//task

//work with array

//auto keyword

//recursiya

//quick sort view

//template<typename T>

//void initArray(T arr[], int size) {

// int min = 10;

// int max = 1000;

// T random = 0;

// for (size\_t i = 0; i < size; i++)

// {

// random = min + rand() % (max - min);

// arr[i] = random / 10;

// }

//}

//

//void initArray(char arr[], int size) {

// int min = 65;

// int max = 90;

// int random = 0;

// for (size\_t i = 0; i < size; i++)

// {

// random = min + rand() % (max - min);

// arr[i] = random;

// }

//}

//

//

//template<typename T>

//void printArray(T arr[], int size) {

// for (size\_t i = 0; i < size; i++)

// {

// cout << arr[i] << " ";

// }cout << endl;

//}

//

//

//void main() {

// const int size = 10;

// double arr[size] = {};

// initArray(arr, size);

// printArray(arr, size);

//

//

//

//}

//template<typename HakunaMatata>

//void AddNumbers(HakunaMatata data1, HakunaMatata data2) {

// auto result = data1 + data2;

// cout << result << endl;

//}

//void show(int number) {

// if (number == 0) {

// return;

// }

// cout << number << endl;

// show(number - 1);

//}

//

//

//void main() {

// show(10);

//}

//int fact(int number) {

// if (number == 0) {

// return 1;

// }

// return number \* fact(number - 1);

//}

//

//

//void main() {

//

// cout << fact(10) << endl;

//}

//void Show(int end) {

// if (end == 0) {

// return;

// }

// Show(end - 1);

// cout << end << endl;

//}

//void main() {

//

// Show(10);

//

//}