🧠 Problem: **Dynamic Table Generator**

📝 Scenario

You're building a **basic spreadsheet engine** in C. Each column of the table can be a different data type: `int`, `float`, `char`, or `double`. Users can define how many **columns** and **rows** they want, and the type of each column.

Your task is to create a program that:

1. **Dynamically creates a 2D table** with rows and columns of **varying data types**.
2. Each column can be of a **different type** (like Excel or SQL).
3. Each cell is initialized with **random values** depending on its type.
4. The table is printed in a **formatted way**.
5. After use, all memory is properly freed.

💡 Requirements

- Use a `Column` struct to define each column's type and data.
- Use a `Table` struct to contain an array of columns and the row count.
- Support these types: `'i'`, `'f'`, `'d'`, `'c'`
- Let the user define the number of columns, their types, and the number of rows.

🧩 Example:

For a table with:

- 3 columns: `int`, `char`, `float`
- 4 rows

You might print:

```
Row 0:  32  e  88.62
Row 1:  76  z  12.38
Row 2:  14  b  77.21
Row 3:  59  t  33.49
```

✏️ Structures

```c
typedef struct {
    char type;      // 'i', 'f', 'd', 'c'
    void* data;     // pointer to dynamically allocated column data
} Column;

typedef struct {
    Column* columns;  // array of columns
    int num_columns;
    int num_rows;
} Table;
```

🧪 Bonus Concepts You'll Practice

- Structs + dynamic memory
- Function pointers (if you want to get fancy with type handling)
- Abstracting memory access
- Type casting
- Memory cleanup logic
- Possibly error-checking user input

🛠️ Challenge Tasks

1. Create a function: `Table* createTable(int rows, int cols, char types[])`
2. Fill each column with the correct type and random data
3. Write `void printTable(Table* t)`
4. Write `void freeTable(Table* t)` to free all allocated memory
5. (Optional) Allow saving/loading the table from a file in binary or CSV
