguous blocks of memory locations, where each  
location holds a single element of the array. This contiguous allocation allows for efficient  
access and manipulation of elements, as the memory address of each element can be  
calculated using the base address and index. The advantages of arrays include fast access  
and modification of elements, cache efficiency, and minimal memory overhead. Additionally,  
arrays allow for simple implementation of algorithms and data structures, making them a  
fundamental data structure in programming. Overall, arrays provide a compact and efficient  
way to store and manipulate collections of data.

Arrays have limitations, including fixed size, homogeneous elements, and inefficient  
insertion/deletion. They waste memory if not fully utilized. However, arrays excel in  
performance-critical code, cache efficiency, and simple data structures like vectors or  
matrices. Use arrays when data size is fixed and speed is crucial. For dynamic data or varied  
element types, consider alternative data structures like linked lists, vectors, or dictionaries,  
which offer greater flexibility and efficiency in those scenarios.