

1- Write a program that uses a three-dimensional array  $x$  to store integer values. The program prompts the user to input values into the array using three nested loops. After inputting the values, the program prints the content of the array.

اكتب برنامجًا يستخدم array ثلاثية الأبعاد  $x$  لتخزين قيم الأعداد الصحيحة.  
يطلب البرنامج المستخدم بإدخال القيم في array باستخدام ثلاث حلقات متداخلة.  
بعد إدخال القيم، يقوم البرنامج بطباعة محتوى array.

Input

```
1 2 3 4 5 6 7 8 9 2 5 6 5 8 7 9 3 6 5 4 8 2 4 46 3 9 6
```

Output

```
1  2  3
4  5  6
7  8  9

2  5  6
5  8  7
9  3  6

5  4  8
2  4  46
3  9  6
```

# Solution

```
// www.gammal.tech
#include<stdio.h>

int main() {
    int i, j, k, x[3][3][3];

    // Input values into the array
    for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++)
            for (k = 0; k < 3; k++)
                scanf("%d", &x[i][j][k]);

    // Print the content of the array
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            for (k = 0; k < 3; k++) {
                printf("%d\t", x[i][j][k]);
            }
            printf("\n");
        }
        printf("\n");
    }

    return 0;
}
```

2- Write a program that initializes a three-dimensional array x with predefined integer values. The program then prints the content of the array using nested loops.

اكتب برنامجًا يقوم بتهيئة array ثلاثية الأبعاد x بقيم أعداد صحيحة محددة مسبقًا. يقوم البرنامج بعد ذلك بطباعة محتوى array باستخدام الحلقات المتداخلة.

## Output

```
Array Content:
1  2  3
4  5  6
7  8  9

6  5  8
9  4  5
3  5  9

9  8  2
9  6  3
3  3  3
```

# Solution

```
//www.gammal.tech
#include<stdio.h>

int main() {
    int i, j, k, x[3][3][3] = {
        {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},
        {{6, 5, 8}, {9, 4, 5}, {3, 5, 9}},
        {{9, 8, 2}, {9, 6, 3}, {3, 3, 3}}
    };

    // Print the content of the array
    printf("Array Content:\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            for (k = 0; k < 3; k++) {
                printf("%d\t", x[i][j][k]);
            }
            printf("\n");
        }
        printf("\n");
    }

    return 0;
}
```

3- Write a program that initializes a three-dimensional array x with predefined integer values. The program then calculates and prints the sum of all elements in the first row of the array using nested loops.

اكتب برنامجًا يقوم بتهيئة array ثلاثية الأبعاد x بقيم أعداد صحيحة محددة مسبقًا. يقوم البرنامج بعد ذلك بحساب وطباعة مجموع كل العناصر الموجودة في الصف الأول من array باستخدام الحلقات المتداخلة.

## Output

```
Sum of First Row:
Sum: 45
```

# Solution

```
// www.gammal.tech
#include<stdio.h>

int main() {
    int i, j, k, x[3][3][3] = {
        {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},
        {{6, 5, 8}, {9, 4, 5}, {3, 5, 9}},
        {{9, 8, 2}, {9, 6, 3}, {3, 3, 3}}
    };

    int sumFirstRow = 0;

    // Calculate and print the sum of all elements in the first row
    printf("Sum of First Row:\n");
    for (j = 0; j < 3; j++) {
        for (k = 0; k < 3; k++) {
            sumFirstRow += x[0][j][k];
        }
    }
    printf("Sum: %d\n", sumFirstRow);

    return 0;
}
```

4- Write a program that initializes a three-dimensional array x with predefined integer values. The program then calculates and prints the sum of all elements in the second row of the array using nested loops.

اكتب برنامجًا يقوم بتهيئة array ثلاثية الأبعاد x بقيم أعداد صحيحة محددة مسبقًا. يقوم البرنامج بعد ذلك بحساب وطباعة مجموع كافة العناصر الموجودة في الصف الثاني من array باستخدام الحلقات المتداخلة.

## Output

```
Sum of Second Row:
Sum: 54
```

# Solution

```

// www.gammal.tech

#include<stdio.h>

int main() {
    int i, j, k, x[3][3][3] = {
        {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},
        {{6, 5, 8}, {9, 4, 5}, {3, 5, 9}},
        {{9, 8, 2}, {9, 6, 3}, {3, 3, 3}}
    };

    int sumSecondRow = 0;

    // Calculate and print the sum of all elements in the second row
    printf("Sum of Second Row:\n");
    for (j = 0; j < 3; j++) {
        for (k = 0; k < 3; k++) {
            sumSecondRow += x[1][j][k];
        }
    }
    printf("Sum: %d\n", sumSecondRow);

    return 0;
}
```

5- Write a program that initializes a three-dimensional array x with predefined integer values. The program then calculates and prints the sum of all elements in the first column of the array using nested loops.

اكتب برنامجًا يقوم بتهيئة array ثلاثية الأبعاد x بقيم أعداد صحيحة محددة مسبقًا. يقوم البرنامج بعد ذلك بحساب وطباعة مجموع كافة العناصر الموجودة في العمود الأول من array باستخدام الحلقات المتداخلة.

## Output

```
Sum of First Column:
Sum: 44
```

# Solution

```
// www.gammal.tech
#include<stdio.h>

int main() {
    int i, j, k, x[3][3][3] = {
        {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},
        {{6, 5, 8}, {9, 4, 5}, {3, 5, 9}},
        {{9, 8, 2}, {9, 6, 3}, {3, 3, 3}}
    };

    int sumFirstColumn = 0;

    // Calculate and print the sum of all elements in the first column
    printf("Sum of First Column:\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            sumFirstColumn += x[i][0][j];
        }
    }
    printf("Sum: %d\n", sumFirstColumn);

    return 0;
}
```

6- Write a program that initializes a three-dimensional array x with predefined integer values. The program then calculates and prints the sum of all elements in the last column of the array using nested loops.

اكتب برنامجًا يقوم بتهيئة array ثلاثية الأبعاد x بقيم أعداد صحيحة محددة مسبقًا. يقوم البرنامج بعد ذلك بحساب وطباعة مجموع كافة العناصر الموجودة في العمود الأخير من array باستخدام الحلقات المتداخلة.

## Output

```
Sum of Last Column:
Sum: 50
```

# Solution

```

//www.gammal.tech

#include<stdio.h>

int main() {
    int i, j, k, x[3][3][3] = {
        {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},
        {{6, 5, 8}, {9, 4, 5}, {3, 5, 9}},
        {{9, 8, 2}, {9, 6, 3}, {3, 3, 3}}
    };

    int sumLastColumn = 0;

    // Calculate and print the sum of all elements in the last column
    printf("Sum of Last Column:\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            sumLastColumn += x[i][2][j];
        }
    }
    printf("Sum: %d\n", sumLastColumn);

    return 0;
}
```

7- Write a program that initializes a three-dimensional array x with predefined integer values. The program then calculates and prints the sum of all elements in the array using nested loops.

اكتب برنامجًا يقوم بتهيئة array ثلاثية الأبعاد x بقيم أعداد صحيحة محددة مسبقًا. يقوم البرنامج بعد ذلك بحساب وطباعة مجموع كافة العناصر في array باستخدام الحلقات المتداخلة.

## Output

```
Sum of Array Elements: 145
```

## Solution

```
//www.gammal.tech
#include<stdio.h>

int main() {
    int i, j, k, sum = 0, x[3][3][3] = {
        {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},
        {{6, 5, 8}, {9, 4, 5}, {3, 5, 9}},
        {{9, 8, 2}, {9, 6, 3}, {3, 3, 3}}
    };

    // Calculate the sum of all elements in the array
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            for (k = 0; k < 3; k++) {
                sum += x[i][j][k];
            }
        }
    }

    // Print the sum of all elements
    printf("Sum of Array Elements: %d\n", sum);

    return 0;
}
```

8- Write a program that initializes a three-dimensional array x with predefined integer values. The program then calculates and prints the count of occurrences of the value '3' in the array using nested loops.

اكتب برنامجًا يقوم بتهيئة array ثلاثية الأبعاد x بقيم أعداد صحيحة محددة مسبقًا. يقوم البرنامج بعد ذلك بحساب وطباعة عدد تكرارات القيمة "3" في array باستخدام حلقات متداخلة.

## Output

```
Count of Number 3: 6
```



# Solution

```
//www.gammal.tech

#include<stdio.h>

int main() {
    int i, j, k, counter = 0, x[3][3][3] = {
        {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},
        {{6, 5, 8}, {9, 4, 5}, {3, 5, 9}},
        {{9, 8, 2}, {9, 6, 3}, {3, 3, 3}}
    };

    // Calculate the count of occurrences of the value '3' in the array
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            for (k = 0; k < 3; k++) {
                if (x[i][j][k] == 3)
                    counter++;
            }
        }
    }

    // Print the count of occurrences of the value '3'
    printf("Count of Number 3: %d\n", counter);

    return 0;
}
```

9- Write a program that initializes a three-dimensional array x with predefined integer values. The program then calculates and prints the count of occurrences of the values '3' and '5' in the array using nested loops.

اكتب برنامجًا يقوم بتهيئة array ثلاثية الأبعاد x بقيم أعداد صحيحة محددة مسبقًا. يقوم البرنامج بعد ذلك بحساب وطباعة عدد تكرارات القيمتين '3' و '5' في array باستخدام حلقات متداخلة.

## Output

```
Count of Number 3 and 5: 10
```

## Solution

```

//www.gammal.tech

#include<stdio.h>

int main() {
    int i, j, k, counter = 0, x[3][3][3] = {
        {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},
        {{6, 5, 8}, {9, 4, 5}, {3, 5, 9}},
        {{9, 8, 2}, {9, 6, 3}, {3, 3, 3}}
    };

    // Calculate the count of occurrences of the values '3' and '5' in the array
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            for (k = 0; k < 3; k++) {
                if (x[i][j][k] == 3 || x[i][j][k] == 5)
                    counter++;
            }
        }
    }

    // Print the count of occurrences of the values '3' and '5'
    printf("Count of Number 3 and 5: %d\n", counter);

    return 0;
}
```

10- Write a program that initializes a three-dimensional array x with predefined integer values. The program then checks whether the value '5' is present in the array using nested loops. If '5' is found, it prints "found," otherwise, it prints "not found."


اكتب برنامجًا يقوم بتهيئة array ثلاثية الأبعاد x بقيم أعداد صحيحة محددة مسبقًا. يقوم البرنامج بعد ذلك بالتحقق مما إذا كانت القيمة "5" موجودة في array باستخدام حلقات متداخلة. إذا تم العثور على الرقم "5"، فسيتم طباعة "تم العثور عليه"، وإلا فسيتم طباعة "لم يتم العثور عليه".

## Output

Found

# Solution

```


//www.gammal.tech

#include<stdio.h>

int main() {
    int i, j, k, counter = 0, x[3][3][3] = {
        {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},
        {{6, 5, 8}, {9, 4, 5}, {3, 5, 9}},
        {{9, 8, 2}, {9, 6, 3}, {3, 3, 3}}
    };

    // Check if the value '5' is present in the array
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            for (k = 0; k < 3; k++) {
                if (x[i][j][k] == 5)
                    counter++;
            }
        }
    }

    // Print whether '5' is found or not
    if (counter)
        printf("Found\n");
    else
        printf("Not Found\n");

    return 0;
}
```

---