Assignement 1

The objective of this project is to implement and compare 4 classical sorting algorithms entirely in RISC-V assembler, measuring their performance using a software counter simulating the execution time.
You will work on an **array of statically declared integers** in memory, which each algorithm will have to sort separately.
The main program must:

1. Show the original table.
2. Copy this table into four versions.
3. Apply the 4 sorting algorithms separately.
4. Count the number of statements executed.
5. View sorted results and performance of each algorithm.

Functions to be implemented:

- bubble_sort_asm(array, size)
- selection_sort_asm(array, size)
- insertion_sort_asm(array, size)
- quick_sort_asm(array, left, right)
- print_array_asm(array, size)
- print_perf_asm(label, count)

Expected display:

```
Initial Array :
[34, 7, 23, 87, 12, 5, 9, 66, 18, 42]

Bubble Sort     : [5, 7, 9, 12, 18, 23, 34, 42, 66, 87]
Instructions    : 124

Selection Sort  : [5, 7, 9, 12, 18, 23, 34, 42, 66, 87]
Instructions    : 93

Insertion Sort  : [5, 7, 9, 12, 18, 23, 34, 42, 66, 87]
Instructions    : 78

Quick Sort      : [5, 7, 9, 12, 18, 23, 34, 42, 66, 87]
Instructions    : 47
```

Bonus:
Generates a pseudo-random array.