

## 30. Найти объединение двух заданных графов

Алгоритм записи графов:

[https://github.com/Bandix123/sem3/blob/master/DZ2\(graph\).c](https://github.com/Bandix123/sem3/blob/master/DZ2(graph).c)

- 1) Пользователь вводит количество вершин графа
- 2) Пользователь вводит значения в вершинах (проверяется, чтобы не было повторяющихся значений)
- 3) Пользователь вводит соединения вершин

```
First graph:
Write number of nodes
5
Write node's values:
11
22
33
44
55
66

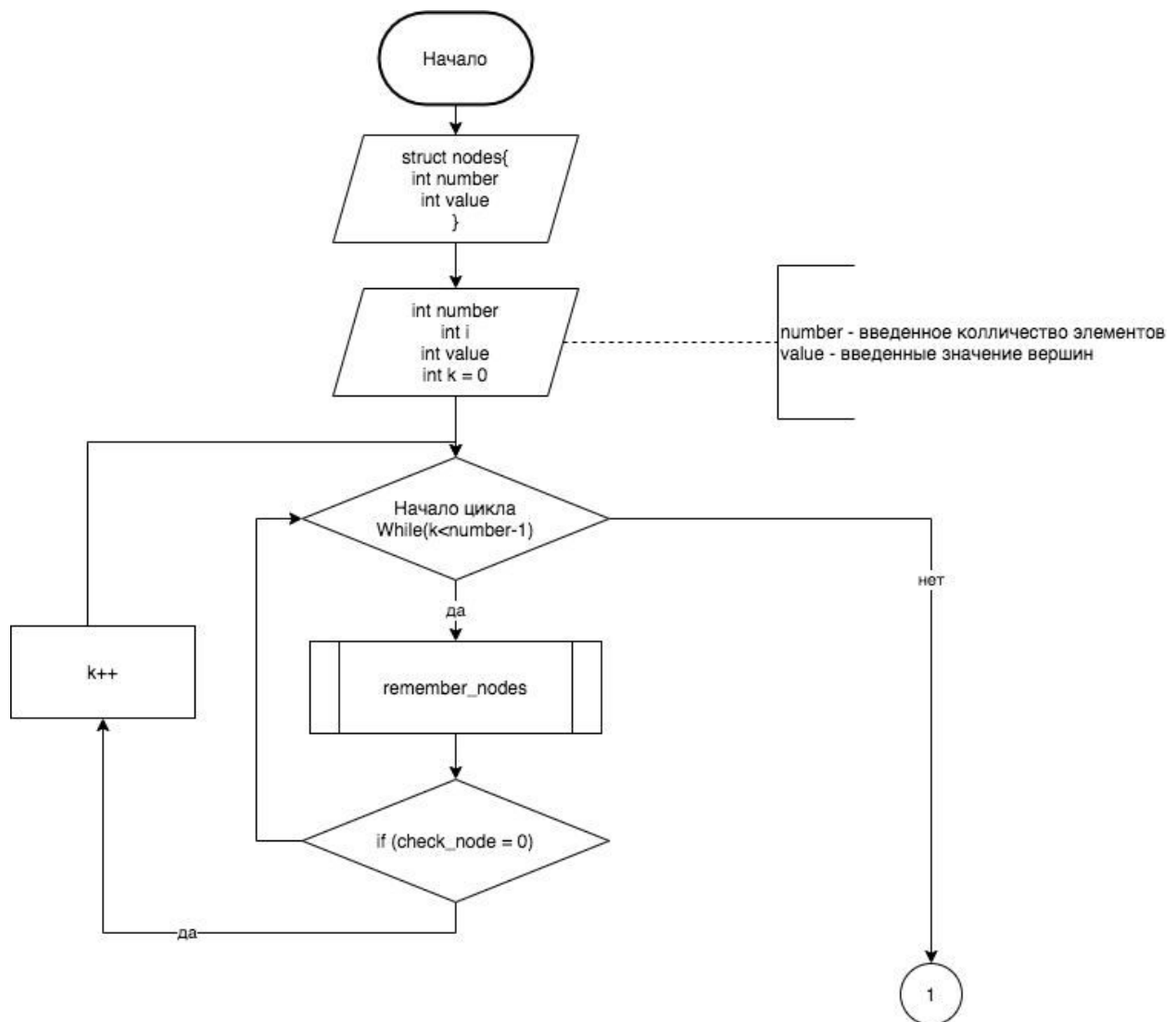
Here Are you node's values:

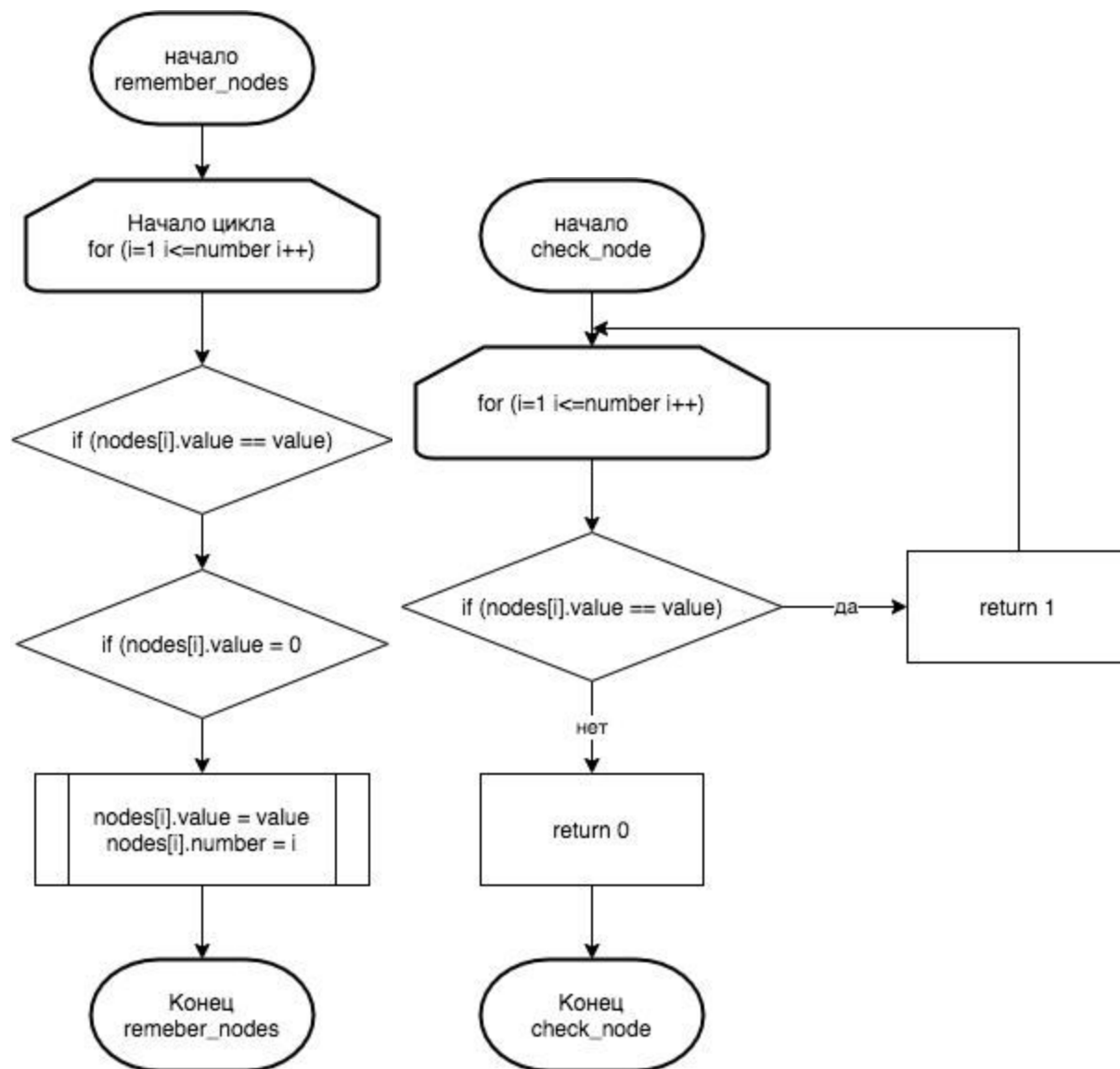
|number|value|
|  1  |  11  |
|  2  |  22  |
|  3  |  33  |
|  4  |  44  |
|  5  |  55  |

Write connections (write 0 to end)
1 2
2 3
3 1
4 2
5 5
1 5
0
```

(Пример ввода пользователем первого графа)

Структурные схемы алгоритмов:





При записи соединений, значения записываются в промежуточный файл cache.txt

```

139 while ((connection1 != 0)&&(connection2 != 0)){
140     scanf("%d %d\n",&connection1,&connection2);
141     print_connection(number,connection1,connection2);
142     k++; if ((connection1 == 0)|| (connection2 == 0)) {
143     }else{
144     // find_value(connection1, connection2,number);
145     for (int i = 1; i <= number; i++)
146         for (int j = 1; j <= number; j++)
147             if ((connection1 == nodes[i].number)&&(connection2 == nodes[j].number)){
148                 connection1 = nodes[i].value;
149                 connection2 = nodes[j].value;
150             }
151
152     con1[a] = connection1;
153     con2[a] = connection2;
154     a++;}
155 }
156
157 //check_svez(number,i);
158
159 fclose(fp);
160
161 i = 0;
162 cashe = fopen("cashe1.txt","w");
163 while (con1[i]!=0) {
164     fprintf(cashe,"%d %d \n",con1[i],con2[i]);
165     i++;
166 }
167 fclose(cashe);
168


```

Функция print\_connection

```

52 void print_connection(int count,int number1, int number2){
53     int i = 0;
54     int j = 0;
55     FILE *fp;
56
57     fp = fopen("gr1.txt","a");
58     for (i = 1; i <= count; i++) {
59         for (j = 1; j <= count; j++)
60             if ((number1 == nodes[i].number)&&(number2 == nodes[j].number)){
61                 fprintf(fp, "node%d -- node%d; \n",nodes[i].number,nodes[j].number );
62                 number1 = nodes[i].value;
63                 number2 = nodes[j].value;
64             } //else { printf("Error \n");}
65     }
66 }

```

 gr1.txt - Notepad  
File Edit Format View Help  

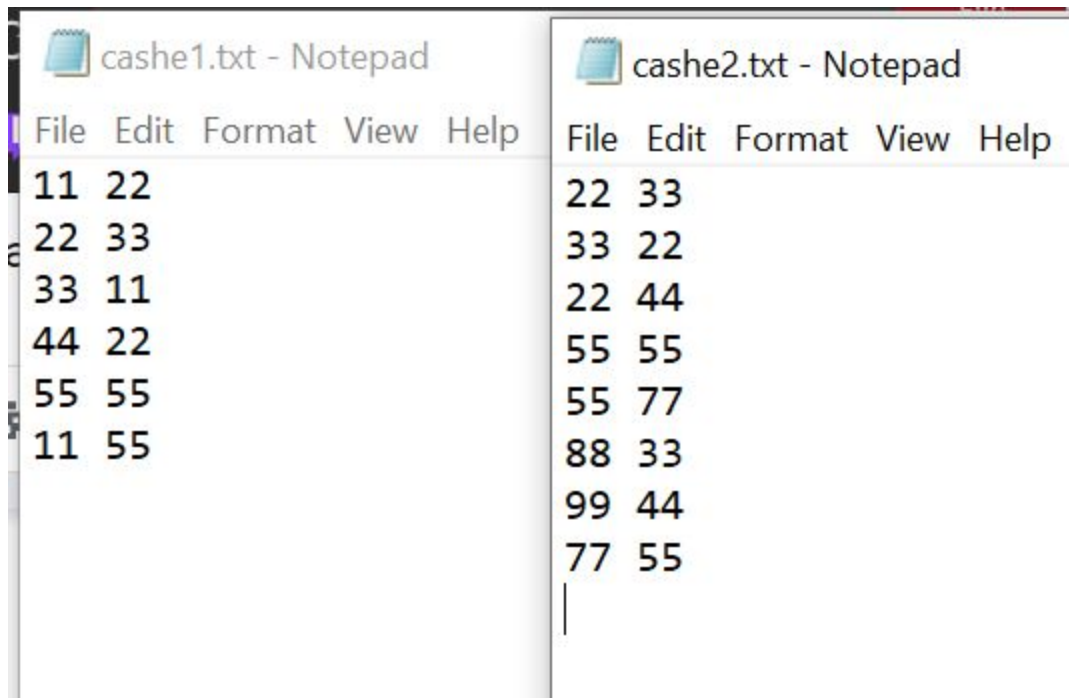
```
graph G {  
node1 [label = " 11 " ];  
node2 [label = " 22 " ];  
node3 [label = " 33 " ];  
node4 [label = " 44 " ];  
node5 [label = " 55 " ];  
node1 -- node5;  
node5 -- node5;  
node4 -- node2;  
node3 -- node1;  
node2 -- node3;  
node1 -- node2;  
  
}
```

(Пример записывания графа в текстовый файл)

После создания графа, программа создает еще один.

Далее программа смотрит в файлы cashe.txt и выбирает количество вершин

```
41  
42 i = 0;  
43 cashe1 = fopen("cashe1.txt","r");  
44 while(fscanf(cashe1,"%d %d \n",&cashe[i].number1,&cashe[i].number2)!=EOF){  
45     i++;  
46 }  
47 count1 = i;  
48 i = 0;  
49 cashe2 = fopen("cashe2.txt","r");  
50 while(fscanf(cashe2,"%d %d \n",&cashe[i].number3,&cashe[i].number4)!=EOF){  
51     i++;  
52 }  
53 count2 = i;  
54  
55 count = MAX(count1,count2);  
56
```



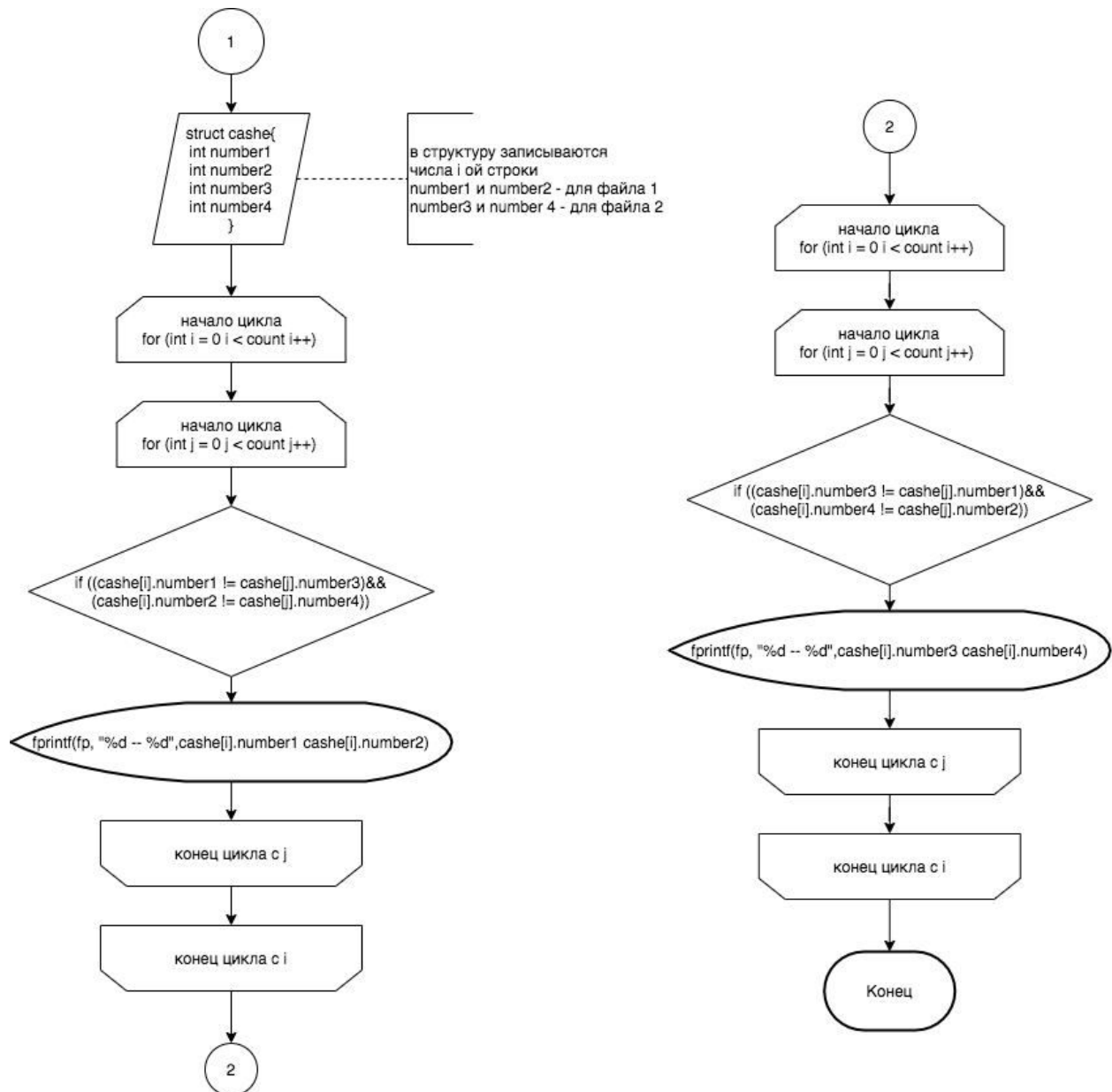
(Файлы cashe.txt)

Путем сравнения первых чисел из таблиц и вторых, программа находит неравные строки из 1го графа и выводит их в финальный объединенный граф.


Далее программа выводит весь второй граф в финальный файл

```
62  for (int i = 0; i < count; i++)
63      for (int j = 0; j < count; j++){
64          if ((cashe[i].number1 == cashe[j].number3)&&(cashe[i].number2 == cashe[j].number4)) {
65              // fprintf(fp, "%d -- %d ;\n",cashe[i].number1,cashe[i].number2);
66              break;
67              // printf("%d -- %d \n",cashe[i].number1,cashe[i].number2);
68          }else if ((cashe[i].number1 != cashe[j].number3)&&(cashe[i].number2 != cashe[j].number4)) {
69              if ((cashe[i].number1 != 0)&&(cashe[i].number2 != 0))
70                  fprintf(fp, "%d -- %d ;\n",cashe[i].number1,cashe[i].number2);
71              break;
72          }
73      }
74  //fprintf(fp, "\n__0____0____ \n" );
75  for (int i = 0; i < count; i++)
76      for (int j = 0; j < count; j++){
77          if ((cashe[i].number3 != cashe[j].number1)&&(cashe[i].number4 != cashe[j].number2)) {
78              // fprintf(fp, "%d -- %d ;\n",cashe[i].number1,cashe[i].number2);
79              // printf("%d -- %d \n",cashe[i].number1,cashe[i].number2);
80              if ((cashe[i].number3 != 0)&&(cashe[i].number4 != 0))
81                  fprintf(fp, "%d -- %d ;\n",cashe[i].number3,cashe[i].number4);
82              break;
83          }
84      }
```

Структурная схема алгоритма:



Итог выполнения программы:

 finalgraph.txt - Notepad  
File Edit Format View Help  
graph G {  
22 -- 33 ;  
55 -- 55 ;  
  
}

Вывод:

Мне удалось реализовать программу по объединению двух графов.