

```
#include <iostream>
#include <time.h>

using namespace std;

// 0 - puste pole      ->1
// 1 - pudło          --
// 2 - odkryty statek  ->3
// 3 - trafiony statek --
// 7 - ukryty statek   ->3

// int plansza[10][10] = {
//     0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
//     7, 7, 0, 0, 0, 0, 0, 0, 0, 0,
//     0, 0, 0, 7, 0, 7, 7, 7, 0, 0,
//     0, 0, 0, 7, 0, 0, 0, 0, 0, 0,
//     1, 0, 0, 2, 1, 0, 0, 0, 0, 0,
//     0, 0, 0, 7, 0, 0, 0, 0, 0, 0,
//     0, 7, 0, 0, 0, 0, 7, 7, 7, 0,
//     0, 7, 0, 0, 0, 0, 0, 0, 0, 0,
//     0, 7, 0, 0, 2, 3, 2, 7, 0, 0,
//     0, 7, 0, 0, 0, 0, 0, 0, 0, 0};

int plansza[10][10];
int licznik = 0;
```

```
void rysuj_palsze(bool odkryj = false);
void strzal(string w, int k);
void sprawdz_sasiednie(int x, int y);
bool sprawdz_czy_koniec_gry();
void wyczyszc_pansze();
bool kolizja(int x, int y);
bool ustaw_statek_1(int x, int y);
bool ustaw_statek_2(int x, int y, int kierunek);
bool ustaw_statek_3(int x, int y, int kierunek);
bool ustaw_statek_4(int x, int y, int kierunek);
void wylosuj_plansze();

int main()
{
    srand(time(NULL));
    wylosuj_plansze();
    string wiersz;
    int kolumna;
    do
    {
        rysuj_palsze();
        cout << "Podaj litere: ";
        cin >> wiersz;
        cout << "Podaj liczbe: ";
        if (wiersz == "x")
        {
```

```
        rysuj_palsze(true);
        break;
    }
    cin >> kolumna;
    strzal(wiersz, kolumna);
    licznik++;
} while (!sprawdz_czy_koniec_gry() && wiersz != "x");

rysuj_palsze(true);
return 0;
}
```

```
void rysuj_palsze(bool odkryj)
{
    system("cls");
    cout << " ";
    for (int i = 1; i <= 10; i++)
        cout << " " << i << " ";
    cout << endl;
    for (int i = 0; i < 10; i++)
    {
        cout << (char)(i + 65) << " ";
        for (int j = 0; j < 10; j++)
            switch (plansza[i][j])
            {
                case 0:
```

```
        cout << " . ";
        break;
    case 1:
        cout << " * ";
        break;
    case 2:
        cout << (char)176 << (char)176 << (char)176;
        break;
    case 3:
        cout << (char)176 << "#" << (char)176;
        break;
    case 7:
        if (odkryj)
            cout << (char)219 << (char)219 << (char)219;
        else
            cout << " . ";
        break;
    }

    cout << endl;
}
cout << "Strzałow: " << licznik << endl;
}
```

```
void strzal(string w, int k)
{
    char znak = w[0];
    int y = (int)znak - 65;
    int x = k - 1;
    switch (plansza[y][x])
    {
        case 0:
            plansza[y][x] = 1;
            break;
        case 1:
            break;
        case 2:
            plansza[y][x] = 3;
            break;
        case 3:
            break;
        case 7:
            plansza[y][x] = 3;
            break;
    }
    sprawdz_sasiednie(x, y);
}
```

```
void sprawdz_sasiednie(int x, int y)
{
    if (x < 9 && plansza[y][x + 1] == 7)
        plansza[y][x + 1] = 2;
    if (x > 0 && plansza[y][x - 1] == 7)
        plansza[y][x - 1] = 2;
    if (y < 9 && plansza[y + 1][x] == 7)
        plansza[y + 1][x] = 2;
    if (y > 0 && plansza[y - 1][x] == 7)
        plansza[y - 1][x] = 2;
}

bool sprawdz_czy_koniec_gry()
{
    for (int i = 0; i < 10; i++)
        for (int j = 0; j < 10; j++)
            if (plansza[i][j] == 7)
                return false;
    return true;
}

void wyczyszc_pansze()
{
    for (int i = 0; i < 10; i++)
        for (int j = 0; j < 10; j++)
            plansza[i][j] = 0;
}
```

```
}  
  
bool kolizja(int x, int y)  
{  
    if (plansza[x][y] == 7)  
        return true; //pole było zajęte  
    if (x < 0 || x > 9 || y < 0 || y > 9)  
        return true; //pole poza plansza  
    if (plansza[x - 1][y] == 7)  
        return true; //po lewej  
    if (plansza[x + 1][y] == 7)  
        return true; //po prawej  
    if (plansza[x][y - 1] == 7)  
        return true; //górną  
    if (plansza[x][y + 1] == 7)  
        return true; //dół  
    if (plansza[x - 1][y - 1] == 7)  
        return true; //lewa górna  
    if (plansza[x - 1][y + 1] == 7)  
        return true; //lewa dolna  
    if (plansza[x + 1][y - 1] == 7)  
        return true; //prawą górna  
    if (plansza[x + 1][y + 1] == 7)  
        return true; //lewa górna  
    return false;  
}
```

```
bool ustaw_statek_1(int x, int y)
{
    if (kolizja(x, y) == true)
        return false;
    else
        plansza[x][y] = 7;
    return true;
}

bool ustaw_statek_2(int x, int y, int kierunek) // kierunek==0 - poziomo w praw
                                                // kierunek==1 - pionowo w dół
{
    if (kierunek==0 && !kolizja(x,y) && !kolizja(x+1,y))
    {
        plansza[x][y]=7;
        plansza[x+1][y]=7;
        return true;
    }
    if (kierunek == 1 && !kolizja(x, y) && !kolizja(x, y+1))
    {
        plansza[x][y] = 7;
        plansza[x][y+1] = 7;
        return true;
    }
    return false;
}
```



```
bool ustaw_statek_3(int x, int y, int kierunek) // kierunek==0 - poziomo w praw
                                                // kierunek==1 - pionowo w dół
{
    if (kierunek == 0 && !kolizja(x, y) && !kolizja(x + 1, y) && !kolizja(x + 2, y))
    {
        plansza[x][y] = 7;
        plansza[x + 1][y] = 7;
        plansza[x + 2][y] = 7;
        return true;
    }
    if (kierunek == 1 && !kolizja(x, y) && !kolizja(x, y + 1) && !kolizja(x, y + 2))
    {
        plansza[x][y] = 7;
        plansza[x][y + 1] = 7;
        plansza[x][y + 2] = 7;
        return true;
    }
    return false;
}
```

```
bool ustaw_statek_4(int x, int y, int kierunek) // kierunek==0 - poziomo w praw
                                                // kierunek==1 - pionowo w dół
{
    if (kierunek == 0 && !kolizja(x, y) && !kolizja(x + 1, y) && !kolizja(x + 2, y) && !kolizja(x
+ 2, y))
    {
        plansza[x][y] = 7;
        plansza[x + 1][y] = 7;
        plansza[x + 2][y] = 7;
        plansza[x + 3][y] = 7;
        return true;
    }
    if (kierunek == 1 && !kolizja(x, y) && !kolizja(x, y + 1) && !kolizja(x, y + 2) &&
!kolizja(x, y + 3))
    {
        plansza[x][y] = 7;
        plansza[x][y + 1] = 7;
        plansza[x][y + 2] = 7;
        plansza[x][y + 3] = 7;
        return true;
    }
    return false;
}
```

```
void wylosuj_plansze()
{
    wyczysc_pansze();
    int x, y, k, n = 0; // liczba wylosowanych statków

    //statki czteromasztowe (1 sztuki)
    n = 0;
    do
    {
        x = rand() % 10;
        y = rand() % 10;
        k = rand() % 2;
        if (ustaw_statek_4(x, y, k))
            n++;
    } while (n < 1);
    //statki trzymasztowe (2 sztuki)
    n = 0;
    do
    {
        x = rand() % 10;
        y = rand() % 10;
        k = rand() % 2;
        if (ustaw_statek_3(x, y, k))
            n++;
    } while (n < 2);
}
```

```
//statki dwumasztowe (4 sztuki)
n = 0;
do
{
    x = rand() % 10;
    y = rand() % 10;
    k = rand() % 2;
    if (ustaw_statek_2(x, y, k))
        n++;
} while (n < 4);
//statki jednomasztowe (5 sztuk)
n = 0;
do
{
    x = rand() % 10;
    y = rand() % 10;
    if (ustaw_statek_1(x,y))
        n++;
} while (n < 5);
}
```