Дано поле 10 х 10 позиций. Нумерация позиций по горизонтали и по вертикали начинается с 1. Поле 10 х 10 формируется в файле field.txt и первоначально все позиции заполняются 8.  
На вход подаются последовательно данные: координата позиции и символ для отметки.  
Координата позиции задается посредством пары целых чисел от 1 до 10. Первое число номер строки, второе номер столбца.  
Начало координат верхний левый угол поля. Символ для отметки принадлежит латинскому алфавиту.  
Таких троек данных может быть множество. После получения очередной тройки **выдается сигнал** с текстом, содержащим координату и символ латинского алфавита.  
Если координата равна ( 0, 0 ), программа завершает работу.  
Первая строка входных данных не содержит ( 0, 0 ).  
  
  
Использовать объекты:  
1. Для установки позиции в файле согласно координате. Объект **выдает сигнал** об установке позиции или выводит в конце файла сообщение об ошибке.  
2. Для записи символа в файле в установленную позицию. Объект пишет символ в установленной позиции или выводит в конце файла сообщение об ошибке.  
3. Для вывода результата из файла на консоль. Все сообщения об ошибках пишутся построчно, с новой строки. Первое сообщение выводится с 11 строки (после поля 10 х 10).  
  
  
Написать программу, реализующую следующий алгоритм:  
1. Создание файла и формирование исходного содержания.  
2. Ввод первой тройки данных.  
3. Начало цикла.  
    3.1. Выдача сигнала о вводе данных.  
    3.2. Ввод очередной тройки данных и выдача сигнала.  
4. Завершение цикла, если введена координата ( 0, 0 ).  
5. Вывод на консоль содержимого файла.

**Входные данные**

Построчно множество координат и символ:  
«целое число» «целое число» «символ»

**Выходные данные**

Десять строк по десять символов в каждой, согласно сформированному в файле полю.

Если какая-либо координата не принадлежит интервалу [ 1, 10 ] то выводится сообщение:  
Coordinate is wrong ( «значение», «значение» )  
  
Если символ не принадлежит латинскому алфавиту:  
Not a letter of the Latin alphabet: «символ»

Используя потоки Ввода/Вывода - cin/cout

Используя void buildFile для строить фаил

Используя void createNewFile() для создать новый фаил

Используя void scanPosition() для обрабатывать координаты

Используя int exec\_app() для показать фаил

Используя void printToScreen() для применять

Используя void setPosition(int x, int y, char k) для установки позиции в файле согласно координате.

Используя void insertChar(int x, int y) и void insertChar(char k, bool letter) для записи символа в файле в установленную позицию.

Используя void showFile() для вывода результата из файла на консоль.

int main()

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Предикат | Действие | № перехода |
| 1 |  | cl\_application file\_application; file\_application.buildFile(); return file\_application.exec\_app(); | Ø |

void cl\_base::setNameFile(string nameFile)

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Предикат | Действие |  |
| 1 |  | this->nameFile = nameFile; | Ø |

string cl\_base::getNameFile()

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 |  | return this->nameFile; | Ø |

void cl\_base::setPosition(int x, int y, char k)

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Предикат | Действие | № перехода |
| 1 | if (!((1 <= x && x <= 10)&&(1 <= y && y <= 10))) | addDataFile.open(nameFile, ios::app); insertChar(x, y); | Ø |
| else |  | 2 |
| 2 | if (!(('a' <= k && k <= 'z')||('A' <= k && k <= 'Z'))) | addDataFile.open(nameFile, ios::app); insertChar(k, false); | Ø |
| else | addDataFile.open(nameFile, ios::in); int position = (x-1)\*11 + y -1; addDataFile.seekp(position, ios::beg); insertChar(k, true); | Ø |

void cl\_base::insertChar(int x, int y)

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Предикат | Действие | № перехода |
| 1 |  | addDataFile << endl << "Coordinate is wrong ( " << x << ", " << y << " )";  addDataFile.close(); | Ø |

void cl\_base::insertChar(char k, bool letter)

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Предикат | Действие | № перехода |
| 1 | if (letter) | addDataFile.put(k); addDataFile.close(); | Ø |
| else | addDataFile << endl << "Not a letter of the Latin alphabet: " << k;  addDataFile.close(); | Ø |

void cl\_base::showFile()

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Предикат | Действие | № перехода |
| 1 |  | readFile.open(nameFile, ios::in); | 2 |
| 2 | if (!readFile) | return; | Ø |
| else |  | 3 |
| 3 |  | string line; | 4 |
| 4 | while(getline(readFile, line)) | cout << line; | 5 |
| 0 |  | 6 |
| 5 | if (!readFile.eof()) | cout << endl; | 4 |
| else |  | 6 |
| 6 |  | readFile.close(); | Ø |

cl\_application::cl\_application()

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 |  | string name ="file.txt";  toDo.setNameFile(name); | Ø |

void cl\_application::buildFile()

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 |  | createNewFile();  scanPosition(); | Ø |

void cl\_application::createNewFile()

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 |  | ofstream newFile(toDo.getNameFile(), ios::out); | 2 |
| 2 | if (!newFile) | return; | Ø |
| else |  | 3 |
| 3 | for (int indexR = 1; indexR <= 10; indexR++) |  | 4 |
| indexR > 10; |  | 6 |
| 4 | if (indexR != 1) | newFile << endl; | 5 |
| else |  | 5 |
| 5 | for (int indexC = 1; indexC <= 10; indexC++) | int number = 8; newFile << number; | 5 |
| indexC > 10 |  | 3 |
| 6 |  | newFile.close(); | Ø |

void cl\_application::scanPosition()

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 | while(true) | cin >> x >> y; | 2 |
| 2 | if (x == 0 && y == 0) | break; | Ø |
| else | char k; cin >>k; toDo.setPosition(x, y, k); | 1 |

int cl\_application::exec\_app()

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 |  | printToScreen(); return 0; | Ø |

void cl\_application::printToScreen()

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 |  | toDo.showFile(); | Ø |