```cpp
#include <iostream>
#include <cstring>

using namespace std;

int main() {
    char c[5][3] = {"12", "34", "56", "78", "90"};

    char (*ptr_1d)[3] = c;
    char (*ptr_2d)[5][3] = &c;

    cout<<ptr_1d<<" "<<*(ptr_1d + 4)<<" "<<ptr_1d[4]<<endl;
    cout<<ptr_2d<<" "<<*( *(ptr_2d) + 4)<<" "<<ptr_2d[0][4]<<endl;
}
```

1. Array Initialization

cpp

Copy code

char c[5][3] = {"12", "34", "56", "78", "90"};

This initializes a **2D character array** c with 5 rows and 3 columns. Each row contains a string (2 characters + a null terminator \0).

The memory layout looks like this:

arduino

Copy code

c[0] = {'1', '2', '\0'}

c[1] = {'3', '4', '\0'}

c[2] = {'5', '6', '\0'}

c[3] = {'7', '8', '\0'}

c[4] = {'9', '0', '\0'}

---

2. Pointer Declarations

Pointer to a 1D Array

cpp

Copy code

```
char (*ptr_1d)[3] = c;
```

- ptr_1d is a **pointer to a 1D array of 3 characters**.
- Since c is a 2D array, c decays to a pointer to its first row (c[0]), which is a 1D array of 3 characters.
- Now, ptr_1d points to c[0].

Pointer to a 2D Array

cpp

Copy code

```
char (*ptr_2d)[5][3] = &c;
```

- ptr_2d is a **pointer to the entire 2D array c**.
- &c gives the address of the entire 2D array, so ptr_2d now points to c.

---

3. First Set of Outputs

ptr_1d

cpp

Copy code

```
cout << ptr_1d << " " << *(ptr_1d + 4) << " " << ptr_1d[4] << endl;
```

1. **ptr_1d**
    - ptr_1d points to c[0]. When printed, it outputs the **address of c[0]**.
    - **Output**: <address of c> (the first row of the array).
2. ***(ptr_1d + 4)**
    - ptr_1d + 4 moves 4 rows forward from c[0], so it points to c[4] (the last row).
    - Dereferencing ptr_1d + 4 gives the contents of c[4], i.e., "90".
    - **Output**: "90".
3. **ptr_1d[4]**
    - ptr_1d[4] is equivalent to *(ptr_1d + 4), so it also points to c[4].
    - **Output**: "90".

First line output:

Copy code

<Address of c> 90 90

---

4. Second Set of Outputs

ptr_2d

cpp

Copy code

```
cout << ptr_2d << " " << *( *(ptr_2d) + 4) << " " << ptr_2d[0][4] << endl;
```

1. **ptr_2d**
    - ptr_2d points to the entire 2D array c. When printed, it outputs the **address of the 2D array**, but since it's a pointer to a pointer, it decays to a pointer to the first element (c[0]).
    - **Output**: Address of c.
2. ***(*(ptr_2d) + 4)**
    - *ptr_2d dereferences ptr_2d, giving the 2D array c.
    - *(ptr_2d) is equivalent to c, so *(*(ptr_2d) + 4) moves 4 rows forward (to c[4]) and dereferences it to get the value "90".
    - **Output**: "90".
3. **ptr_2d[0][4]**

    Gives the character array at the forth row i.e. “90”

Final Output

bash

Copy code

<Address of c> 90 90

<Address of c> 90 90