struct rectangle

{ int length;

double width;

}

rectangle brick={10,13.4};

Структура - это совокупность переменных, объединенных одним именем, предоставляющая общепринятый способ совместного хранения информации.

Для объявления массива структур следует сначала определить структуру, а затем объявить массив переменных данного типа. Например, для объявления 100-элементного массива структур типа addr следует написать:  
  
struct addr addr\_info[100];  
  
В результате получаем набор из 100 переменных, устроенных, как объявлено в типе структуры addr.

2. Очередь – это динамическая структура данных которая состоит из набора элементов которые размещены последовательно друг за другом. При этом добавление элементов осуществляется с одной стороны, а удаление (вытягивание) – с другой стороны.  
Очередь работает по принципу FIFO (First In — First Out), то есть «первым пришел – первым вышел».

**Виды очередей**

* простая очередь;
* кольцевая очередь. В такой очереди элемент, который выходит с начала очереди, будет помещен в ее конец;
* очередь с приоритетами. В такой очереди элементы размещаются по их приоритетам (весовым коэффициентам). Первым из очереди выходит элемент с наивысшим приоритетом.

#include <iostream>

#include <queue>  // подключили библиотеку queue

using namespace std;

int main() {

  setlocale(LC\_ALL,"rus");

  queue <int> q;  // создали очередь q

  cout << "Пользователь, пожалуйста введите 7 чисел: " << endl;

  for (int h = 0; h < 7; h++) {

    int a;

    cin >> a;

    q.push(a);  // добавляем в очередь элементы

  }

  cout << endl;

  cout << "Самый первый элемент в очереди: " << q.front() << endl;  // выводим первый

                                                                   // элемент очереди

  q.pop();  // удаляем элемент из очереди

  cout << "Новый первый элемент (после удаления): " << q.front() << endl;

  if (!q.empty()) cout << "Очередь не пуста!";  // проверяем пуста ли очередь (нет)

  system("pause");

  return 0;

}

3. #include <iostream>

#include <windows.h>

using namespace std;

int main()

{

system("color 0a");

setlocale(LC\_ALL, "Rus");

char str[100] = { 0 };

cout << "Введите латиницу и кириллицу ";

cin.getline(str, 100);

OemToCharA(str, str);

char x[] = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЪЭЮЯ";

char x1[] = "абвгдеёжзиёклмнопрстуфхцчшщьыъэюя"; // Создал для дальнейшего использования

for (int i = 0; i < strlen(str); i++)

{

if (str[i] >= 'А' && str[i] <= 'Я')

{

for (int j = 0; j < strlen(str);j++)

{

if (str[i] == x[j])

{

\*\*\*\*\* //Нужно присвоить str[i] число j

};

}

}

}

cout << str << endl;

system("pause");

return 0;

}

4. ----

5. #include <iostream>

void ShowArray(int \* arr, int n);

void InputArray(int \* arr, int n);

void ModifyArray(int \* arr, int n);

int main()

{

    setlocale(LC\_ALL, "Russian");

    int size = 0;

    std::cout << "Ввод кол-ва элементов: ";

    std::cin >> size;

    if (size <= 0)

        return 0;

    int \* arr = new int[size];

    InputArray(arr, size);

    std::cout << "**\n**Исходный массив: ";

    ShowArray(arr, size);

    ModifyArray(arr, size);

    std::cout << "**\n**Массив после обработки: ";

    ShowArray(arr, size);

    std::cout << std::endl;

    delete[] arr;

    return 0;

}

void ModifyArray(int \* arr, int n)

{

    int temp = 0;

    for (int i = 0; i < (n / 2); i++)

    {

        temp = arr[n - i - 1];

        arr[n - i - 1] = arr[i];

        arr[i] = temp;

    }

}

void ShowArray(int \* arr, int n)

{

    for (int i = 0; i < n; i++)

        std::cout << arr[i] << " ";

}

void InputArray(int \* arr, int n)

{

    std::cout << "Ввод массива:" << std::endl;

    for (int i = 0; i < n; i++)

    {

        std::cout << "Ввод элемента № " << i + 1 << " = ";

        std::cin >> arr[i];

    }

}