Core Programming

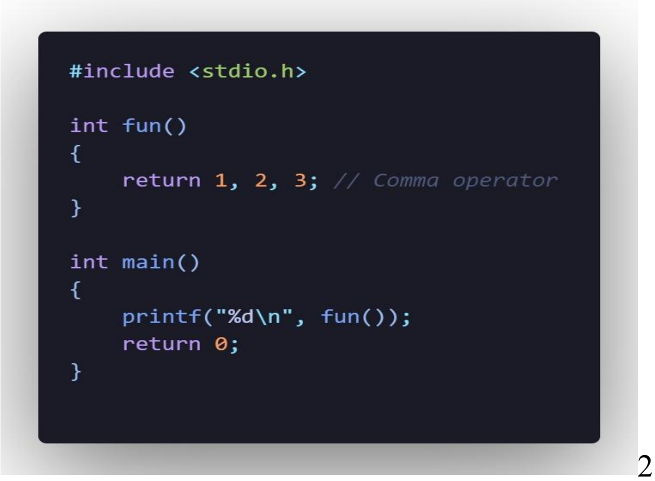
1. Write a program to check whether a given year is a leap year.
2. Write a program to check if a string containing parentheses is balanced. A string is balanced if the opening parentheses have corresponding closing ones in the correct order.
3. Write a program to count the number of 1s in the binary representation of a given integer.
4. Given a 2D matrix, return its elements in spiral order.
5. Implement a function that compresses a string by replacing consecutive repeated characters with the character followed by the count.
6. Given an n x n matrix, rotate it 90 degrees clockwise in place.
7. Write a function to remove duplicates from a sorted array in-place.
8. Given a positive integer n, find the minimum number of operations required to reduce n to 1. The allowed operations are: subtract 1, divide by 2, divide by 3.
9. Given an array, find the next greater element for each element. The next greater element for an element is the first greater element that appears to the right of it in the array.
10. Find the element that appears more than n/2 times in an array. Assume such an element always exists.
11. Given an array of n integers where each integer is in the range 1 to n, some elements appear more than once. Find all duplicates in the array.
12. Write a function to convert a Roman numeral string into an integer.
13. Write a program to count the number of occurrences of a specific element in an array.
14. Write a program to check if a given number is a perfect square.
15. For a given array, find out the maximum sum a sub-array can have.
16. Given an array of non-negative integers, find a contiguous sub-array that sums to a given target S.

Written And Maths

1. Trace the given array using quick sort
   A[] = [5, 3, 2, 6, 4, 1, 3, 7]
2. Write the algorithm of Binary Search.
3. Write the pseudo code of Merge sort.
4. Write down the pseudo code of implementation of Linked list as stack (Push and Pop).
5. Find the coefficient of x^4y^3 in the expansion $(2x + 3y)^7$.

Output Prediction

1.

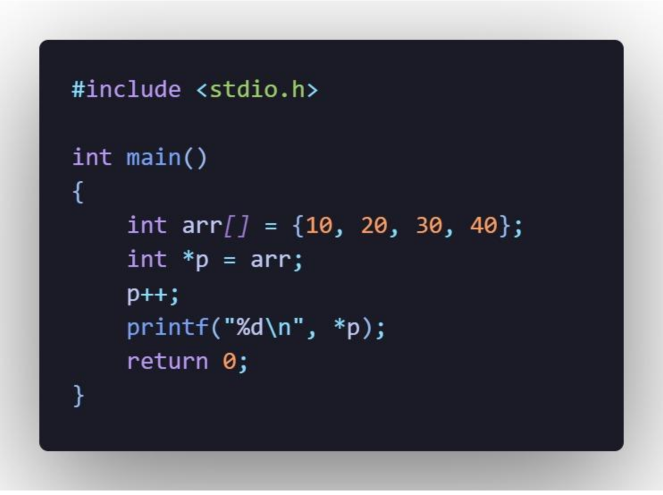


```c
#include <stdio.h>

int fun()
{
    return 1, 2, 3; // Comma operator
}

int main()
{
    printf("%d\n", fun());
    return 0;
}
```

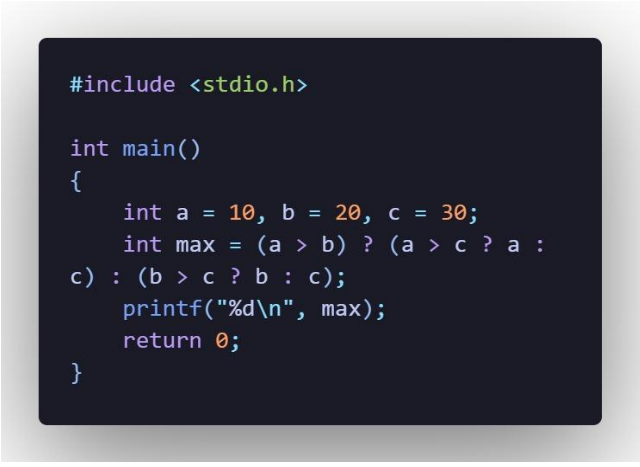
2.



```c
#include <stdio.h>

int main()
{
    int arr[] = {10, 20, 30, 40};
    int *p = arr;
    p++;
    printf("%d\n", *p);
    return 0;
}
```

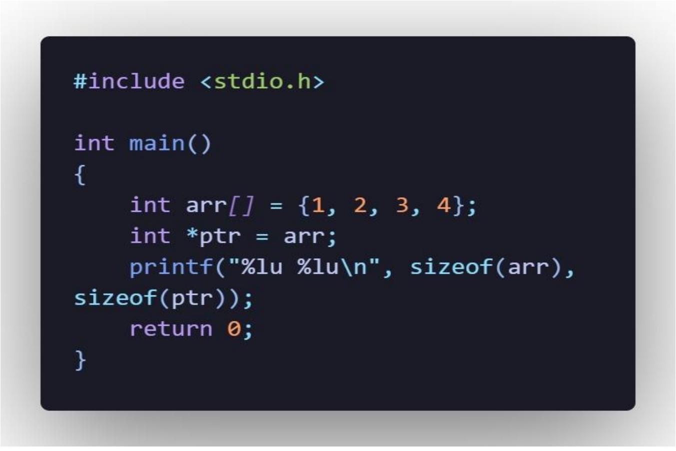
3.



```c
#include <stdio.h>

int main()
{
    int a = 10, b = 20, c = 30;
    int max = (a > b) ? (a > c ? a :
c) : (b > c ? b : c);
    printf("%d\n", max);
    return 0;
}
```

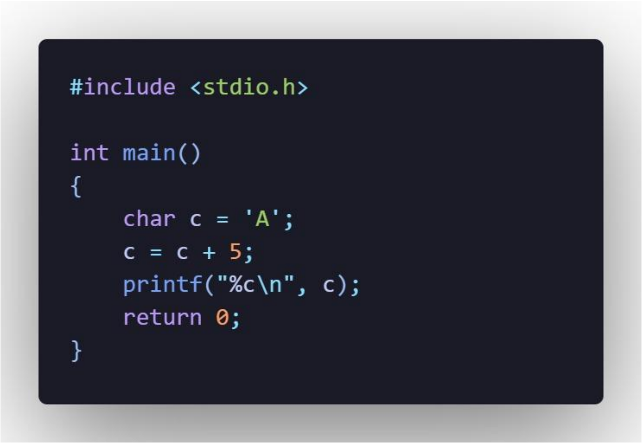
4.



```c
#include <stdio.h>

int main()
{
    int arr[] = {1, 2, 3, 4};
    int *ptr = arr;
    printf("%lu %lu\n", sizeof(arr),
sizeof(ptr));
    return 0;
}
```

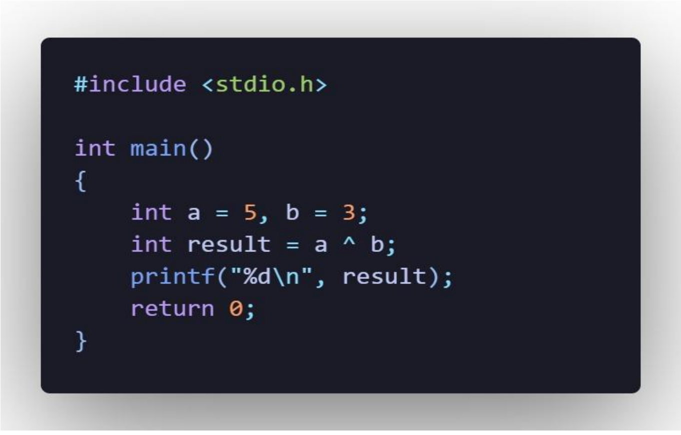
5.



```c
#include <stdio.h>

int main()
{
    char c = 'A';
    c = c + 5;
    printf("%c\n", c);
    return 0;
}
```

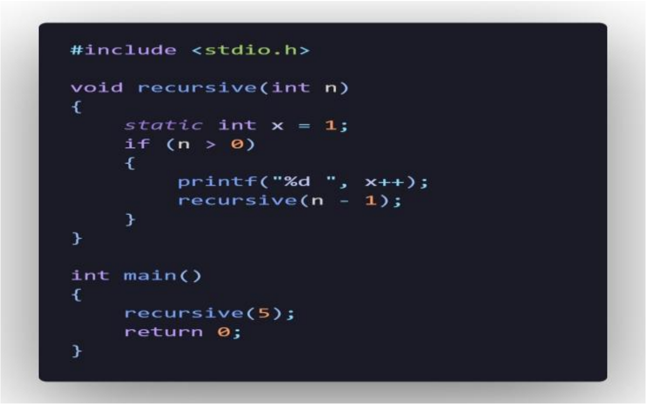
6.



```c
#include <stdio.h>

int main()
{
    int a = 5, b = 3;
    int result = a ^ b;
    printf("%d\n", result);
    return 0;
}
```

7.

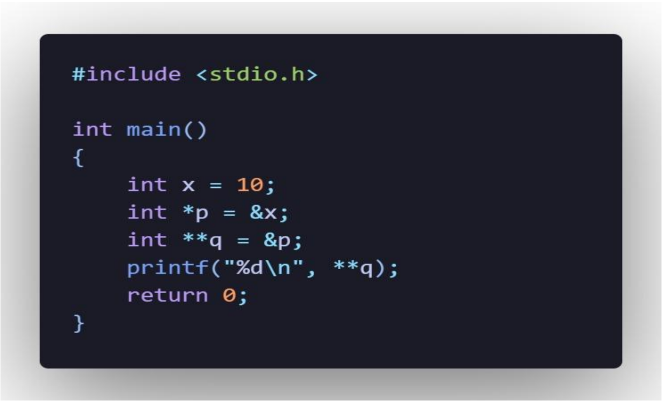


```c
#include <stdio.h>

void recursive(int n)
{
    static int x = 1;
    if (n > 0)
    {
        printf("%d ", x++);
        recursive(n - 1);
    }
}

int main()
{
    recursive(5);
    return 0;
}
```

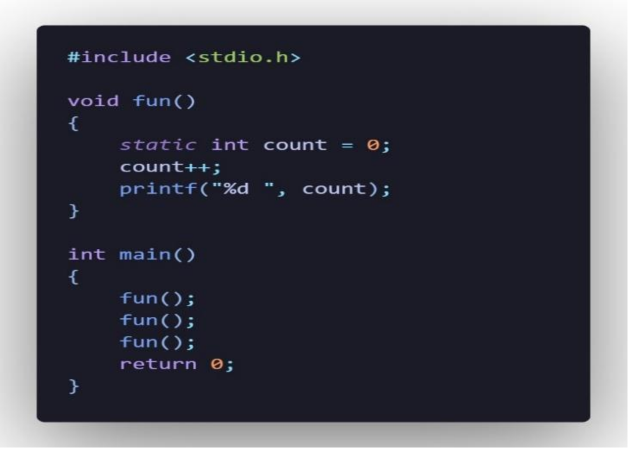
8.



```c
#include <stdio.h>

int main()
{
    int x = 10;
    int *p = &x;
    int **q = &p;
    printf("%d\n", **q);
    return 0;
}
```

9.

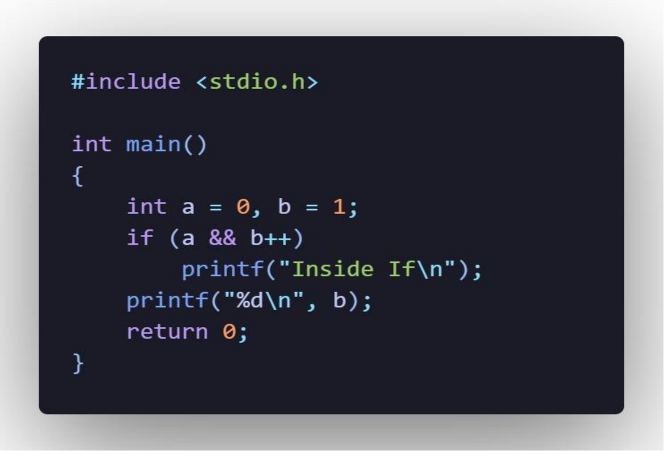


```c
#include <stdio.h>

void fun()
{
    static int count = 0;
    count++;
    printf("%d ", count);
}

int main()
{
    fun();
    fun();
    fun();
    return 0;
}
```

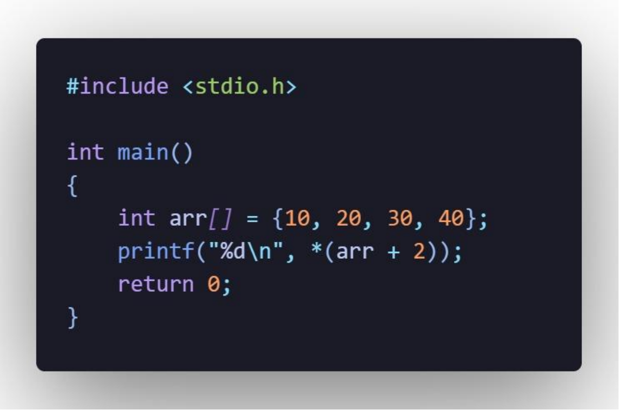
10.



```c
#include <stdio.h>

int main()
{
    int a = 0, b = 1;
    if (a && b++)
        printf("Inside If\n");
    printf("%d\n", b);
    return 0;
}
```

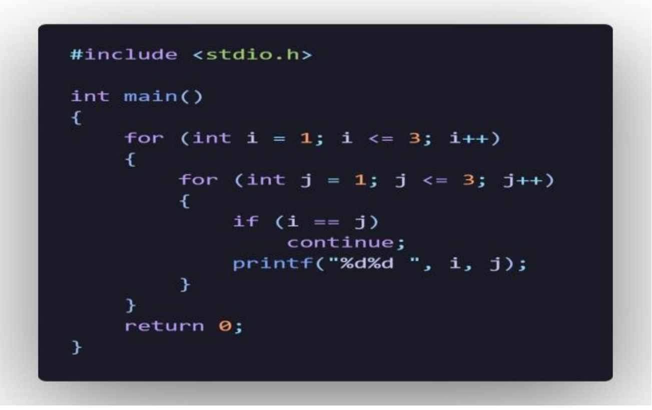
11.



```c
#include <stdio.h>

int main()
{
    int arr[] = {10, 20, 30, 40};
    printf("%d\n", *(arr + 2));
    return 0;
}
```

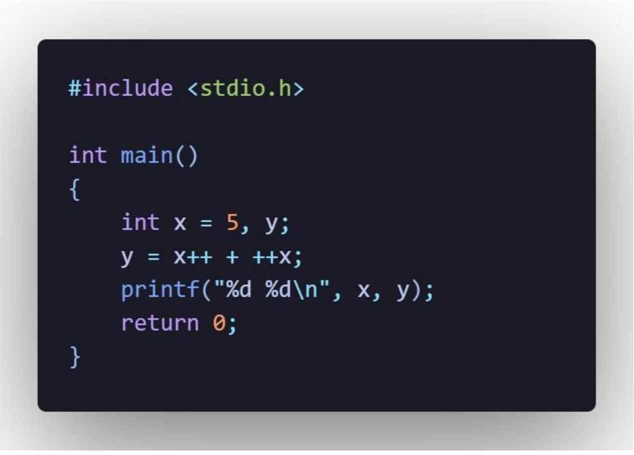
12.



```c
#include <stdio.h>

int main()
{
    for (int i = 1; i <= 3; i++)
    {
        for (int j = 1; j <= 3; j++)
        {
            if (i == j)
                continue;
            printf("%d%d ", i, j);
        }
    }
    return 0;
}
```

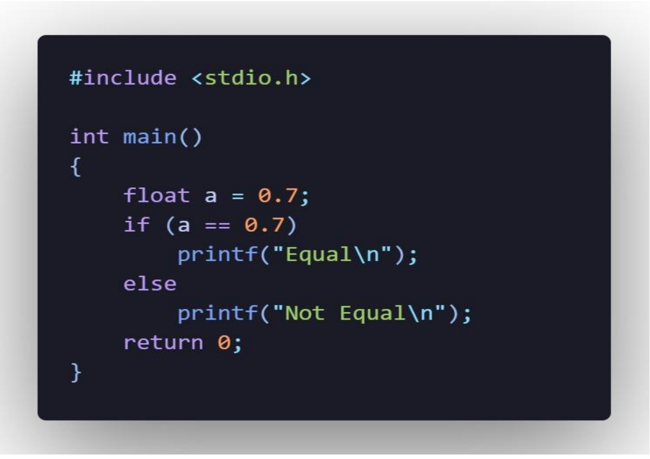
13.



```c
#include <stdio.h>

int main()
{
    int x = 5, y;
    y = x++ + ++x;
    printf("%d %d\n", x, y);
    return 0;
}
```

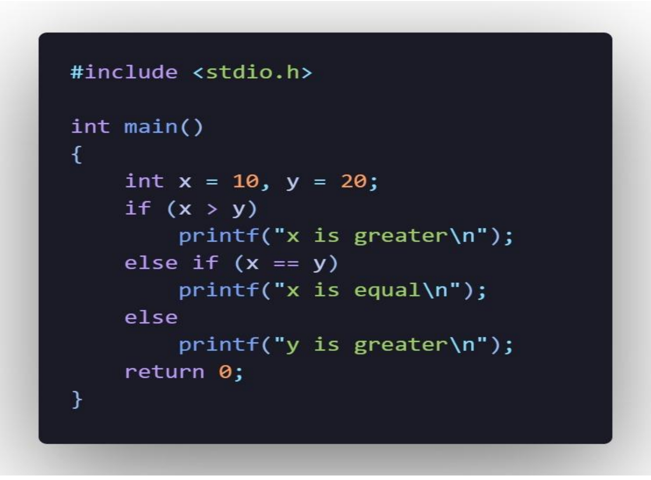
14.



```c
#include <stdio.h>

int main()
{
    float a = 0.7;
    if (a == 0.7)
        printf("Equal\n");
    else
        printf("Not Equal\n");
    return 0;
}
```

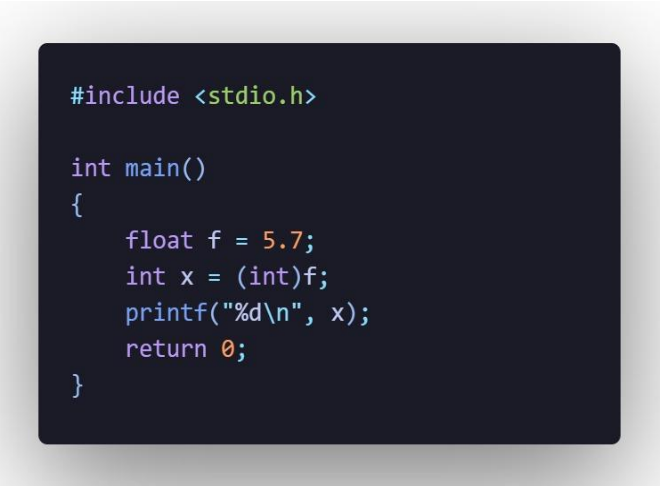
15.



```c
#include <stdio.h>

int main()
{
    int x = 10, y = 20;
    if (x > y)
        printf("x is greater\n");
    else if (x == y)
        printf("x is equal\n");
    else
        printf("y is greater\n");
    return 0;
}
```

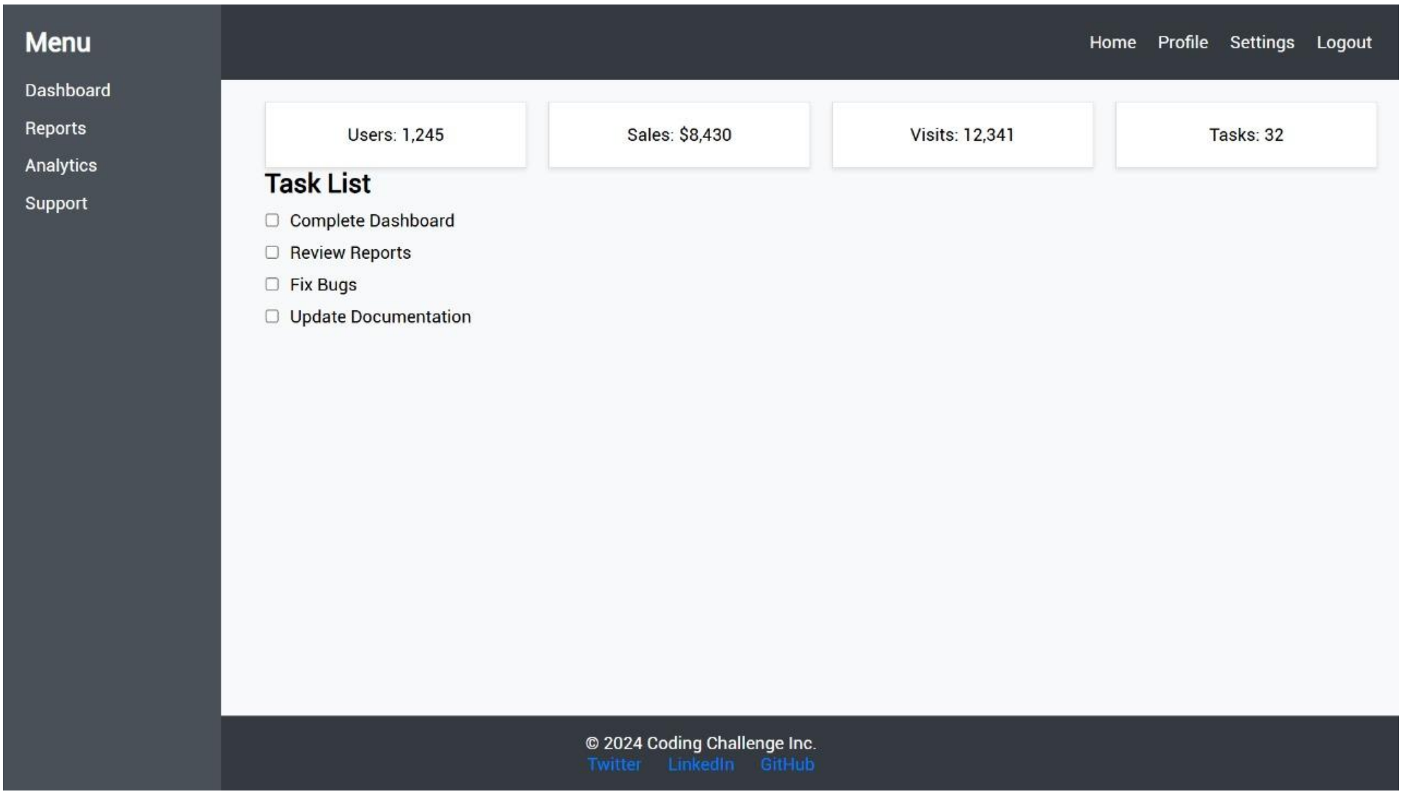
16.



```c
#include <stdio.h>

int main()
{
    float f = 5.7;
    int x = (int)f;
    printf("%d\n", x);
    return 0;
}
```

Design



Users: 1,245

Sales: $8,430

Visits: 12,341

Tasks: 32

Task List

- ☐ Complete Dashboard
- ☐ Review Reports
- ☐ Fix Bugs
- ☐ Update Documentation

© 2024 Coding Challenge Inc.