```c
/**
 * Problem 8 hetmanów - implementacja w C
 * www.algorytm.org
 * @author Tomasz Lubinski (c) 2006
 *
 */

#include "stdio.h"

short int dia_r[256];
short int dia_l[256];   // tablice z zajętymi przekątnymi
short int col[8];       // tablica z zajętymi kolumnami
short int chart[8][8];  // tablica z ustawieniami hetmanów
int found = 0;          // licznik znalezionych pozycji

void printCombination(void) {
        int x,y;
    printf("kombinacja: %d\n", found);
    printf("  ABCDEFGH\n\n");
    for (y=7; y>=0; y--) {
        printf("%d ", y+1);
        for (x=0; x<8; x++) {
            if (chart[x][y]) {
                printf("H");
            }
            else {
                printf(" ");
            }
        }
        printf("\n");
    }
    printf("\n\n");
}

void add_hetman(int row) {
        int i;
    for (i=0; i<8; i++) {
        if (!((col[i] == 1) || (dia_r[i-row+128] == 1) ||
           (dia_l[row+i+128] == 1))) {
            dia_r[i-row+128] = 1;      // dodaj przekątne
            dia_l[row+i+128] = 1;
            col[i] = 1;                // dodaj kolumnę
            chart[i][row] = 1;         // postaw hetmana w tablicy
            if (row<7) {
                add_hetman(row+1);        // analizuj nastepny wiersz
                   (tylko jesli nie jestesmy juz w ostatnim)
            }
```

```
            else {                          // to jest ostatni hetman -
               zapisz wynik
                found++;
                printCombination();
            }
            // po wyjsciu z procedury rekurencyjnej
               (add_hetman(row+1)) usun hetmana i szukaj dla niego
               nastepnego pola
            dia_r[i-row+128] = 0;
            dia_l[row+i+128] = 0;
            col[i] = 0;
            chart[i][row] = 0;
            }
        }
}

int main(void) {
        int i,j;
        //inicjuj strukury
        for (i=0; i<256; i++) {
                dia_r[256] = 0;
                dia_l[256] = 0;
        }
        for (i=0; i<8; i++) {
                col[8] = 0;
        }
        for (i=0; i<8; i++) {
                for (j=0; j<8; j++) {
                        chart[i][j] = 0;
                }
        }

        add_hetman(0);              // postaw pierwszego hetmana

        return 0;
}
```