```cpp
#include <iostream>
using namespace std;

int main() {
    char str[7];
    cout << "Enter your number: ";
    cin >> str;

    int num = atoi(str);
    cout << num << " " << num * 3 << endl;

    return 0;
}
```

```cpp
#include <iostream>
using namespace std;

int main() {
    int N, M;
    cin >> N >> M;

    int matrix[N][M], transpose[M][N];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            cin >> matrix[i][j];
        }
    }
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            transpose[j][i] = matrix[i][j];
        }
    }
    for (int i = 0; i < M; i++) {
        for (int j = 0; j < N; j++) {
            cout << transpose[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}
```

D: > future > Yahia > task1.cpp > main()

```cpp
#include <iostream>
using namespace std;

int main() {
    char str[101];
    cin >> str;

    char groups[101][101];
    int index = 0, groupIndex = 0;

    groups[groupIndex][0] = str[0];
    int pos = 1;

    for (int i = 1; str[i] != '\0'; i++) {
        if (str[i] == str[i - 1]) {
            groups[groupIndex][pos++] = str[i];
        } else {
            groups[groupIndex][pos] = '\0';
            groupIndex++;
            pos = 0;
            groups[groupIndex][pos++] = str[i];
        }
    }
    groups[groupIndex][pos] = '\0';

    for (int i = 0; i <= groupIndex; i++) {
        cout << groups[i] << " ";
    }

    return 0;
}
```

D: > future > Yahia > task1.cpp > main()

```cpp
#include <iostream>
using namespace std;

int main() {
    char str[100];
    cin >> str;

    char result[200];
    int resIndex = 0, count = 1;

    for (int i = 1; str[i - 1] != '\0'; i++) {
        if (str[i] == str[i - 1]) {
            count++;
        } else {
        result[resIndex++] = str[i - 1];
        int numIndex = resIndex;
        while (count > 0) {
        result[resIndex++] = (count % 10) + '0';
           count /= 10;
      }
      for (int j = numIndex, k = resIndex - 1; j < k; j++, k--) {
                char temp = result[j];
                result[j] = result[k];
                result[k] = temp;
            }
            if (str[i] != '\0') result[resIndex++] = '_';
            count = 1;
        }
    }

    result[resIndex] = '\0';
    cout << result;

    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main() {
    char str1[100], str2[100];
    cin >> str1 >> str2;
    int i = 0;
    while (str1[i] != '\0' && str2[i] != '\0') {
        if (str1[i] < str2[i]) {
            cout << "YES";
            return 0;
        }
        if (str1[i] > str2[i]) {
            cout << "NO";
            return 0;
        }
        i++;
    }
    if (str1[i] == '\0') {
        cout << "YES";
    } else {
        cout << "NO";
    }
    return 0;
}
```

```cpp
#include <iostream>
#include <string>
using namespace std;

int main() {
    string num;
    cin >> num;

    int carry = 5, i = num.size() - 1;

    while (i >= 0 && carry) {
        int sum = (num[i] - '0') + carry;
        num[i] = (sum % 10) + '0';
        carry = sum / 10;
        i--;
    }
    if (carry) num = string("5555").substr(0, carry) + num;

    cout << num << endl;
    return 0;
}
```

```
struct Employee {
    string name;  int age; int salary;char gender;    };
Employee employees[100];
int count = 0;
void addEmployee() {
    cin >> employees[count].name >> employees[count].age >> employees[count].salary >> employees[count].gender;
    count++;   }
void printEmployees() {
    for (int i = 0; i < count; i++) {
        cout << employees[i].name << " " << employees[i].age << " " << employees[i].salary << " " << employees[i].gender << endl;
    } }
void deleteByAge() {
    int start, end;
    cin >> start >> end;
    int newCount = 0;
    for (int i = 0; i < count; i++) {
        if (employees[i].age < start || employees[i].age > end) {
            employees[newCount++] = employees[i];
        }  }    count = newCount; }
void updateSalary() {
    string name;
    int newSalary;
    cin >> name >> newSalary;
    for (int i = 0; i < count; i++) {
        if (employees[i].name == name) {
            employees[i].salary = newSalary;
            break;
          }  }  }
int main() {
    int choice;
    while (true) {
        cin >> choice;
        if (choice == 1) addEmployee();
        else if (choice == 2) printEmployees();
        else if (choice == 3) deleteByAge();
        else if (choice == 4) updateSalary(); }
```

```cpp
#include <iostream>
using namespace std;
int main() {
    int N, M;
    cin >> N >> M;
    int matrix[N][M];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            cin >> matrix[i][j];
        }
    }
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            bool isMountain = true;
            int current = matrix[i][j];
            int dx[] = {-1, -1, -1, 0, 1, 1, 1, 0};
            int dy[] = {-1, 0, 1, 1, 1, 0, -1, -1};

            for (int k = 0; k < 8; k++) {
                int ni = i + dx[k], nj = j + dy[k];
                if (ni >= 0 && ni < N && nj >= 0 && nj < M) {
                    if (current <= matrix[ni][nj]) {
                        isMountain = false;
                        break;
                    }
                }
            }

            if (isMountain) {
                cout << i << " " << j << endl;
            }
        }
    }

    return 0;
}
```

```cpp
using namespace std;
char board[9][9];
int N;
void printBoard() {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++)
            cout << (board[i][j] == ' ' ? '.' : board[i][j]) << " ";
        cout << endl;}}
bool checkWin(char p) {
    for (int i = 0; i < N; i++) {
        if (board[i][0] == p && board[i][1] == p && board[i][2] == p) return true;
        if (board[0][i] == p && board[1][i] == p && board[2][i] == p) return true;}
    if (board[0][0] == p && board[1][1] == p && board[2][2] == p) return true;
    if (board[0][2] == p && board[1][1] == p && board[2][0] == p) return true;
    return false; }
int main() {
    cin >> N;
    if (N < 3 || N > 9) return 0;
    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            board[i][j] = ' ';
    char player = 'x';
    while (true) {
        printBoard();
        int r, c;
        cout << "Player " << player << " enter (r, c): ";
        cin >> r >> c;
        if (board[r][c] != ' ') {
            cout << "Invalid! Try again.\n";
            continue; }
        board[r][c] = player;
        if (checkWin(player)) {
            printBoard();
            cout << "Player " << player << " won!\n";
            break; }
        player = (player == 'x') ? 'o' : 'x';
    }
}
```

```cpp
#include <iostream>
using namespace std;

int main() {
    int D, R, C, type;
    cin >> D >> R >> C >> type;

    if (type == 1) {
        int d, r, c;
        cin >> d >> r >> c;
        cout << (d * R * C) + (r * C) + c << endl;
    }
    else if (type == 2) {
        int pos;
        cin >> pos;
        int d = pos / (R * C);
        int r = (pos % (R * C)) / C;
        int c = (pos % (R * C)) % C;
        cout << d << " " << r << " " << c << endl;
    }
}
```

```cpp
#include <iostream>
using namespace std;

int main() {
    int N, M, K;
    cin >> N >> M >> K;
    int x = 0, y = 0;
    for (int i = 0; i < K; i++) {
        int dir, steps;
        cin >> dir >> steps;
        if (dir == 1) x = (x - steps + N) % N;
        if (dir == 2) y = (y + steps) % M;
        if (dir == 3) x = (x + steps) % N;
        if (dir == 4) y = (y - steps + M) % M;
        cout << "(" << x << ", " << y << ")" << endl;
    }
}

```

```cpp
#include <iostream>
using namespace std;

bool isPrime(int n) {
    if (n < 2) return false;
    for (int i = 2; i * i <= n; i++)
        if (n % i == 0) return false;
    return true;
}
int main() {
    int N, M;
    cin >> N >> M;
    int matrix[100][100];
    for (int i = 0; i < N; i++)
        for (int j = 0; j < M; j++)
            cin >> matrix[i][j];
    int Q;
    cin >> Q;
    while (Q--) {
        int i, j, r, c, count = 0;
        cin >> i >> j >> r >> c;
        for (int x = i; x < i + r; x++) {
            for (int y = j; y < j + c; y++) {
                if (isPrime(matrix[x][y])) count++;
            }
        }

        cout << count << endl;
    }
}
```