

Мазмұны

Кіріспе.....	4
Теориялық бөлім	4
1.1)C++ тілі туралы.....	4
Тәжірибиелік бөлім.....	5
2.1)Есепте қолданылған C++ тілінің кітапханалары мен технологиялары және Есеп моделі.....	5
2.2)Есептің программасы.....	6
Қорытынды.....	15
Пайдаланылған әдебиеттер.....	15
Қосымша сабақтағы есептер.....	15

КІРІСПЕ

C++ тілін Бьерн Страуструп AT&T компаниясының Bell Labs бөлімшесінде C тіліне қосымша ретінде жасаған. Ол қазіргі уақыттағы ең танымал тілдердің бірі және бағдарламалық жасақтаманы әзірлеу үшін кеңінен қолданылады. Оның көмегімен операциялық жүйелер, әртүрлі қолданбалы бағдарламалар, құрылғы драйверлері, ойындар және т.б. жасалады.

C++-да (немесе оны пайдалану арқылы) жазылған ең танымал бағдарламалық жасақтаманың қатарына MySQL ДҚБЖ, Mozilla Firefox интернет-шолушысы, Microsoft бағдарламалық жасақтамасының көп бөлігі кіреді: Windows операциялық жүйелері, Ide Visual Studio, Explorer, Microsoft Office. Adobe Photoshop, Adobe және Adobe Premiere толығымен C++ - да жазылған.

Менің тәжірибелік жұмысымның тақырыбы C++ тілінде кеңінен қолданылатын технологиялармен танысу

Теориялық бөлім:

1.1) C++ тілі туралы

C++ - бұл жүйелік бағдарламалық жасақтаманы, қосымшаларды, ойындарды және басқа да бағдарламалық өнімдерді әзірлеу үшін кеңінен қолданылатын жоғары деңгейлі, статикалық терілген бағдарламалау тілі.

Кейбір негізгі ерекшеліктері:

- Объектіге бағытталған бағдарламалау : C++ инкапсуляция, мұрагерлік және полиморфизм сияқты ООП негізгі тұжырымдамаларын қолдайды. Бұл әзірлеушілерге модульдік және масштабталатын бағдарламалар жасауға мүмкіндік береді.
- Үлгілер (Templates): үлгілер әртүрлі деректер түрлерімен жұмыс істей алатын жалпыланған функциялар мен сыныптарды жасауға мүмкіндік береді. Бұл кодты икемді етеді және оның қайта пайдаланылуын арттырады.
- Стандартты кітапхана (STL): c++ деректер контейнерлерін (мысалы, векторлар, тізімдер, массивтер), алгоритмдерді (мысалы, сұрыптау, іздеу) және басқа да утилиталарды қамтитын кең стандартты кітапханамен бірге келеді.

Жұмыстың өзектілігі : Технологиялық прогрестің күннен күнге дамып келе жатқан біздің заманымызда білімді бағалау өте өзекті болып табылады. Күн сайын әлемде тез әрі тиімді бағалауды қажет ететін ақпараттар пайда болып

жатыр. Ең қолжетімді бағалау әдістерінің ішінде тестілеуге арналған программалар танымалы және ыңғайлылардың бірі саналады. Бұл бағдарлама әртүрлі салаларда қолданылады және білімді тиімді бақылап қана қоймай сонымен қатар тесттерді пайдаланушылардың әртүрлі қажеттіліктермен мақсаттарына қарай бейімдеу мүмкіндігін береді. Бұл құралдың арқасында мұғалімдер студенттердің білім деңгейін дәлірек бағалай алады, ал студенттер емтиханға тиімдірек дайындалып, үлгерімін өз бетінше бақылай алады және HR мамандары кандидаттардың білімімен дағдыларын бағалай отырып кадрларды сапалы және объективті іріктеу жүргізе алады.

Жұмыстың мақсаты:

- Бірнеше жауабы бар таңдау тесттерін әзірлеу және басқару үшін бағдарламалық жасақтаманы құру.
- Әртүрлі пайдаланушы қажеттіліктеріне сәйкес жүйенің икемділігін қамтамасыз ету.
- Тесттерді жасау және тапсыру үшін ыңғайлы және интуитивті интерфейсті қамтамасыз ету.

Міндеттері:

1. Берілген техникалық талаптар негізінде бағдарлама кодын әзірлеу.
2. Бағдарламаның сенімді жұмысын қамтамасыз ету үшін тестілеу және дебаггинг жасау .
3. Бағдарламаны қолдау және қажет болған жағдайда өзгерістер мен жақсартулар енгізу.
4. Жасалған бағдарламалық өнімді тестілеу

Тәжірибиелік бөлім

2.1) Есепте қолданылған C++ тілінің кітапханалары мен технологиялары және Есеп моделі

1. `std::vector` — элементтердің динамикалық массивін қамтамасыз ететін C++ стандартты кітапханасындағы контейнерлердің бірі. Кәдімгі массивтерден айырмашылығы, `std::vector` бағдарламаның орындалу барысында өзінің өлшемін динамикалық түрде өзгерте алады. Ол массивтің соңындағы элементтерді қосуды және жоюды, сонымен қатар элементтерге индекс бойынша қол жеткізуді қамтамасыз етеді.
2. `std::map` — элементтерді кілт-мән жұбы түрінде сақтайтын контейнер. Әрбір кілт `std::map` ішінде бірегей болып табылады, бұл кілт арқылы мәнді жылдам табуға мүмкіндік береді. `std::map` ағаш деректер құрылымын (әдетте қызыл-қара ағаш) пайдалану арқылы элементтерге жылдам қол жеткізуді қамтамасыз етеді.
3. `std::random_device` — кездейсоқ сандар генераторларына (бар болса) қол жеткізуді қамтамасыз ететін кездейсоқ сандар генераторы. Ол

көбінесе генераторларды инициализациялау үшін пайдаланылатын ерекше кілттерді жасау үшін кездейсоқ сандар генераторларымен бірге қолданылады.

4. OOP (Object Oriented Programming): OOP - бұл объектілер мен класстар тұжырымдамаларына негізделген бағдарламалау әдістемесі. Объектілер – бұл деректермен (өрістер) және осы деректермен өзара әрекеттесуді қамтамасыз ететін әдістерді (функцияларды) қамтуы мүмкін класстар. Код OOP негізгі принциптерін көрсететін Answer, MultipleChoiceQuiz және Exam класстарын пайдаланады.
5. Ағынды басқару бағдарламада нұсқаулардың орындалу ретін басқару әдістерін білдіреді. Код шарттар негізінде орындалу ағынын басқару үшін циклдарды, шартты if-else операторларын және exit() пәрмендерін пайдаланады.
6. Интернационалдандыру және локализация: Интернационалдандыру (i18n) және локализациялау (l10n) әртүрлі тілдер мен тілдерді қолдауға арналған бағдарламалық жасақтаманы әзірлеу процестері болып табылады. Код әртүрлі тілдер үшін интерфейс пен мәтінді локализациялауды қамтамасыз ету үшін std::map құрылымын қолданатын аударма механизмін пайдаланады.
7. Көрсеткіштер мен сілтемелерді пайдалану: Көрсеткіштер мен сілтемелер жадты өңдеуге және функциялар мен нысандар арасында деректерді беруге мүмкіндік береді. Олар деректердің үлкен көлемін тиімді тасымалдау және қажетсіз көшіруді болдырмау үшін қолданылады.
8. Қателерді өңдеу: бұл бағдарламадағы күтпеген жағдайларды өңдеу тәсілдері. Код std::cerr көмегімен енгізу қателерін және қате туралы хабарларды шығару үшін std::cin.fail() сияқты ерекше жағдайларды өңдеу механизмдерін пайдаланады.

Бұл тұжырымдамалар мен технологиялар бірнеше тілдер мен параметрлерді қолдайтын қуатты және икемді тестілеу қосымшасын жасау үшін бірге жұмыс істейді.

Жеке жұмыс : Бір немесе бірнеше дұрыс жауабы бар тестілеуге арналған программа жазу.

```
#include <iostream>
#include <vector>
#include <map>
#include <algorithm>
#include <set>
#include <random>

enum Languages
{
    KZ,
    RU,
    EN,
};

enum Translate
{
```

```

USER_ANSWER_NUM,
WRONG_INDEX,
ERROR_WHILE_CHECKING_ID,
CFG_SHUFFLE_QUESTION,
CFG_SHUFFLE_ANSWERS,
COMPLEXITY_EASY,
COMPLEXITY_MEDIUM,
COMPLEXITY_DIFFICULT,
COMPLEXITY_NOT_SET,
QUESTION,
CORRECT_ANSWER,
CORRECT_ANSWERS,
WRONG_ANSWER,
CORRECT_ANSWER_COUNT,
YOUR_ANSWERS,
YOUR_SCORE,
TYPE_C_OR_Q_TO,
BOOL_TRUE,
BOOL_FALSE,
};

// Configs start

const bool SHUFFLE_ANSWERS = true;
const bool SHUFFLE_QUESTIONS = true;
const Languages UI_LANG = RU;

std::map<Languages, std::map<Translate, std::string>> translate = {
{
    Languages::KZ,
    std::map<Translate, std::string>{
        {Translate::USER_ANSWER_NUM, "Жауап индексін таңдаңыз # "},
        {Translate::WRONG_INDEX, "Индекс жарамсыз. Қайталап көріңіз!"},
        {Translate::ERROR_WHILE_CHECKING_ID, "Қате: Викторина идентификаторларын тексеріңіз!"},
        {Translate::CFG_SHUFFLE_QUESTION, "Сұрақтарды араластыру = "},
        {Translate::CFG_SHUFFLE_ANSWERS, "Жауаптарды араластыру = "},
        {Translate::COMPLEXITY_EASY, "Күрделелілігі : Оңай"},
        {Translate::COMPLEXITY_MEDIUM, "Күрделелілігі : Орташа"},
        {Translate::COMPLEXITY_DIFFICULT, "Күрделелілігі : Қиын"},
        {Translate::COMPLEXITY_NOT_SET, "The Күрделелілігі орнатылмаған"},
        {Translate::QUESTION, "Сұрақ: "},
        {Translate::CORRECT_ANSWER, "Дұрыс жауап #"},
        {Translate::WRONG_ANSWER, "Қате жауап #"},
        {Translate::CORRECT_ANSWER_COUNT, "Дұрыс жауаптар саны: "},
        {Translate::YOUR_ANSWERS, "Сіздің жауаптарыңыз: "},
        {Translate::CORRECT_ANSWERS, "Дұрыс жауаптар: "},
        {Translate::YOUR_SCORE, "Сіздің ұпайыңыз: "},
        {Translate::TYPE_C_OR_Q_TO, "Жалғастыру үшін `с` немесе шығу үшін [`q` немесе кез келген таңба] теріңіз: "},
        {Translate::BOOL_TRUE, "Қосылған"},
        {Translate::BOOL_FALSE, "Өшірілген"},
    },
},
{
    Languages::RU,
    std::map<Translate, std::string>{
        {Translate::USER_ANSWER_NUM, "Выберите индекс ответа # "},
        {Translate::WRONG_INDEX, "Неверный индекс. Попробуйте еще раз!"},
        {Translate::ERROR_WHILE_CHECKING_ID, "Ошибка: Проверьте свои идентификаторы викторины!"},
        {Translate::CFG_SHUFFLE_QUESTION, "Перетасовать вопросы = "},
        {Translate::CFG_SHUFFLE_ANSWERS, "Перетасовать ответы = "},
        {Translate::COMPLEXITY_EASY, "Сложность: Легко"},
        {Translate::COMPLEXITY_MEDIUM, "Сложность: Средняя"},
        {Translate::COMPLEXITY_DIFFICULT, "Сложность: Сложно"},
        {Translate::COMPLEXITY_NOT_SET, "Сложность не установлена"},
        {Translate::QUESTION, "Вопрос: : "},
        {Translate::CORRECT_ANSWER, "Правильный ответ #"},
        {Translate::WRONG_ANSWER, "Неправильный ответ #"},
        {Translate::CORRECT_ANSWER_COUNT, "Количество правильных ответов: "},
        {Translate::YOUR_ANSWERS, "Ваши ответы: "},
        {Translate::CORRECT_ANSWERS, "Правильные ответы: "},
        {Translate::YOUR_SCORE, "Ваша оценка: "},

        {Translate::TYPE_C_OR_Q_TO, "Введите `с`, чтобы продолжить, или [`q` или любой символ], чтобы выйти: "},
        {Translate::BOOL_TRUE, "Активировано"},
        {Translate::BOOL_FALSE, "Неактивировано"},
    },
},
{
    Languages::EN,
    std::map<Translate, std::string>{
        {Translate::USER_ANSWER_NUM, "Choose answer index : # "},
        {Translate::WRONG_INDEX, "Wrong index. Try again!"},
        {Translate::ERROR_WHILE_CHECKING_ID, "Error : Check your quizzes IDs!"},
        {Translate::CFG_SHUFFLE_QUESTION, "SHUFFLE_QUESTION = "},
        {Translate::CFG_SHUFFLE_ANSWERS, "SHUFFLE_ANSWERS = "},
        {Translate::COMPLEXITY_EASY, "Complexity : Easy"},
        {Translate::COMPLEXITY_MEDIUM, "Complexity : Medium"},
        {Translate::COMPLEXITY_DIFFICULT, "Complexity : Difficult"},
    },
},
};

```

```

        {Translate::COMPLEXITY_NOT_SET, "The Complexity:: is not set"},
        {Translate::QUESTION, "Question : "},
        {Translate::CORRECT_ANSWER, "Correct answer #"},
        {Translate::WRONG_ANSWER, "Wrong answer #"},
        {Translate::CORRECT_ANSWER_COUNT, "Correct answer count : "},
        {Translate::YOUR_ANSWERS, "Your answers : "},
        {Translate::CORRECT_ANSWERS, "Correct answers : "},
        {Translate::YOUR_SCORE, "Your score : "},
        {Translate::TYPE_C_OR_Q_TO, "Type `c` to continue or `[q` or any char] to quit : "},
        {Translate::BOOL_TRUE, "Active"},
        {Translate::BOOL_FALSE, "Inactive"},
    },
},
};

// Configs end

enum Color
{
    RED,
    GREEN,
    YELLOW,
    BLUE,
    ORANGE,
    CYAN,
    RESET,
};

std::map<Color, std::string> termColor = {
    {Color::RED, "\033[91m"},
    {Color::GREEN, "\033[92m"},
    {Color::YELLOW, "\033[93m"},
    {Color::BLUE, "\033[34m"},
    {Color::ORANGE, "\u001b[38;5;166m"},
    {Color::CYAN, "\033[36m"},
    {Color::RESET, "\033[0m"},
};

template <typename T>
void shuffleVector(std::vector<T> &v)
{
    std::random_device rd;
    std::mt19937 gen(rd());
    std::uniform_int_distribution<> dis(0, v.size() - 1);
    for (int i = 0; i < v.size(); ++i)
    {
        int randIndex = dis(gen);
        T temp = v[i];
        v[i] = v[randIndex];
        v[randIndex] = temp;
    }
}

class Answer
{
public:
    std::vector<std::string> correctAnswers;
    std::vector<std::string> wrongAnswers;

    std::vector<std::string> getAnswers()
    {
        std::vector<std::string> merged;
        merged.insert(merged.end(), correctAnswers.begin(), correctAnswers.end());
        merged.insert(merged.end(), wrongAnswers.begin(), wrongAnswers.end());
        return merged;
    }

    std::vector<std::string> getShuffled()
    {
        std::vector<std::string> merged;
        merged.insert(merged.end(), correctAnswers.begin(), correctAnswers.end());

        merged.insert(merged.end(), wrongAnswers.begin(), wrongAnswers.end());
        shuffleVector(merged);
        return merged;
    }
};

enum Complexity
{
    EASY,
    MEDIUM,
    DIFFICULT,
};

struct MultipleChoiceQuiz
{
    int id;

```

```

        std::string question;
        Answer answers;
        Complexity complexity;
    };

class Exam
{
private:
    std::vector<MultipleChoiceQuiz> quizzes;

    std::string to_str(int a)
    {
        return std::to_string(a);
    }

    std::string bool_2_str(bool b)
    {
        return (b) ? translate[UI_LANG][Translate::BOOL_TRUE] : translate[UI_LANG][Translate::BOOL_FALSE];
    }

    std::vector<int> getAnswersFromUser(int answerCount, int exclusiveHighBound)
    {
        std::set<int> result;
        for (int i = 0; i < answerCount; ++i)
        {
            int input;
            std::cout << translate[UI_LANG][Translate::USER_ANSWER_NUM] << i + 1 << ": ";
            std::cin >> input;
            if (input < 0 or input >= exclusiveHighBound or std::cin.fail())
            {
                std::cin.clear();
                std::cin.ignore(32767, '\n');
                --i;
                std::cout << translate[UI_LANG][Translate::WRONG_INDEX] << "\n";
                continue;
            }
            result.insert(input);
        }
        return std::vector<int>(result.begin(), result.end());
    }

public:
    Exam(std::vector<MultipleChoiceQuiz> _quizzes)
    {
        addQuizzes(_quizzes);
    }

    void addQuizzes(std::vector<MultipleChoiceQuiz> _quizzes)
    {
        quizzes.insert(quizzes.end(), _quizzes.begin(), _quizzes.end());
        checkDuplicatedIDs();
    }

    auto getQuizzes()
    {
        return quizzes;
    }

    void checkDuplicatedIDs()
    {
        std::set<int> IDs;
        for (const auto &quiz : quizzes)
            IDs.insert(quiz.id);

        if (quizzes.size() != IDs.size())
        {
            std::cout << translate[UI_LANG][Translate::ERROR_WHILE_CHECKING_ID] << "\n";
            for (const auto &quiz : quizzes)
                std::cout << quiz.id << " ";
            exit(1);
        }
    }

    void debug(bool printable)
    {
        std::string log = "";

        log.append(translate[UI_LANG][Translate::CFG_SHUFFLE_QUESTION] + bool_2_str(SHUFFLE_QUESTIONS) + "\n");
        log.append(translate[UI_LANG][Translate::CFG_SHUFFLE_ANSWERS] + bool_2_str(SHUFFLE_ANSWERS) + "\n");

        for (const auto &quiz : quizzes)
        {
            // id
            log += std::string("id : " + to_str(quiz.id) + "\n");

            // complexity
            switch (quiz.complexity)

```

```

{
    case Complexity::EASY:
        log += std::string(translate[UI_LANG][Translate::COMPLEXITY_EASY] + "\n");
        break;
    case Complexity::MEDIUM:
        log += std::string(translate[UI_LANG][Translate::COMPLEXITY_MEDIUM] + "\n");
        break;
    case Complexity::DIFFICULT:
        log += std::string(translate[UI_LANG][Translate::COMPLEXITY_DIFFICULT] + "\n");
        break;
    default:
        std::cerr << translate[UI_LANG][Translate::COMPLEXITY_NOT_SET];
        exit(1);
}
// question
log += std::string(translate[UI_LANG][Translate::QUESTION] + quiz.question + "\n");

// answers
for (int i = 0; i < quiz.answers.correctAnswers.size(); ++i)
    log += std::string(translate[UI_LANG][Translate::CORRECT_ANSWER] + to_str(i) + " " + quiz.answers.correctAnswers[i] +
                                                                "\n");

for (int i = 0; i < quiz.answers.wrongAnswers.size(); ++i)
    log += std::string(translate[UI_LANG][Translate::WRONG_ANSWER] + to_str(i) + " " + quiz.answers.wrongAnswers[i] + "\n");
log += std::string("\n");
}
if (printable)
    std::cout << log;
}

void conduct()
{
    // std::cout << "Conducting exam..." << "\n";
    while (true)
    {
        // Generate quiz index for default or shuffled
        std::vector<int> quiz_indexes;
        quiz_indexes.reserve(quizzes.size());
        for (int i = 0; i < quizzes.size(); ++i)
            quiz_indexes.push_back(i);
        if (SHUFFLE_QUESTIONS)
            shuffleVector(quiz_indexes);

        // Start exam...
        for (const auto &quiz_it : quiz_indexes)
        {
            MultipleChoiceQuiz quiz = quizzes[quiz_it];

            std::cout << termColor[Color::BLUE] << "id: " << quiz.id << " " << quiz.question << " " << termColor[Color::CYAN];
            switch (quiz.complexity)
            {
                case Complexity::EASY:
                    std::cout << translate[UI_LANG][Translate::COMPLEXITY_EASY] << "\n";
                    break;
                case Complexity::MEDIUM:
                    std::cout << translate[UI_LANG][Translate::COMPLEXITY_MEDIUM] << "\n";
                    break;
                case Complexity::DIFFICULT:
                    std::cout << translate[UI_LANG][Translate::COMPLEXITY_DIFFICULT] << "\n";
                    break;
                default:
                    std::cerr << translate[UI_LANG][Translate::COMPLEXITY_NOT_SET];
                    exit(1);
            }

            std::vector<std::string> answers;
            if (SHUFFLE_ANSWERS)
                answers = quiz.answers.getShuffled();
            else
                answers = quiz.answers.getAnswers();

            std::cout << termColor[Color::RESET] << "\n";

            for (int ans_i = 0; ans_i < answers.size(); ++ans_i)
                std::cout << termColor[Color::ORANGE] << ans_i << " " << answers[ans_i] << "\n"
                            << termColor[Color::RESET];
            std::cout << "\n";

            std::vector<std::string> correctAnswers = quiz.answers.correctAnswers;

            std::cout << translate[UI_LANG][Translate::CORRECT_ANSWER_COUNT] << correctAnswers.size() << "\n";
            std::vector<int> userAnswers = getAnswersFromUser(correctAnswers.size(), answers.size());
            std::cout << "\n";
            std::cout << termColor[Color::YELLOW] << translate[UI_LANG][Translate::YOUR_ANSWERS];
            for (const auto &i : userAnswers)
                std::cout << answers[i] << " ";
            std::cout << "\n";

            std::cout << translate[UI_LANG][Translate::CORRECT_ANSWERS];

```



```

        for (const auto &e : correctAnswers)
            std::cout << e << " ";
        std::cout << "\n";

        int score = 0;
        for (const auto &userInput : userAnswers)
        {
            if (std::find(correctAnswers.begin(), correctAnswers.end(), answers[userInput]) != correctAnswers.end())
                score += 1;
        }

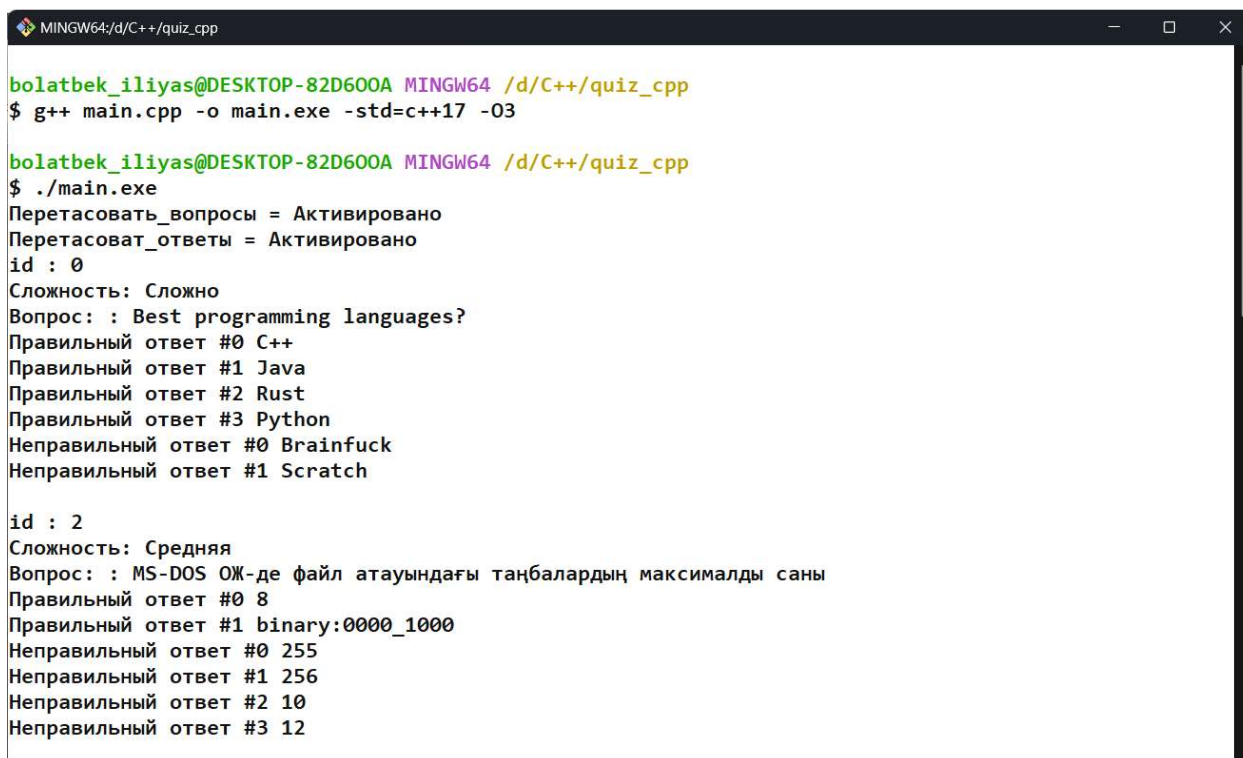
        std::cout << "\n";
        std::cout << termColor[Color::GREEN] << translate[UI_LANG][Translate::YOUR_SCORE] << score << "/" << correctAnswers.size()
                                                                << "\n";
        std::cout << "_____";
        std::cout << "\n\n\n";
    }
    char isExit;
    std::cout << termColor[Color::RESET] << translate[UI_LANG][Translate::TYPE_C_OR_Q_TO];
    std::cin >> isExit;
    if (isExit == 'c')
        continue;
    exit(1);
}
}
};
int main()
{
    Exam exam(
        std::vector<MultipleChoiceQuiz>{
            MultipleChoiceQuiz{
                0,
                "Best programming languages?",
                Answer{
                    std::vector<std::string>{"C++", "Java", "Rust", "Python"},
                    std::vector<std::string>{"Brainfuck", "Scratch"},
                },
                Complexity::DIFFICULT,
            },
            MultipleChoiceQuiz{
                2,
                "MS-DOS ОЖ-де файл атауындағы таңбалардың максималды саны",
                Answer{
                    std::vector<std::string>{"8", "binary:0000_1000"},
                    std::vector<std::string>{"255", "256", "10", "12"},
                },
                Complexity::MEDIUM,
            },
            MultipleChoiceQuiz{
                3,
                "Операциялық жүйелер құрамына мыналар кіреді:",
                Answer{
                    std::vector<std::string>{"жүйелік бағдарламалық қамтамасыз ету"},
                    std::vector<std::string>{"қолданбалы бағдарламалық қамтамасыз ету",
                                            "бағдарламалау жүйелері",
                                            "деректер базасын басқару жүйелері",
                                            "бірегей бағдарламалық қамтамасыз ету"},
                },
                Complexity::EASY,
            },
            MultipleChoiceQuiz{
                4,
                "WINDOWS операциялық жүйесі келесі функцияны орындамайды",
                Answer{
                    std::vector<std::string>{"бағдарламалық жүйелерді құру"},
                    std::vector<std::string>{"қолданбалы бағдарламаларды іске қосу",
                                            "файлдық жүйемен жұмыс істеуді қамтамасыз ету",
                                            "компьютердің аппараттық құралдарын басқару",
                                            "бірнеше қосымшалардың бір уақытта жұмыс істеуін қамтамасыз етеді"},
                },
                Complexity::MEDIUM,
            },
            MultipleChoiceQuiz{
                5,
                "Command.com бағдарламасы ...",
                Answer{
                    std::vector<std::string>{"пайдаланушы енгізген командаларды өңдейді"},
                    std::vector<std::string>{"операциялық жүйенің барлық командаларын сақтайды",
                                            "командалар мен бағдарламаларды өңдейді",
                                            "барлық командаларды өз жұмысында сақтайды",
                                            "дискіні пішімдейді"},
                },
                Complexity::EASY,
            },
            MultipleChoiceQuiz{
                6,
                "Жедел жадтың жылдамдығы немен өлшенеді",
                Answer{
                    std::vector<std::string>{"наносекундпен", "nanosec"},
                    std::vector<std::string>{"миллисекундпен",
                                            "секундтармен",
                                            "мегагерцпен"},
                },
            },
        }
    );
}

```

```
        },  
        Complexity::EASY},  
    });  
    exam.debug(true);  
    exam.conduct();  
}
```

Нәтижесі:

Сурет-1



```
MINGW64:/d/C++/quiz_cpp  
bolatbek_iliyas@DESKTOP-82D600A MINGW64 /d/C++/quiz_cpp  
$ g++ main.cpp -o main.exe -std=c++17 -O3  
  
bolatbek_iliyas@DESKTOP-82D600A MINGW64 /d/C++/quiz_cpp  
$ ./main.exe  
Перетасовать_вопросы = Активировано  
Перетасовать_ответы = Активировано  
id : 0  
Сложность: Сложно  
Вопрос: : Best programming languages?  
Правильный ответ #0 C++  
Правильный ответ #1 Java  
Правильный ответ #2 Rust  
Правильный ответ #3 Python  
Неправильный ответ #0 Brainfuck  
Неправильный ответ #1 Scratch  
  
id : 2  
Сложность: Средняя  
Вопрос: : MS-DOS ОЖ-де файл атауындағы таңбалардың максималды саны  
Правильный ответ #0 8  
Правильный ответ #1 binary:0000_1000  
Неправильный ответ #0 255  
Неправильный ответ #1 256  
Неправильный ответ #2 10  
Неправильный ответ #3 12
```

Бұл суретте exam.debug(true); функциясының шақырылғандағы экранға шығарылған қосымша ақпарат көрсетілген. Егер тек қана тексеру керек болса false мәнімен шақыруға болады. Бағдарламада қолданушының қате енгізген ақпаратын өңдейтін және қолданушының ыңғайына қарай өзгертуге болатын және тілді өзгертетін функциялар бар.

Сурет -2

```
MINGW64:d/C++/quiz_cpp
Неправильный ответ #1 секундтармен
Неправильный ответ #2 мегагерцпен
Неправильный ответ #3 гигагерцпен

id: 4) WINDOWS операциялық жүйесі келесі функцияны орындамайды Сложность: Средняя

0) қолданбалы бағдарламаларды іске қосу
1) файлдық жүйемен жұмыс істеуді қамтамасыз ету
2) бағдарламалық жүйелерді құру
3) компьютердің аппараттық құралдарын басқару
4) бірнеше қосымшалардың бір уақытта жұмыс істеуін қамтамасыз етеді

Количество правильных ответов: 1
Выберите индекс ответа # 1: -1
Неверный индекс. Попробуйте еще раз!
Выберите индекс ответа # 1: 5
Неверный индекс. Попробуйте еще раз!
Выберите индекс ответа # 1: asdf
Неверный индекс. Попробуйте еще раз!
Выберите индекс ответа # 1: 0

Ваши ответы: қолданбалы бағдарламаларды іске қосу
Правильные ответы: бағдарламалық жүйелерді құру

Ваша оценка: 0/1

id: 3) Операциялық жүйелер құрамына мыналар кіреді: Сложность: Легко

0) деректер базасын басқару жүйелері
1) бағдарламалау жүйелері
2) қолданбалы бағдарламалық қамтамасыз ету
```

Сурет -3

```
MINGW64:d/C++/quiz_cpp
Сұрақ: Жедел жадтың жылдамдығы немен өлшенеді
Дұрыс жауап #0 наносекундпен
Дұрыс жауап #1 nanosec
Қате жауап #0 миллисекундпен
Қате жауап #1 секундтармен
Қате жауап #2 мегагерцпен
Қате жауап #3 гигагерцпен

id: 2) MS-DOS ОЖ-де файл атауындағы таңбалардың максималды саны Күрделелігі : Орташа

0) 8
1) 255
2) binary:0000_1000
3) 256
4) 12
5) 10

Дұрыс жауаптар саны: 2
Жауап индексін таңдаңыз # 1: 1
Жауап индексін таңдаңыз # 2: 3

Сіздің жауаптарыңыз: 255 256
Дұрыс жауаптар: 8 binary:0000_1000

Сіздің ұпайыңыз: 0/2

id: 4) WINDOWS операциялық жүйесі келесі функцияны орындамайды Күрделелігі : Орташа

0) бағдарламалық жүйелерді құру
1) файлдық жүйемен жұмыс істеуді қамтамасыз ету
2) компьютердің аппараттық құралдарын басқару
3) бірнеше қосымшалардың бір уақытта жұмыс істеуін қамтамасыз етеді
4) қолданбалы бағдарламаларды іске қосу

Дұрыс жауаптар саны: 1
Жауап индексін таңдаңыз # 1:
```

Сурет -4

```
MINGW64:d/C++/quiz_cpp
Complexity : Easy
Question : Жедел жадтың жылдамдығы немен өлшенеді
Correct answer #0 наносекундпен
Correct answer #1 nanosec
Wrong answer #0 миллисекундпен
Wrong answer #1 секундтармен
Wrong answer #2 мегагерцпен
Wrong answer #3 гигагерцпен

id: 3) Операциялық жүйелер құрамына мыналар кіреді: Complexity : Easy

0) деректер базасын басқару жүйелері
1) бағдарламалау жүйелері
2) жүйелік бағдарламалық қамтамасыз ету
3) бірегей бағдарламалық қамтамасыз ету
4) қолданбалы бағдарламалық қамтамасыз ету

Correct answer count : 1
Choose answer index : # 1: 4

Your answers : қолданбалы бағдарламалық қамтамасыз ету
Correct answers : жүйелік бағдарламалық қамтамасыз ету

Your score : 0/1

id: 0) Best programming languages? Complexity : Difficult

0) Rust
1) Scratch
2) Python
3) Brainfuck
4) C++
5) Java

Correct answer count : 4
Choose answer index : # 1:
```

Сурет -5

```
MINGW64:d/C++/quiz_cpp
Correct answers : жүйелік бағдарламалық қамтамасыз ету

Your score : 0/1

id: 0) Best programming languages? Complexity : Difficult

0) Rust
1) Scratch
2) Python
3) Brainfuck
4) C++
5) Java

Correct answer count : 4
Choose answer index : # 1: 1
Choose answer index : # 2: 1
Choose answer index : # 3: 1
Choose answer index : # 4: 1

Your answers : Scratch
Correct answers : C++ Java Rust Python

Your score : 0/4

id: 4) WINDOWS операциялық жүйесі келесі функцияны орындамайды Complexity : Medium

0) бірнеше қосымшалардың бір уақытта жұмыс істеуін қамтамасыз етеді
1) компьютердің аппараттық құралдарын басқару
2) бағдарламалық жүйелерді құру
3) қолданбалы бағдарламаларды іске қосу
4) файлдық жүйемен жұмыс істеуді қамтамасыз ету

Correct answer count : 1
Choose answer index : # 1:
```

Қорытынды:

C++ тілін үйрену өте қызықты болды. Мен деректердің негізгі түрлерін, операторларды және басқару құрылымдарын зерттей отырып, негіздерден бастадым. Бірте-бірте функциялар және оларды бағдарламаны басқару үшін пайдалану сияқты күрделі ұғымдарға көштім.

Ең қызықты аспектілердің бірі құрылымдар мен сыныптарды зерттеу болды. Өз деректер түрлерін құруды, олармен жұмыс істеу әдістерін қосуды және кодты ұйымдастыру үшін мұраны пайдалануды білдім.

Менің зерттеуім деректерді енгізу және шығару жұмысын да қамтыды. Мен консольден ақпаратты оқуды, оны файлдарға жазуды және керісінше үйрендім. Бұл дағдылар пайдаланушымен деректер алмасуды немесе жұмыс нәтижелерін сақтауды қажет ететін бағдарламаларды құруда маңызды болды.

Әрине, мен кесте туралы да ұмытқан жоқпын. Тиісті кітапханаларды қолдана отырып, қарапайым графикалық интерфейстерді құруға және деректерді визуализациялауға тәжірибе жасадым. Бұл тілді үйренуге қызығушылықтың қосымша деңгейін қосты және көрнекі түрде тартымды нәрсе жасай отырып, білімімді іс жүзінде қолдануға мүмкіндік берді.

Пайдаланылған әдебиеттер:

<https://kk.wikipedia.org/wiki/C%2B%2B>

<https://martebe.kz/s-ba-darlamalau-tili/>

Қосымша сабақтағы есептер:

Есеп 1:

Қатарда қайталанатын символдарды санымен бірге шығару

```
#include <iostream>
#include <string>
#include <vector>

int count_chars(const std::string &str, char ch){
    int result = 0;
    for(const auto &e : str)
        if(e == ch) result++;
    return result;
}

int find_char(const std::vector<char> &v, char ch){
    int result = 0;
    for(const auto &e : v)
        if(e == ch) return true;
}
```

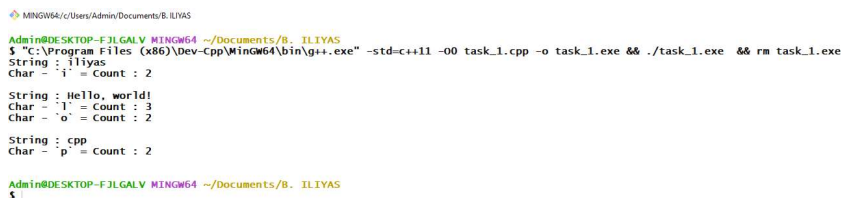
```

        return false;
    }

int main()
{
    std::vector<std::string> strs = {"iliyas", "Hello, world!", "cpp"};
    for(const auto &str : strs){
        std::cout << "String : " << str << std::endl;
        std::vector<char> alreadyMet;
        for(const auto &ch : str){
            int ch_count = count_chars(str, ch);
            if (ch_count > 1 and !find_char(alreadyMet, ch)){
                std::cout << "Char - `" << ch << "` = Count : " << ch_count << std::endl;
                alreadyMet.push_back(ch);
            }
        }
        std::cout << std::endl;
    }
    return 0;
}

```

Нәтижесі:



```

MINGW64/c:/Users/Admin/Documents/B. ILIYAS
Admin@DESKTOP-F3JLGALV MINGW64 ~/Documents/B. ILIYAS
$ "C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\g++.exe" -std=c++11 -O0 task_1.cpp -o task_1.exe && ./task_1.exe && rm task_1.exe
String : iliyas
Char - 'i' = Count : 2

String : Hello, world!
Char - 'l' = Count : 3
Char - 'o' = Count : 2

String : cpp
Char - 'p' = Count : 2

Admin@DESKTOP-F3JLGALV MINGW64 ~/Documents/B. ILIYAS
$

```

Есеп 2:

Дүкенде аты-жөні, сатып алушының үй мекен-жайы және берілген жеңілдік мөлшері бар тұрақты клиенттердің тізімі жасалады. Сатып алушының аты-жөні мен үй мекенжайын тексеру арқылы осы тізімнен барлық қайталанатын жазбаларды жойыңыз.

```

#include <iostream>
#include <vector>

namespace Shop
{
    class Customer
    {
    public:
        std::string m_name;
        std::string m_address;
        int m_discount;

        Customer(std::string name, std::string address, int discount) : m_name(name),
m_address(address), m_discount(discount) {}

        bool operator==(const Customer &another)
        {
            if (m_name == another.m_name)
                return true;
            else if (m_address == another.m_address)
                return true;

```

```

        else
            return false;
    }
    friend std::ostream &operator<<(std::ostream &os, const Customer &c)
    {
        std::cout << "Customer {\n\tName : " << c.m_name << "\n\tAddress : " << c.m_address <<
"\n\tDiscount : " << c.m_discount << "\n}";
    }
};

void rmSimilar(std::vector<Shop::Customer> &v)
{
    int rmSize = 0;
    for (int i = 0; i < v.size() - rmSize; ++i)
        for (int j = i + 1; j < v.size() - rmSize; ++j)
            if (v[i] == v[j])
            {
                Customer temp = v[j];
                v[j] = v[v.size() - 1 - rmSize];
                v[v.size() - 1 - rmSize] = temp;
                rmSize++;
                --j;
            }
    v.erase(v.begin() + (v.size() - rmSize), v.end());
}

template <typename T>
void printVector(const std::vector<T> &v)
{
    std::cout << "Vector : " << std::endl;
    for (const auto &e : v)
    {
        std::cout << e << std::endl;
    }
    std::cout << std::endl;
}

int main()
{
    // Бірдей адресстер немесе есімдер жоқ. Экранға барлығы шығарылады
    std::vector<Shop::Customer> testRegularCustomers1 = {
        Shop::Customer{
            "Zhantore",
            "Taraz, mkr.10",
            10},
        Shop::Customer{
            "Magzhan",
            "Taraz, mkr.5",
            25},
        Shop::Customer{
            "Iliyas",
            "Hostel, TarGU",
            100},
    };
    Shop::rmSimilar(testRegularCustomers1);
    printVector(testRegularCustomers1);

    // Екі Жантөре деген есім бар. Екіншісі өшіріледі. Экранға индексі 2 элементтен басқалары
    шығарылады
    std::vector<Shop::Customer> testRegularCustomers2 = {
        Shop::Customer{
            "Zhantore",

```



```

        "Taraz, mkr.10",
        10},
Shop::Customer{
    "Magzhan",
    "Taraz, mkr.5",
    25},
Shop::Customer{
    "Zhantore",
    "Taraz, mkr.10",
    15},
Shop::Customer{
    "Iliyas",
    "Hostel, TarGU",
    100},
};
Shop::rmSimilar(testRegularCustomers2);
printVector(testRegularCustomers2);

// Үш Жантөре деген есім бар. Екіншісі және үшіншісі өшіріледі. Экранға индексі 1-2 элементтен
басқалары шығарылады
std::vector<Shop::Customer> testRegularCustomers3 = {
    Shop::Customer{
        "Zhantore",
        "Taraz, mkr.10",
        10},
    Shop::Customer{
        "Zhantore",
        "Taraz, mkr.10",
        15},
    Shop::Customer{
        "Zhantore",
        "Taraz, mkr.10",
        35},
};
Shop::rmSimilar(testRegularCustomers3);
printVector(testRegularCustomers3);

// Екі Жантөре деген есім бар. Және бір Жантөренің адресінде тұратын досы бар. Экранға индексі
2-3 элементтен басқалары шығарылады
std::vector<Shop::Customer> testRegularCustomers4 = {
    Shop::Customer{
        "Zhantore",
        "Taraz, mkr.10",
        10},
    Shop::Customer{
        "Iliyas",
        "Hostel, TarGU",
        35},
    Shop::Customer{
        "Zhantorenin dosy",
        "Taraz, mkr.10",
        10},
    Shop::Customer{
        "Zhantore",
        "Taraz, mkr.10",
        15},
};
Shop::rmSimilar(testRegularCustomers4);
printVector(testRegularCustomers4);
}

```


Нәтижесі:

```
MINGW64/c/Users/Admin/Documents/B. ILIYAS
Admin@DESKTOP-FJLGALV MINGW64 ~/Documents/B. ILIYAS
$ "C:\Program Files (x86)\Dev-cpp\MinGW64\bin\g++.exe" -std=c++11 -O0 task_2.cpp -o task_2.exe && ./task_2.exe
&& rm task_2.exe
Vector :
Customer {
    Name : Zhanore
    Address : Taraz, mkr.10
    Discount : 10
};
Customer {
    Name : Magzhan
    Address : Taraz, mkr.5
    Discount : 25
};
Customer {
    Name : Iliyas
    Address : Hostel, TarGU
    Discount : 100
};
Vector :
Customer {
    Name : Zhanore
    Address : Taraz, mkr.10
    Discount : 10
};
Customer {
    Name : Magzhan
    Address : Taraz, mkr.5
    Discount : 25
};
Customer {
    Name : Iliyas
    Address : Hostel, TarGU
    Discount : 100
};
Vector :
Customer {
    Name : Zhanore
    Address : Taraz, mkr.10
    Discount : 10
};
Vector :
Customer {
    Name : Zhanore
    Address : Taraz, mkr.10
    Discount : 10
};
Customer {
    Name : Iliyas
    Address : Hostel, TarGU
    Discount : 35
};
Admin@DESKTOP-FJLGALV MINGW64 ~/Documents/B. ILIYAS
$
```

Есеп 3:

Number1.txt -100-ден 100-ге дейінгі диапазонға жататын кездейсоқ бүтін сандармен бағдарламалық түрде толтырылады. Файлдағы сандар саны пернетақтадан беріледі. Әр жолдағы сандар саны 0-ден 10-ға дейінгі диапазондағы ерікті сан болып табылады және rand () функциясы арқылы беріледі. Number1 файлына сәйкес жолдарға бөлуді сақтай отырып, сандарды консольге шығарыңыз.

```
#include <fstream>
#include <iostream>
#include <random>
#include <ctime>

const std::string FILENAME = "input_task3.txt";

namespace fileStream{
    std::string readToStr(const std::string &filename){
        std::ifstream iFile(FILENAME);
        std::string fileData;
        while(!iFile.eof()){
            fileData.push_back(iFile.get());
        }
        //std::cout << fileData;
        iFile.close();
        return fileData;
    }
}

int main(){
    srand(time(0));
    int numCount, numInStr;
    std::cout << "Сандар : ";
```

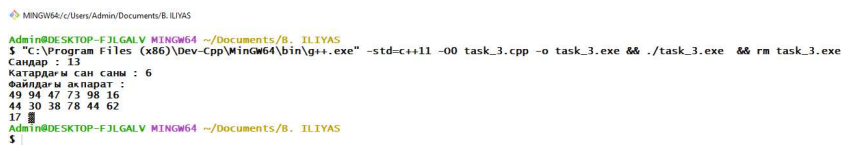
```

std::cin >> numCount;
numInStr = (rand() % 10);
std::cout << "Қатардағы сан саны : " << numInStr << std::endl;
std::ofstream oFile(FILENAME);
for(int i = 1; i <= numCount; ++i){
    oFile << rand() % 100 << " ";
    if(i % numInStr == 0)
        oFile << "\n";
}
oFile.close();

std::cout << "Файлдағы ақпарат : " << std::endl;
std::cout << fileStream::readToStr(FILENAME);
}

```

Нәтижесі:



```

MINGW64/c/Users/Admin/Documents/B. ILIYAS
Admin@DESKTOP-F3JGALV MINGW64 ~/Documents/B. ILIYAS
$ "C:\Program Files (x86)\dev-cpp\bin\g++.exe" -std=c++11 -O0 task_3.cpp -o task_3.exe && ./task_3.exe && rm task_3.exe
Сандар : 13
Қатардағы сан саны : 6
Файлдағы ақпарат :
49 94 47 73 98 16
44 30 38 78 44 62
17
Admin@DESKTOP-F3JGALV MINGW64 ~/Documents/B. ILIYAS
$

```

Есеп 4:

Спираль схемасы бойынша бүтін сандардың өсіп келе жатқан тізбегімен n өлшемді квадрат матрицаны толтыру функциясын жазыңыз. Матрицаның өлшемі тұрақты болып. Матрицаны экранға шығару.

```

#include <iostream>
#include <vector>

std::vector<int> getPartSize(int n){
    std::vector<int> result = {n, };
    for(int i = n - 1; i > 0; --i){
        result.push_back(i);
        result.push_back(i);
    }
    return result;
}

enum Direction{
    UP,
    DOWN,
    RIGHT,
    LEFT,
};

std::vector< std::pair<int, Direction> > toDirectionPair(const std::vector<int> &v){
    std::vector<Direction> baseDirection = {RIGHT, DOWN, LEFT, UP};
    std::vector< std::pair<int, Direction> > result;
    for(int i = 0, di = 0; i < v.size(); ++i, ++di){
        if(di >= baseDirection.size()) di = 0;
        result.push_back(std::pair<int, Direction>(v[i], baseDirection[di]));
    }
    return result;
}

```

```

template <typename T>
void printVec(const std::vector<T> &v){
    for(const auto &e : v)
        std::cout << e << " ";
    std::cout << std::endl;
}

std::vector<std::vector<int>> initQuadMat(int size){
    std::vector<std::vector<int>> result;
    for(int i = 0; i < size; ++i){
        result.push_back(std::vector<int>(size));
    }
    return result;
}

void printQuadMat(const std::vector<std::vector<int>> &m){
    for(const auto &v : m){
        printVec(v);
    }
}

int main(){
    int matSize;
    std::cout << "Квадрат матрица көлемі: ";
    std::cin >> matSize;
    auto quadMat = initQuadMat(matSize);
    printQuadMat(quadMat);
    auto directions = toDirectionPair(getPartSize(matSize));
    for(const auto &d : directions){
        std::cout << d.first << " ";
        switch(d.second){
            case Direction::DOWN: std::cout << "Down" << std::endl; break;
            case Direction::UP: std::cout << "Up" << std::endl; break;
            case Direction::RIGHT: std::cout << "Right" << std::endl; break;
            case Direction::LEFT: std::cout << "Left" << std::endl; break;
        }
    }

    int matElem = 1;
    int currX = -1, currY = 0;
    for(const auto &d : directions){
        for(int i = 0; i < d.first; ++i){
            switch (d.second)
            {
                case Direction::DOWN: currY++; break;
                case Direction::UP: currY--; break;
                case Direction::RIGHT: currX++; break;
                case Direction::LEFT: currX--; break;
            }
            quadMat[currY][currX] = matElem++;
        }
    }
    printQuadMat(quadMat);
}

```

Нәтижесі:

Admin@DESKTOP-FJLGALV MINGW64 ~/Documents/B. ILIYAS

```
$ "C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\g++.exe" -std=c++11 -O0 task_4.cpp -o task_4.exe && ./task_4.exe && rm task_4.exe
```

Квадрат матрица ke-newi: 7

```
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

7 Right

6 Down

6 Left

5 Up

5 Right

4 Down

4 Left

3 Up

3 Right

2 Down

2 Left

1 Up

1 Right

```
1 2 3 4 5 6 7
24 25 26 27 28 29 8
```

```
23 40 41 42 43 30 9
```

```
22 39 48 49 44 31 10
```

```
21 38 47 46 45 32 11
```

```
20 37 36 35 34 33 12
```

```
19 18 17 16 15 14 13
```

Admin@DESKTOP-FJLGALV MINGW64 ~/Documents/B. ILIYAS

\$