#include<iostream>

using namespace std;

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

// int min = 1;

// int max = 100;

// int random = 0;

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

// {

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

// arr[i] = random;

// }

//}

//

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

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

// {

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

// }cout << endl;

//}

//

//const int mysize = 10;

//void slideArray(int arr[], int index) {

// int newarray[mysize] = {};

// int i2 = 0;

// for (size\_t i = mysize-index; i < mysize; i++,i2++)

// {

// newarray[i2] = arr[i];

// }

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

// {

// newarray[i2] = arr[i];

// ++i2;

//

// }

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

// {

// arr[i] = newarray[i];

// }

//

//}

//

//void main() {

// /\*

// 8. Massivin funksiyaya ötürülən rəqəm qədər sağa dövrü sürüşməsini

//yerinə yetirən funksiya yazın. (Funksiyaya massiv, ölçüsü, və

//sürüşmə indeksi göndərilir) Məsələn massiv : 1,2,3,4,5 rəqəmləri ilə

//dolurudur , sürüşmə indexi 2dirsə, cavab : 4,5,1,2,3

// \*/

// //int arr[mysize] = {};

// //initArray(arr, mysize);

// //printArray(arr, mysize);

// //slideArray(arr, 2);

// //printArray(arr, mysize);

//

//

//}

//template<typename T>

//void addNumbers(T n1, T n2) {

// cout << n1 + n2 << endl;

//}

//

//template<typename T>

//void addNumbers(T n1, T n2,T n3) {

// cout << n1 + n2 +n3<< endl;

//}

//

//

//void main() {

// addNumbers(10.2f, 20.1f);

//}

//

//template<typename T1,typename T2>

//void MultNumbers(T1 num1, T2 num2) {

// cout << "Result : " << num1 \* num2 << endl;

//}

//

//void main() {

//

// MultNumbers(10, 2.334);

//

//}

//template<typename T>

//T getSum(T num1, T num2) {

// T data = num1 + num2;

// return data;

//}

//template<typename T1,typename T2>

//T1 getSum(T1 num1, T2 num2) {

// T1 data = num1 + num2;

// return data;

//}

//

//void main() {

//

// cout << getSum(10, 20.1f) << endl;

//

//}

//

//

//template<typename T>

//void showData(T data) {

// cout << "general type" << endl;

// cout << "Data is : " << data << endl;

//}

//

//void showData(char data) {

// cout << "CHAR type" << endl;

// cout << "This is character : " << data << endl;

//}

//

//void main() {

// showData<char>('A');

//}

//task

//work with array

//recursiya

//quick sort view

//auto keyword