#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

typedef struct

{

char name [20];

char city [20];

int place;

int score;

} Club;

int main()

{

Club \*teams;

int n, i, j, min, place;

do

{

printf ( "Enter number of clubs (max 10): " );

scanf ( "%i", &n );

}

while ( n < 1 || n > 10 );

teams = ( Club\* ) calloc ( sizeof ( Club ), n );

for ( i = 0; i < n; i++ )

{

printf ( "Enter date for team %i(name, town, place, score): ", i + 1 );

scanf ( "%s %s %i %i", ( teams + i )->name, ( teams + i )->city, & ( teams + i )->place, & ( teams + i )->score );

}

for ( i = 0; i < n; i++ )

{

printf ( "%s %s %i %i\n", ( teams + i )->name, ( teams + i )->city, ( teams + i )->place, ( teams + i )->score );

}

free ( teams );

printf("After free memory\n");

for ( i = 0; i < n; i++ )

{

printf ( "%s %s %i %i\n", ( teams + i )->name, ( teams + i )->city, ( teams + i )->place, ( teams + i )->score );

}

getch();

return 0;

}

Якщо ми звільняємо пам'ять, то це означає, що тепер її можуть використовувати інші програми комп'ютера. Покажчики у собі містять адресу, що означає, що навіть при звільненій пам'яті вони все одно будуть до неї звертатися, а якщо цю пам'ять використає вже інша програма, то буде ситуація коли ми використовуєму не призначену для нас територію. Тому після звільнення пам'яті доцільно обунуляти покажчики.

