Documentation

Overview: project olimpiads can create various competitions, register different students on them and each student has hight or low skill in some science industry.

Classes: project has 3 classes: Skill, Student and Competition

1.Skill: Class Skill has name and level of some skill. Skills is parts of class Students. We can add some new skill and automatically add to every student with level 0, we can manage level with setter

2.Student: Class Student name, surname, vector of Skills, dimension of vector same as amount of skills, because some skills may have 0 level. Each student has fear coefficient that show how student worry about competition, and experience that show how confident student feel on olimpiad. Later we will see that total result has linear dependence from fear coeff and exp. coeff, also result depend from skills. Functions in Student class are setters, getters and function to change level of skill or add some new skill.

3. Competition: Competition has name, type, vector of used skills in this competition, weight of each skill(for example in informatics olimpiad student have to know programing, math, and speak well to explain his work, so he need good math skill, excellent programing skill, and good speaking, so programing has most big influence to total result and we need set programing weight greater than math and math greater than speaking), also class contains date of competition and vector of students(unsorted firstly). Except of setters and getters, function compete build list of student in order based on their skills. We have linear function from student parameters. Every time we add student data to result table(firstly empty), to provide vector always sorted we use binary insertion. Than we add some noise, when some students has approximately total results and one of them has lass fear coeff and smaller position than other, we swap them with 50% probability. Than we reduce fear every time student visit competition(random number from 0 to 9), that we calculate difference between first place and add to experience random number from 0 to 99 divided by difference + 1(we add one that avoid division by zero). Than we push competition into archive(static vector). We can find competitions between some dates, or pick some of them with certain skills used in this competition. findBetweenDates and findBySkills realizes this functionality. Also competition has struct date that defines day month and year of competition organization.

Test: in test we create 6 students and 3 competitions: physics , math, informatics. Than we register students on this competitions in and printed them, and finally we get competitions between some dates and with certain skills