МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Воронежский государственный технический университет»

**Факультет** экономики, менеджмента и информационных технологий

**Кафедра** систем управления и информационных технологий в строительстве

**Отчет по лабораторной работе**

Тема: **«Динамический массив»**

По дисциплине: Основы программирования и алгоритмизации

Выполнил студент: Гладнева Евгения Геннадьевна

Группа: бИСТ-222

Руководитель: доцент, к.т.н. Курипта О. В.

Работа защищена « » \_\_\_\_\_\_\_\_\_ 2022г.

С оценкой \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Воронеж 2022

**Постановка задачи**

**Условие задачи:** написать программу, которая генерирует 3 массива со случайными вещественными числами и случайным количеством элементов (от 10 до 50). Реализовать функцию, которая возвращает новый массив, в котором чередуются положительные элементы первого массива и отрицательные элементы третьего.

**Исходные данные:**

-

**Алгоритм решения:**

1. Сгенерировать 3 массива.
2. В отдельной функции подсчитать количество положительных элементов в первом массиве и отрицательных в третьем, выделить память под эти два массива и новый массив, равный их сумме.
3. Заполнить массивы + и – чисел.
4. Заполнить новый массив, чередуя числа.
5. Если количество элементов разное, в конце добавить в массив оставшиеся числа.

**Контрольный пример**:

Входные данные: -

Результат: -

**Словесный алгоритм**

**Алгоритм представлен пошаговой детализацией:**

**Шаг 1:** установка директивы препроцессора для функции scanf, подключение необходимых заголовочных файлов и прототипирование функций

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <time.h>

#define XMIN -1000

#define XMAX 1000

double\* add\_arr(double\* array, int size);

void print\_arr(double\* ptr\_array, int n, char c);

double\* func(int\* size4, double\* array1, int size1, double\* array3, int size3);

**Шаг 2:** объявление функции main, установка локализации, настройка генератора рандомных чисел

void main(void) {

setlocale(LC\_ALL, "RUS");

srand((unsigned)time(NULL) / 2);

};

**Шаг 3:** функция заполнения массивов

double\* add\_arr(double\* array, int size){

for (int i = 0; i < size; i++)

array[i] = XMIN + 1.f \* (XMAX - XMIN) \* rand() / RAND\_MAX;

return array;

}

**Шаг 4:** объявление и заполнение массивов

setlocale(LC\_ALL, "RUS");

srand((unsigned)time(NULL) / 2);

int size1 = 10 + rand() % (50 - 10 + 1);

double\* array1 = malloc(size1 \* sizeof(double));

array1 = add\_arr(array1, size1);

int size2 = 10 + rand() % (50 - 10 + 1);

double\* array2 = malloc(size2 \* sizeof(double));

array2 = add\_arr(array2, size2);

int size3 = 10 + rand() % (50 - 10 + 1);

double\* array3 = malloc(size3 \* sizeof(double));

array3 = add\_arr(array3, size3);

**Шаг 5:** объявление и заполнение нового массива

double\* array4;

int size4;

array4 = func(&size4, array1, size1, array3, size3);

**Шаг 6:** функция заполнения нового массива

double\* func(int\* size4, double\* array1, int size1, double\* array3, int size3) {

int count1 = 0, count3 = 0, k = 0;

for (int i = 0; i < size1; i++){

if (array1[i] > 0)

count1++;

}

for (int i = 0; i < size3; i++){

if (array3[i] < 0)

count3++;

}

double\* positive = malloc(count1 \* sizeof(double));

double\* negative = malloc(count3 \* sizeof(double));

for (int i = 0; i < size1; i++){

if (array1[i] > 0)

{

positive[k] = array1[i];

k++;

}

}

k = 0;

for (int i = 0; i < size3; i++){

if (array3[i] < 0)

{

negative[k] = array3[i];

k++;

}

}

double\* array = malloc((count1 + count3) \* sizeof(double));

int min\_len = count1 > count3 ? count3 : count1;

k = 0;

int i = 0;

for (i; i < min\_len; i++, k+=2){

array[k] = positive[i];

array[k + 1] = negative[i];

}

// заполнение хвостами

if (count1 > count3)

{

for (k; k < count1 + count3; i++, k++)

array[k] = positive[i];

}

else

{

for (k; k < count3 + count1; i++, k++)

array[k] = negative[i];

}

\*size4 = count1 + count3;

return array;

}

**Шаг 7:** функция вывода массива

void print\_arr(double\* ptr\_array, int n, char c) {

for (int i = 0; i < n; i++)

printf("%c[%d]: %lf\n", c, i, ptr\_array[i]);

}

**Шаг 8:** вывод созданных массивов

print\_arr(array1, size1, 'A');

printf("\n");

print\_arr(array2, size2, 'B');

printf("\n");

print\_arr(array3, size3, 'C');

printf("\n");

print\_arr(array4, size4, 'D');

**Шаг 9:** освобождение памяти

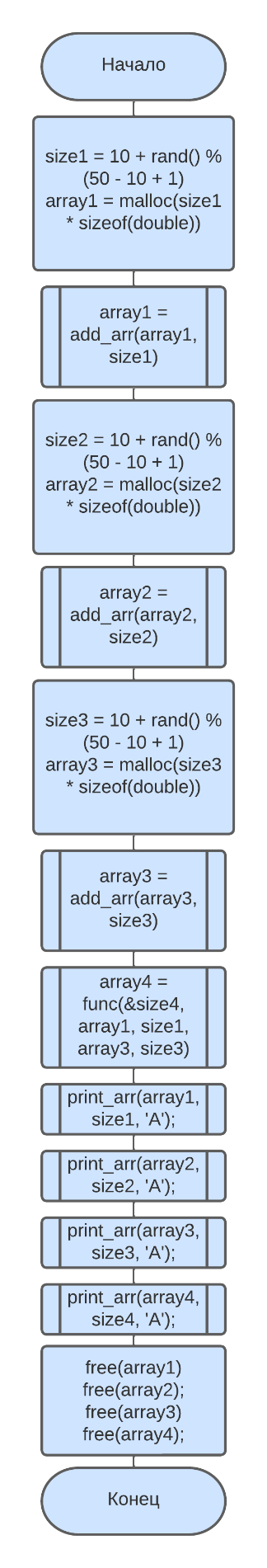
free(array1);

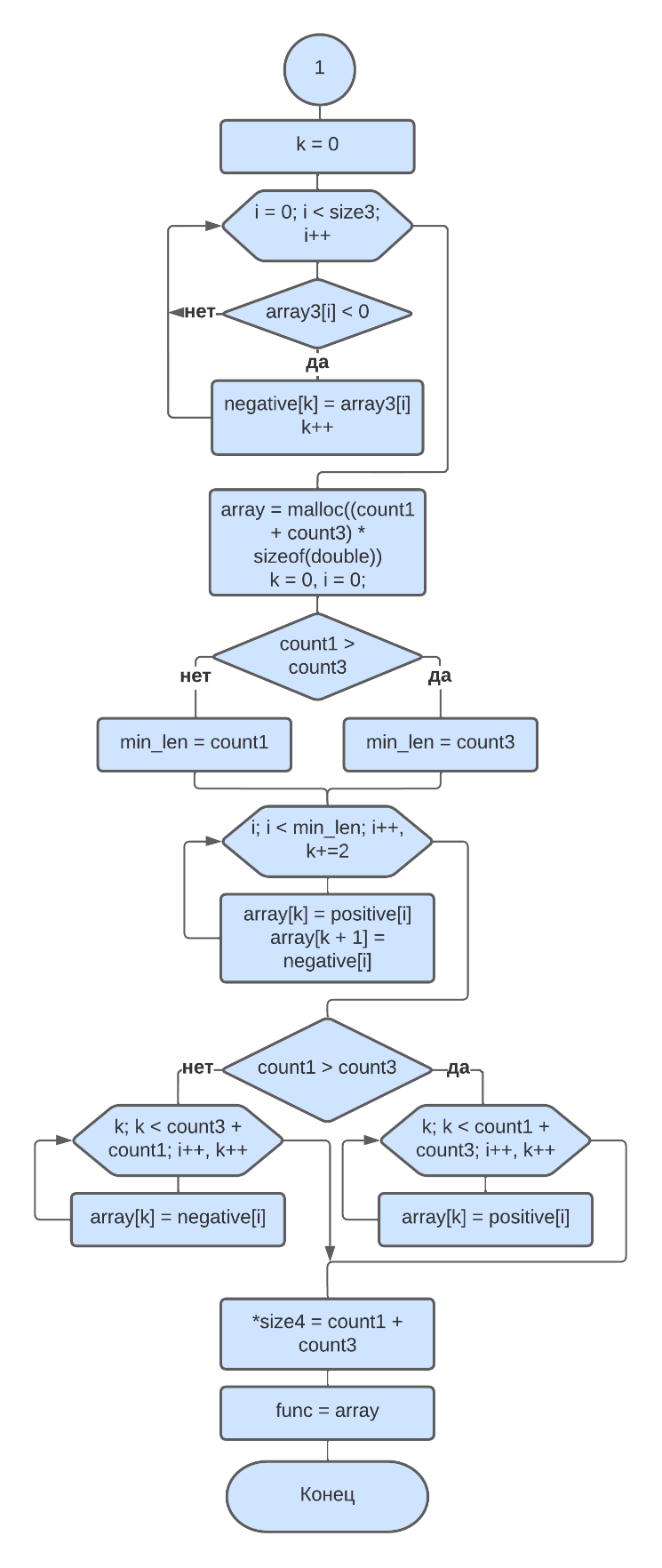
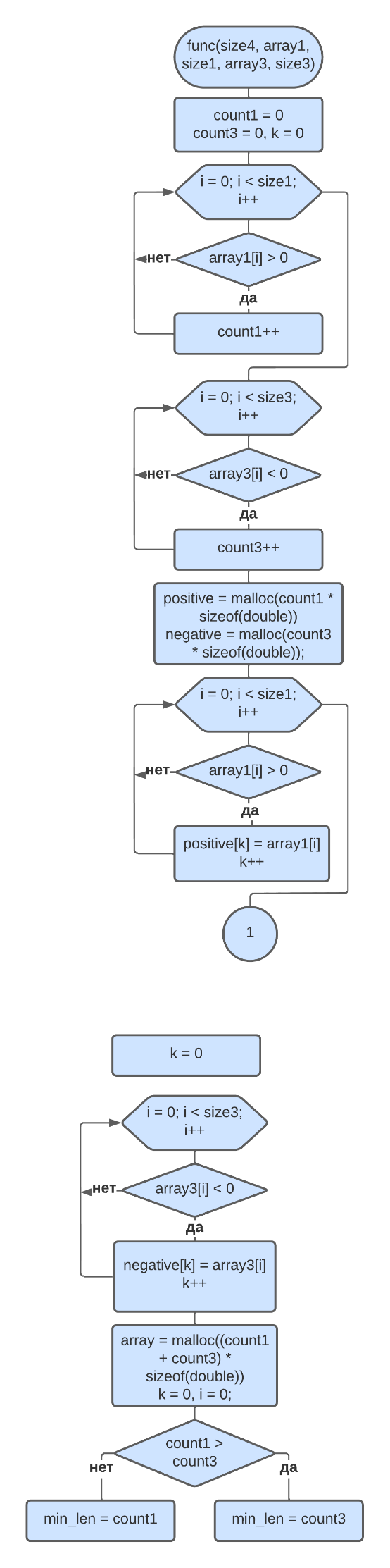
free(array2);

free(array3);

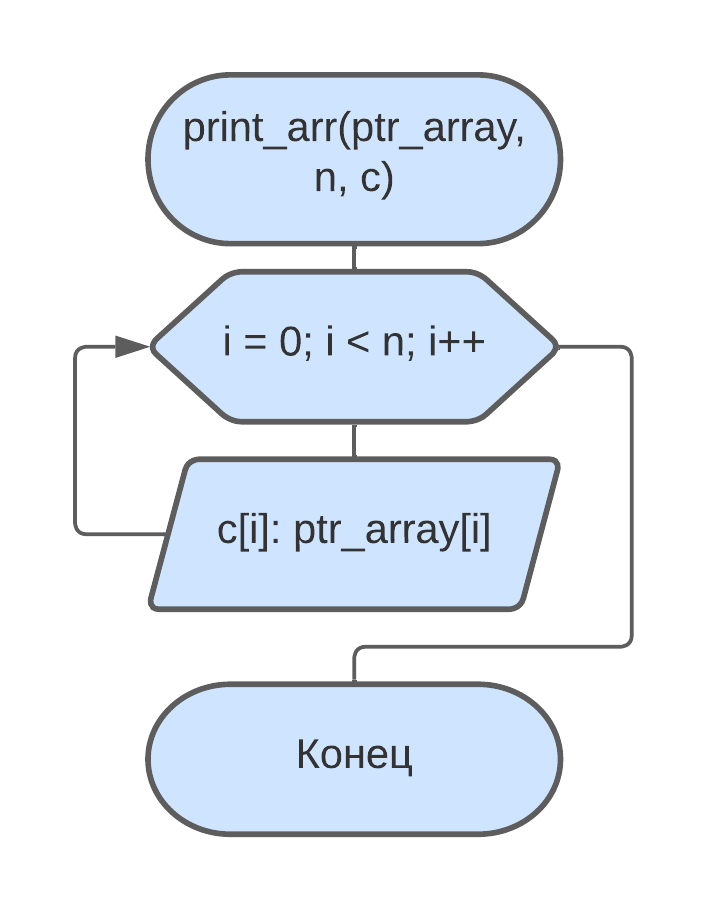
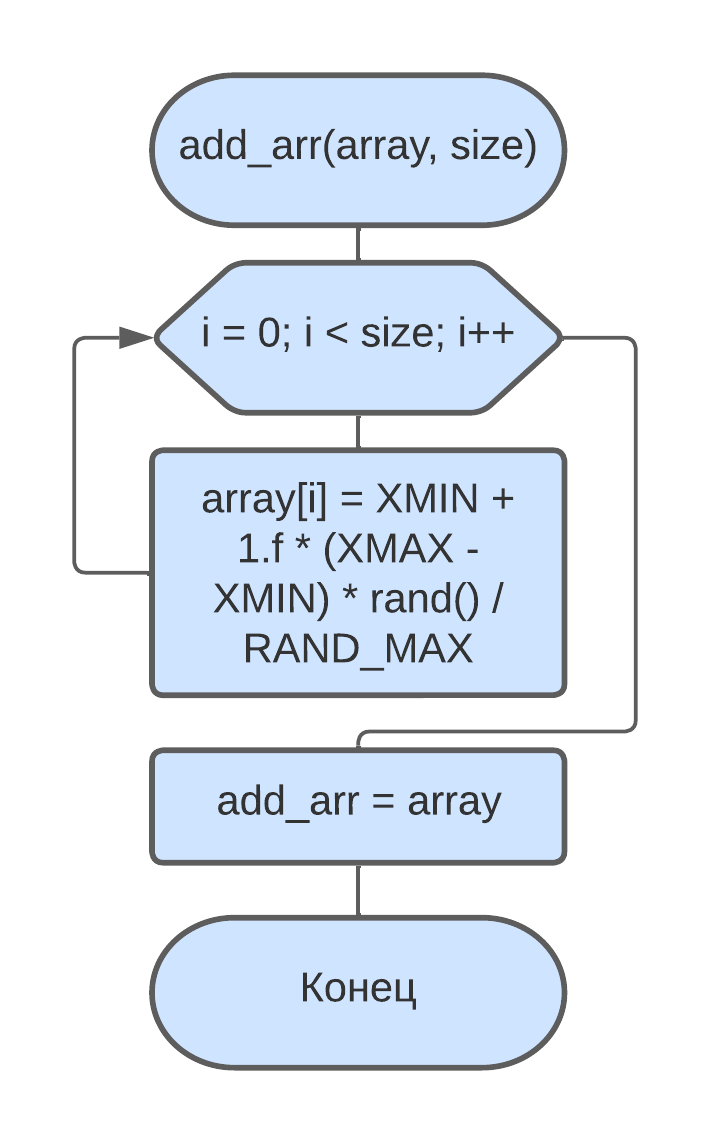
free(array4);

**Блок – схема программы**

Блок – схема программы представлена на рисунке 1.



а) б)



в) г)

а) — блок-схема головного алгоритма;

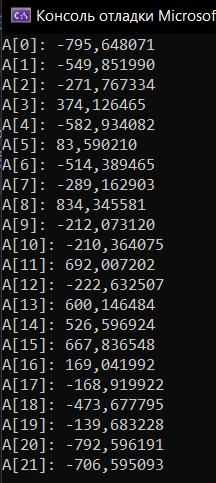
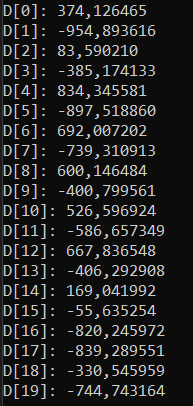
б) — блок-схема функции func;

в) — блок-схема функции add\_arr;

г) — блок-схема функции print\_arr;

Рисунок 1 — Блок-схема программы

**Результат работы программы**

На рисунке 2 представлена работа программы.

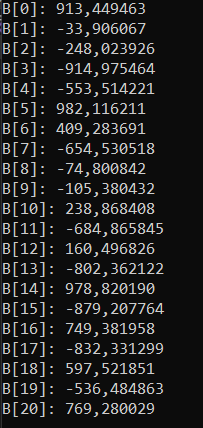
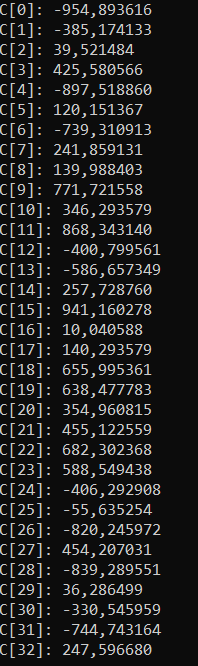


Рисунок 2 — Окно с выполненной программой

**Код программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <time.h>

#define XMIN -1000

#define XMAX 1000

double\* add\_arr(double\* array, int size);

void print\_arr(double\* ptr\_array, int n, char c);

double\* func(int\* size4, double\* array1, int size1, double\* array3, int size3);

void main()

{

setlocale(LC\_ALL, "RUS");

srand((unsigned)time(NULL) / 2);

int size1 = 10 + rand() % (50 - 10 + 1);

double\* array1 = malloc(size1 \* sizeof(double));

array1 = add\_arr(array1, size1);

int size2 = 10 + rand() % (50 - 10 + 1);

double\* array2 = malloc(size2 \* sizeof(double));

array2 = add\_arr(array2, size2);

int size3 = 10 + rand() % (50 - 10 + 1);

double\* array3 = malloc(size3 \* sizeof(double));

array3 = add\_arr(array3, size3);

double\* array4;

int size4;

array4 = func(&size4, array1, size1, array3, size3);

print\_arr(array1, size1, 'A');

printf("\n");

print\_arr(array2, size2, 'B');

printf("\n");

print\_arr(array3, size3, 'C');

printf("\n");

print\_arr(array4, size4, 'D');

free(array1);

free(array2);

free(array3);

free(array4);

}

// заполнение массива случайными числами

double\* add\_arr(double\* array, int size)

{

for (int i = 0; i < size; i++)

array[i] = XMIN + 1.f \* (XMAX - XMIN) \* rand() / RAND\_MAX;

return array;

}

// вывод массива

void print\_arr(double\* ptr\_array, int n, char c)

{

for (int i = 0; i < n; i++)

printf("%c[%d]: %lf\n", c, i, ptr\_array[i]);

}

double\* func(int\* size4, double\* array1, int size1, double\* array3, int size3)

{

int count1 = 0, count3 = 0, k = 0;

for (int i = 0; i < size1; i++)

{

if (array1[i] > 0)

count1++;

}

for (int i = 0; i < size3; i++)

{

if (array3[i] < 0)

count3++;

}

double\* positive = malloc(count1 \* sizeof(double));

double\* negative = malloc(count3 \* sizeof(double));

for (int i = 0; i < size1; i++)

{

if (array1[i] > 0)

{

positive[k] = array1[i];

k++;

}

}

k = 0;

for (int i = 0; i < size3; i++)

{

if (array3[i] < 0)

{

negative[k] = array3[i];

k++;

}

}

double\* array = malloc((count1 + count3) \* sizeof(double));

int min\_len = count1 > count3 ? count3 : count1;

k = 0;

int i = 0;

for (i; i < min\_len; i++, k+=2)

{

array[k] = positive[i];

array[k + 1] = negative[i];

}

// заполнение хвостами

if (count1 > count3)

{

for (k; k < count1 + count3; i++, k++)

array[k] = positive[i];

}

else

{

for (k; k < count3 + count1; i++, k++)

array[k] = negative[i];

}

\*size4 = count1 + count3;

return array;

}