Здравейте ас. Ангелов,

Това е решението ми на задачата, което ми идва на пръв поглед.

Ще обясня накратко, въпреки че съм сложил коментари в кода.

Зависимостта която открих е следната – всяка горна граница на ниво е четно число на квадрат и се редуват през 2 , т.е. 4 = 2^2, 16 = (2+2)^(2+2) и т.н.

Сложих първите 10 граници във вектор(**защото не съм много сигурен как ще е най-оптимално**) и ако търсеното число е по-голяма от тази граница, увеличавам вектора докато не стигне числото(**Вероятно ако се разровя в интернет ще открия по-оптимален начин, за да разбера дори дали има смисъл да започвам с някакви стойности във вектора**). После съм написал функция, която представлява двоично търсене в наредения вектор, с малката разлика, че връщам нивото ако бъде открито директно, а ако не бъде, връщам текущата дясна граница + 2, за да получа търсеното ниво. Моля да ми дадете обратна връзка дали да оптимизирам това решение, защото само го нахвърлих или да мисля за друг – по-оптимален начин. Изпращам ви този файл в Moodle и на e-mail.

Също така ако нещо не е ясно по кода ще се радвам да го разясня.

курс Информационни технологии

#include <iostream>  
#include <vector>  
#include <math.h>  
  
using namespace std;  
  
int spiralLevelSearch(vector<int> arr, int l, int r, int x) {  
  
 //Запомняме текущата дясна граница ако рекурсията ни не открие съвпадение  
 int curRight = r;  
 if (r >= l) {  
 int mid = l + (r - l) / 2;  
  
 if (arr[mid] == x)  
 return mid + 1;  
  
 if (arr[mid] > x)  
 return spiralLevelSearch(arr, l, mid - 1, x);  
  
 return spiralLevelSearch(arr, mid + 1, r, x);  
 }  
  
 //Връщаме текущата дясна граница ако рекурсията ни не открие съвпадение  
 return curRight + 2;  
}  
  
int main() {  
  
 cout << "Enter your number : ";  
 int num;  
 cin >> num;  
  
 vector<int> borders(10);  
  
 //Пълним вектор с първите 10 граници на нивата 4,16,36...  
 for (int i = 2, j = 0; j < borders.size(); i += 2) {  
 borders[j] = i \* i;  
 j++;  
 }  
  
 //Ако числото, което търсим е по-голямо от горната граница, разрастваме вектора докато не надхвърли числото  
 if (num > borders.size()) {  
 while (num > borders[borders.size() - 1]) {  
 int nextEl = pow(sqrt(borders[borders.size() - 1]) + 2, 2);  
 borders.push\_back(nextEl);  
 }  
 }  
  
 cout << spiralLevelSearch(borders, 0, borders.size() - 1, num);  
  
 return 0;  
}