Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО ПРАКТИКУМУ №1
по курсу «Архитектура ЭВМ»
на тему:
«Разработка и отладка программ в вычислительном комплексе Тераграф»

Студент Рунов К.А.

Группа ИУ7-54Б

Вариант 18

Преподаватели Попов А.Ю., Ибрагимов С.В.

2024 г.

СОДЕРЖАНИЕ

1 Задание 1

```
iu7108@dl580:~/lab1$ host/host_main sw-kernel/sw_kernel.rawbinary
Open gpc on /dev/gpc6
Rawbinary loaded from sw-kernel/sw_kernel.rawbinary
sw_kernel version: 0x28102023
Leonhard clock frequency (LNH_CF) 180.021776 MHz
Test done
```

2 Задание 2

```
iu7108@dl580:~/lab2$ host/host_main sw-kernel/sw_kernel.rawbinary
select role from users where user=5 and time>7200;
Роль: 999 - Время доступа: 3596400
Роль: 998 - Время доступа: 3592800
Роль: 997 - Время доступа: 3589200
Роль: 996 - Время доступа: 3585600
Роль: 995 - Время доступа: 3582000
Роль: 994 - Время доступа: 3578400
```

Запись в логе:

```
Открывается доступ к /dev/gpc2
Программное ядро загружено из файла sw-kernel/sw_kernel.rawbinary
Введен запрос: select role from users where user=5 and time>7200;
Запрос принят в обработку.
Поиск ролей пользователя 5и time > 7200
```

3 Индивидуальное задание

3.1 Условие

Устройство формирования индексов SQL UNION. Сформировать в хост-подсистеме и передать в SPE 256 записей множества A (случайные числа в диапазоне 0..1024) и 256 записей множества B (случайные числа в диапазоне 0..1024). Сформировать в SPE множество C = A or B. Выполнить тестирование работы SPE, сравнив набор ключей в множестве C с ожидаемым.

3.2 Изменённый файл host_main.cpp

```
1  #include <iostream>
2  #include <iterator>
3  #include <string>
4  #include <regex>
5  #include <sstream>
6  #include <fstream>
7  #include <ctime>
8  #include "host_main.h"
9
10 using namespace std;
11
12 #define TEST_USER_COUNT 1000
13 #define TEST_ROLE_COUNT 1000
14
15 static unsigned long long seed{0x872d30b1eab1cd6e};
16 unsigned long long grn1024() {
17     seed = 828090353856353729 * seed + 2309503295186496403;
18     return seed % 1025;
19 }
20
21 int main(int argc, char** argv)
22 {
23     ofstream log("lab2.log"); //поток вывода сообщений
24     unsigned long long offs=0ull;
25     gpc *gpc64_inst; //указатель на класс gpc
```

```
26      regex select_regex_query("select␣+(.*?)␣+from␣+(.*?)␣+where␣
            +(.*?)=(.*?)␣+and␣+(.*?)>(.*);", //запрос
27              std::regex_constants::ECMAScript |
                    std::regex_constants::icase);
28
29      //Инициализация gpc
30      if (argc<2) {
31          log<<"Использование:␣host_main␣<путь␣к␣файлу␣
                rawbinary>"<<endl;
32          return -1;
33      }
34
35      //Захват ядра gpc и запись sw_kernel
36      gpc64_inst = new gpc();
37      log<<"Открывается␣доступ␣к␣"<<gpc64_inst->gpc_dev_path<<endl;
38      cout<<"Открывается␣доступ␣к␣"<<gpc64_inst->gpc_dev_path<<endl;
39      if (gpc64_inst->load_swk(argv[1])==0) {
40          log<<"Программное␣ядро␣загружено␣из␣файла␣
                "<<argv[1]<<endl;
41          cout<<"Программное␣ядро␣загружено␣из␣файла␣
                "<<argv[1]<<endl;
42      }
43      else {
44          log<<"Ошибка␣загрузки␣sw_kernel␣файла␣<<␣argv[1]"<<endl;
45          return -1;
46      }
47
48      //Инициализация таблицы для вложенного запроса
49      gpc64_inst->start(__event__(update)); //обработчик вставки
50
51  if (0){
52      //1-й вариант: пересылка коротких сообщений
53      for (uint32_t user=0;user<TEST_USER_COUNT;user++) {
54          for (uint32_t idx=0;idx<TEST_ROLE_COUNT;idx++,offs+=2) {
55              gpc64_inst->mq_send(users::key{.idx=idx,.user=user});
                    //запись о роли #idx
56              gpc64_inst->mq_send(users::val{.role=idx,.time=time_t(0)})
                    //роль и время доступа
57          }
58      }
59  }
```

```
60
61 if (0){
62     //2-й вариант: блочная передача
63     unsigned long long *buf = (unsigned long
           long*)malloc(sizeof(unsigned long
           long)*TEST_USER_COUNT*TEST_ROLE_COUNT*2);
64     for (uint32_t user=0,offs=0;user<TEST_USER_COUNT;user++) {
65         for (uint32_t idx=0;idx<TEST_ROLE_COUNT;idx++,offs+=2) {
66             buf[offs]=users::key{.idx=idx,.user=user};
67             buf[offs+1]=users::val{.role=idx,.time=time_t(idx*3600)};
68         }
69     }
70     auto send_buf_th = gpc64_inst->mq_send(sizeof(unsigned long
           long)*TEST_USER_COUNT*TEST_ROLE_COUNT*2,(char*)buf);
71     send_buf_th->join();
72     free(buf);
73 }
74
75     // XXX
76     for (uint64_t idx=0; idx<(256+256); idx++) {
77         gpc64_inst->mq_send(mystruct::key{.idx=idx});
78         gpc64_inst->mq_send(mystruct::val{.value=grn1024()});
79
80         auto where = gpc64_inst->mq_receive();
81         auto k = gpc64_inst->mq_receive();
82         auto v = gpc64_inst->mq_receive();
83
84         if (where == 0ull) {
85             cout << "inserted␣in␣A␣idx␣" << k << "␣v␣" << v <<
                   endl;
86             log << "inserted␣in␣A␣idx␣" << k << "␣v␣" << v <<
                   endl;
87         } else if (where == 1ull) {
88             cout << "inserted␣in␣B␣idx␣" << k << "␣v␣" << v <<
                   endl;
89             log << "inserted␣in␣B␣idx␣" << k << "␣v␣" << v <<
                   endl;
90         }
91     }
92
93     //Терминальный символ
```

```
94      gpc64_inst->mq_send(-1ull);
95
96      gpc64_inst->start(__event__(select)); //обработчик запроса пои
            ска
97      while(1) {
98          uint64_t result_key = gpc64_inst->mq_receive();
99          if (result_key!=-1ull) {
100             uint64_t result_val = gpc64_inst->mq_receive();
101             cout << "C idx: " <<
                    mystruct::key::from_int(result_key).idx << " - ";
102             cout << "C value: " <<
                    mystruct::val::from_int(result_val).value << endl;
103             log << "C idx: " <<
                    mystruct::key::from_int(result_key).idx << " - ";
104             log << "C value: " <<
                    mystruct::val::from_int(result_val).value << endl;
105         } else {
106             cout << "done" << endl;
107             log << "done" << endl;
108             break;
109         }
110     }
111
112     while(0) {
113         string query1;
114         //разбор полей запроса
115         smatch match_query1;
116         getline(cin, query1);
117         log<<"Введен запрос: "<<query1<<endl;
118         if (!query1.compare("exit")) {
119             gpc64_inst->mq_send(-1ull);
120             break;
121         }
122         if (regex_match (query1, match_query1,
                select_regex_query) &&
123             match_query1[3]=="user" &&
124             match_query1[5] == "time") {
125             //match_query1[1] - возвращаемое поле запроса
126             //match_query1[2] - номер структуры запроса
127             //match_query1[3] - поле поиска 1
128             //match_query1[4] - значение поля поиска 1
```

```
129             //match_query1[5] - поле поиска 2
130             //match_query1[6] - значение поля поиска 2
131             log << "Запрос принят в обработку." << endl;
132             log << "Поиск ролей пользователя " << match_query1[4]
                  << "и time > " << time_t(stoi(match_query1[6])) <<
                  endl;
133             gpc64_inst->mq_send(stoi(match_query1[4]));
                  //пользователь
134             gpc64_inst->mq_send(stoi(match_query1[6])); //время до
                  ступа
135             while (1) {
136                 uint64_t result = gpc64_inst->mq_receive();
137                 if (result!=-1ull) {
138                     cout << "Роль: " <<
                          users::val::from_int(result).role << " - ";
139                     cout << "Время доступа: " <<
                          users::val::from_int(result).time << endl;
140                 } else {
141                     break;
142                 }
143             }
144         } else {
145             log << "Ошибка в запросе!" << endl;
146         }
147     }
148     log << "Выход!" << endl;
149     return 0;
150 }
```

3.3 Изменённый файл common_struct.h

```
1 #ifndef COMMON_STRUCT
2 #define COMMON_STRUCT
3
4 #ifdef __riscv64__
5 #include "map.h"
6 #endif
7 #include "compose_keys.hxx"
8
9
10 //Номера структур данных в SPE
```

```
11 enum Structures : uint32_t {
12     null            = 0,    //Нулевая структура не используется
13     users_pnum      = 1,    //Таблица 1
14     resources_pnum  = 2,        //Таблица 2
15     a_pnum    = 3,
16     b_pnum    = 4,
17     c_pnum    = 5
18 };
19
20 #ifdef __riscv64__
21 //Задание даипазонов и курсоров
22 template<typename Range>
23 struct reverse {
24         Range r;
25         [[gnu::always_inline]] reverse(Range r) : r(r) {}
26         [[gnu::always_inline]] auto begin() {return r.rbegin();}
27         [[gnu::always_inline]] auto end() {return r.rend();}
28 };
29
30 template<typename K, typename V>
31 struct Handle {
32         bool ret_val;
33         K k{get_result_key<K>()};
34         V v{get_result_value<V>()};
35         [[gnu::always_inline]] Handle(bool ret_val) :
              ret_val(ret_val) {
36         }
37
38         [[gnu::always_inline]] operator bool() const {
39                 return ret_val;
40         }
41
42         [[gnu::always_inline]] K key() const {
43                 return k;
44         }
45
46         [[gnu::always_inline]] V value() const {
47                 return v;
48         }
49 };
50 #endif
```

```
51
52
53 ////////////////////////////////////
54 // Описание формата ключа и значения
55 ////////////////////////////////////
56
57
58 struct users {
59     using vertex_t = uint32_t;
60     int struct_number;
61     constexpr users(int struct_number) :
          struct_number(struct_number) {}
62     static const uint32_t idx_bits = 32;
63     static const uint32_t idx_max = (1ull << idx_bits) - 1;
64     static const uint32_t idx_min = idx_max;
65
66     //Запись для формирования ключей (* - наиболее значимые биты п
          оля)
67     STRUCT(key)
68     {
69         uint32_t    idx     :idx_bits;  //Поле 0:
70         uint32_t    user    :32;        //Поле 1*
71     };
72
73     //Запись для формирования значений
74     STRUCT(val)
75     {
76         uint32_t    role    :32;        //Поле 0:
77         time_t      time    :32;        //Поле 1*
78     };
79     //Обязательная типизация
80     #ifdef __riscv64__
81     DEFINE_DEFAULT_KEYVAL(key, val)
82     #endif
83 };
84
85 struct mystruct {
86     int struct_number;
87     constexpr mystruct(int struct_number) :
          struct_number(struct_number) {}
88
```

```
89      STRUCT(key)
90      {
91          uint64_t    idx     :64;
92      };
93
94      STRUCT(val)
95      {
96          uint64_t    value   :64;
97      };
98
99      #ifdef __riscv64__
100     DEFINE_DEFAULT_KEYVAL(key, val)
101     #endif
102 };
103
104 constexpr users USERS(Structures::users_pnum);
105 constexpr mystruct A(Structures::a_pnum);
106 constexpr mystruct B(Structures::b_pnum);
107 constexpr mystruct C(Structures::c_pnum);
108
109 #endif //COMMON_STRUCT
```

3.4 Изменённый файл sw_kernel_main.cpp

```
1  #include <stdlib.h>
2  #include <ctime>
3  #include "lnh64.hxx"
4  #include "gpc_io_swk.h"
5  #include "gpc_handlers.h"
6  #include "iterators.h"
7  #include "common_struct.h"
8  #include "compose_keys.hxx"
9
10 #define __fast_recall__
11
12 extern lnh lnh_core;
13 volatile unsigned int event_source;
14
15 int main(void) {
16     ////////////////////////////////////////////////////
17     //                  Main Event Loop
```

```
18      ////////////////////////////////////////////////////////
19      //Leonhard driver structure should be initialised
20      lnh_init();
21      for (;;) {
22          //Wait for event
23          event_source = wait_event();
24          switch(event_source) {
25              ////////////////////////////////////////////
26              //  Measure GPN operation frequency
27              ////////////////////////////////////////////
28              case __event__(update) : update(); break;
29              case __event__(select) : select(); break;
30          }
31          set_gpc_state(READY);
32      }
33  }
34
35  //-------------------------------------------------------------
36  //      Вставка ключа и значения в структуру
37  //-------------------------------------------------------------
38
39  void update() {
40
41          while(0){ // XXX
42                  users::key key=users::key::from_int(mq_receive());
43                  if (key==-1ull) break;
44                  users::val val=users::val::from_int(mq_receive());
45                  // Поля структуры могут записываться явно следующи
                        м образом
46                  //      auto new_key =
                        users::key{.rec_idx=1,.user=2};
47                  //      auto new_val =
                        users::val{.role=3,.lst_time=0}
48                  // Копирование полей в переменные можно выполнить
                        следующим образом:
49                  //      auto user = key.user;
50                  //      auto [lst_time,role] = val;
51                  USERS.ins_async(key,val); //Вставка в таблицу с ти
                        пизацией uint64_t
52          }
53
```

```
54          unsigned long long i = 0;
55          while(1){ // XXX
56              mystruct::key key =
                    mystruct::key::from_int(mq_receive());
57              if(key==-1ull) break;
58              mystruct::val val =
                    mystruct::val::from_int(mq_receive());
59
60              if (++i % 2) {
61                  A.ins_async(key,val);
62                  mq_send(0ull);
63                  mq_send(key);
64                  mq_send(val);
65              } else {
66                  B.ins_async(key,val);
67                  mq_send(1ull);
68                  mq_send(key);
69                  mq_send(val);
70              }
71          }
72          //mq_send(-1ull);
73  }
74
75
76  //-------------------------------------------------------------
77  //      Передать все роли пользователя и время доступа
78  //-------------------------------------------------------------
79
80  void select() {
81          while(0){ // XXX
82                  uint32_t quser = mq_receive();
83                  if (quser==-1) break;
84                  uint32_t qtime = mq_receive();
85                  //Найдем все роли пользователя и последнее время д
                        оступа:
86                  // Результаты поиска могут быть доступны следующим
                         образом:
87                  //      auto user =
                        USERS.search(users::key{.idx=1,.user=2}).key().user
88                  //      auto role =
                        USERS.search(users::key{.idx=3,.user=4}).value().rol
```

```
89
90                  //Вариант 1 - обход записей пользователя явным обр
                        азом
91                  auto crole =
                        USERS.nsm(users::key{.idx=users::idx_min,.user=quser
92                  while (crole && crole.key().user==quser) {
93                          if (crole.value().time>qtime)
                                mq_send(crole.value());
94                          crole = USERS.nsm(crole.key());
95                  }
96
97                  //Вариант 2 - использование итератора
98                  // for (auto val : role_range(USERS,quser)) {
99                  //         if (val.time>qtime) mq_send(val);
100                 // }
101                 mq_send(-1ull);
102         }
103
104         //C.or_sync(A.struct_number, B.struct_number);
105         lnh_or_sync(A.struct_number, B.struct_number,
                C.struct_number);
106
107         uint64_t idx = 0;
108         while(1){
109             auto kv = C.search(mystruct::key{.idx=idx});
110             if (kv) {
111                 auto k = kv.key().idx;
112                 auto v = kv.value().value;
113                 mq_send(k);
114                 mq_send(v);
115             } else {
116                 break;
117             }
118             ++idx;
119         }
120         mq_send(-1ull);
121 }
```

3.5 Результат работы программы

```
1 Открывается доступ к /dev/gpc0
```

```
Программное ядро загружено из файла sw-kernel/sw_kernel.rawbinary
inserted in A idx 0 v 5
inserted in B idx 1 v 95
inserted in A idx 2 v 300
inserted in B idx 3 v 1023
inserted in A idx 4 v 444
inserted in B idx 5 v 401
inserted in A idx 6 v 798
inserted in B idx 7 v 191
inserted in A idx 8 v 890
inserted in B idx 9 v 432
inserted in A idx 10 v 23
inserted in B idx 11 v 866
inserted in A idx 12 v 645
inserted in B idx 13 v 21
inserted in A idx 14 v 45
inserted in B idx 15 v 196
inserted in A idx 16 v 471
inserted in B idx 17 v 937
inserted in A idx 18 v 355
inserted in B idx 19 v 37
inserted in A idx 20 v 941
inserted in B idx 21 v 727
inserted in A idx 22 v 579
inserted in B idx 23 v 27
inserted in A idx 24 v 120
inserted in B idx 25 v 110
inserted in A idx 26 v 942
inserted in B idx 27 v 612
inserted in A idx 28 v 487
inserted in B idx 29 v 161
inserted in A idx 30 v 876
inserted in B idx 31 v 787
inserted in A idx 32 v 440
inserted in B idx 33 v 703
inserted in A idx 34 v 724
inserted in B idx 35 v 245
inserted in A idx 36 v 77
inserted in B idx 37 v 1004
inserted in A idx 38 v 290
inserted in B idx 39 v 389
```

```
43 inserted in A idx 40 v 249
44 inserted in B idx 41 v 666
45 inserted in A idx 42 v 759
46 inserted in B idx 43 v 353
47 inserted in A idx 44 v 605
48 inserted in B idx 45 v 76
49 inserted in A idx 46 v 168
50 inserted in B idx 47 v 956
51 inserted in A idx 48 v 516
52 inserted in B idx 49 v 887
53 inserted in A idx 50 v 630
54 inserted in B idx 51 v 357
55 inserted in A idx 52 v 160
56 inserted in B idx 53 v 264
57 inserted in A idx 54 v 646
58 inserted in B idx 55 v 920
59 inserted in A idx 56 v 725
60 inserted in B idx 57 v 554
61 inserted in A idx 58 v 527
62 inserted in B idx 59 v 970
63 inserted in A idx 60 v 758
64 inserted in B idx 61 v 603
65 inserted in A idx 62 v 702
66 inserted in B idx 63 v 192
67 inserted in A idx 64 v 400
68 inserted in B idx 65 v 407
69 inserted in A idx 66 v 212
70 inserted in B idx 67 v 733
71 inserted in A idx 68 v 624
72 inserted in B idx 69 v 137
73 inserted in A idx 70 v 73
74 inserted in B idx 71 v 442
75 inserted in A idx 72 v 586
76 inserted in B idx 73 v 964
77 inserted in A idx 74 v 857
78 inserted in B idx 75 v 13
79 inserted in A idx 76 v 404
80 inserted in B idx 77 v 786
81 inserted in A idx 78 v 826
82 inserted in B idx 79 v 257
83 inserted in A idx 80 v 386
```

```
 84 inserted in B idx 81 v 726
 85 inserted in A idx 82 v 899
 86 inserted in B idx 83 v 620
 87 inserted in A idx 84 v 469
 88 inserted in B idx 85 v 771
 89 inserted in A idx 86 v 608
 90 inserted in B idx 87 v 160
 91 inserted in A idx 88 v 902
 92 inserted in B idx 89 v 968
 93 inserted in A idx 90 v 134
 94 inserted in B idx 91 v 587
 95 inserted in A idx 92 v 955
 96 inserted in B idx 93 v 403
 97 inserted in A idx 94 v 221
 98 inserted in B idx 95 v 506
 99 inserted in A idx 96 v 469
100 inserted in B idx 97 v 372
101 inserted in A idx 98 v 247
102 inserted in B idx 99 v 425
103 inserted in A idx 100 v 407
104 inserted in B idx 101 v 801
105 inserted in A idx 102 v 90
106 inserted in B idx 103 v 383
107 inserted in A idx 104 v 561
108 inserted in B idx 105 v 667
109 inserted in A idx 106 v 369
110 inserted in B idx 107 v 602
111 inserted in A idx 108 v 523
112 inserted in B idx 109 v 80
113 inserted in A idx 110 v 38
114 inserted in B idx 111 v 1018
115 inserted in A idx 112 v 188
116 inserted in B idx 113 v 350
117 inserted in A idx 114 v 762
118 inserted in B idx 115 v 456
119 inserted in A idx 116 v 470
120 inserted in B idx 117 v 823
121 inserted in A idx 118 v 431
122 inserted in B idx 119 v 948
123 inserted in A idx 120 v 101
124 inserted in B idx 121 v 712
```

```
125 inserted in A idx 122 v 587
126 inserted in B idx 123 v 8
127 inserted in A idx 124 v 664
128 inserted in B idx 125 v 161
129 inserted in A idx 126 v 801
130 inserted in B idx 127 v 373
131 inserted in A idx 128 v 69
132 inserted in B idx 129 v 90
133 inserted in A idx 130 v 53
134 inserted in B idx 131 v 279
135 inserted in A idx 132 v 447
136 inserted in B idx 133 v 539
137 inserted in A idx 134 v 128
138 inserted in B idx 135 v 334
139 inserted in A idx 136 v 904
140 inserted in B idx 137 v 729
141 inserted in A idx 138 v 1022
142 inserted in B idx 139 v 614
143 inserted in A idx 140 v 456
144 inserted in B idx 141 v 176
145 inserted in A idx 142 v 927
146 inserted in B idx 143 v 215
147 inserted in A idx 144 v 840
148 inserted in B idx 145 v 139
149 inserted in A idx 146 v 945
150 inserted in B idx 147 v 981
151 inserted in A idx 148 v 11
152 inserted in B idx 149 v 560
153 inserted in A idx 150 v 917
154 inserted in B idx 151 v 561
155 inserted in A idx 152 v 152
156 inserted in B idx 153 v 252
157 inserted in A idx 154 v 918
158 inserted in B idx 155 v 429
159 inserted in A idx 156 v 557
160 inserted in B idx 157 v 973
161 inserted in A idx 158 v 43
162 inserted in B idx 159 v 249
163 inserted in A idx 160 v 178
164 inserted in B idx 161 v 466
165 inserted in A idx 162 v 23
```

```
166 inserted in B idx 163 v 826
167 inserted in A idx 164 v 394
168 inserted in B idx 165 v 987
169 inserted in A idx 166 v 558
170 inserted in B idx 167 v 497
171 inserted in A idx 168 v 700
172 inserted in B idx 169 v 466
173 inserted in A idx 170 v 90
174 inserted in B idx 171 v 895
175 inserted in A idx 172 v 635
176 inserted in B idx 173 v 593
177 inserted in A idx 174 v 982
178 inserted in B idx 175 v 104
179 inserted in A idx 176 v 281
180 inserted in B idx 177 v 482
181 inserted in A idx 178 v 92
182 inserted in B idx 179 v 848
183 inserted in A idx 180 v 803
184 inserted in B idx 181 v 454
185 inserted in A idx 182 v 915
186 inserted in B idx 183 v 107
187 inserted in A idx 184 v 119
188 inserted in B idx 185 v 450
189 inserted in A idx 186 v 49
190 inserted in B idx 187 v 634
191 inserted in A idx 188 v 740
192 inserted in B idx 189 v 849
193 inserted in A idx 190 v 64
194 inserted in B idx 191 v 807
195 inserted in A idx 192 v 1008
196 inserted in B idx 193 v 775
197 inserted in A idx 194 v 59
198 inserted in B idx 195 v 906
199 inserted in A idx 196 v 564
200 inserted in B idx 197 v 149
201 inserted in A idx 198 v 69
202 inserted in B idx 199 v 253
203 inserted in A idx 200 v 982
204 inserted in B idx 201 v 300
205 inserted in A idx 202 v 831
206 inserted in B idx 203 v 966
```

```
207 inserted in A idx 204 v 148
208 inserted in B idx 205 v 377
209 inserted in A idx 206 v 768
210 inserted in B idx 207 v 839
211 inserted in A idx 208 v 657
212 inserted in B idx 209 v 330
213 inserted in A idx 210 v 691
214 inserted in B idx 211 v 551
215 inserted in A idx 212 v 860
216 inserted in B idx 213 v 209
217 inserted in A idx 214 v 671
218 inserted in B idx 215 v 95
219 inserted in A idx 216 v 272
220 inserted in B idx 217 v 770
221 inserted in A idx 218 v 481
222 inserted in B idx 219 v 7
223 inserted in A idx 220 v 19
224 inserted in B idx 221 v 601
225 inserted in A idx 222 v 250
226 inserted in B idx 223 v 82
227 inserted in A idx 224 v 617
228 inserted in B idx 225 v 717
229 inserted in A idx 226 v 928
230 inserted in B idx 227 v 461
231 inserted in A idx 228 v 89
232 inserted in B idx 229 v 981
233 inserted in A idx 230 v 454
234 inserted in B idx 231 v 325
235 inserted in A idx 232 v 1
236 inserted in B idx 233 v 600
237 inserted in A idx 234 v 299
238 inserted in B idx 235 v 572
239 inserted in A idx 236 v 612
240 inserted in B idx 237 v 204
241 inserted in A idx 238 v 261
242 inserted in B idx 239 v 270
243 inserted in A idx 240 v 441
244 inserted in B idx 241 v 671
245 inserted in A idx 242 v 895
246 inserted in B idx 243 v 632
247 inserted in A idx 244 v 107
```

```
248 inserted in B idx 245 v 757
249 inserted in A idx 246 v 939
250 inserted in B idx 247 v 943
251 inserted in A idx 248 v 26
252 inserted in B idx 249 v 596
253 inserted in A idx 250 v 713
254 inserted in B idx 251 v 540
255 inserted in A idx 252 v 232
256 inserted in B idx 253 v 367
257 inserted in A idx 254 v 296
258 inserted in B idx 255 v 38
259 inserted in A idx 256 v 749
260 inserted in B idx 257 v 507
261 inserted in A idx 258 v 610
262 inserted in B idx 259 v 162
263 inserted in A idx 260 v 902
264 inserted in B idx 261 v 521
265 inserted in A idx 262 v 270
266 inserted in B idx 263 v 750
267 inserted in A idx 264 v 833
268 inserted in B idx 265 v 336
269 inserted in A idx 266 v 704
270 inserted in B idx 267 v 99
271 inserted in A idx 268 v 922
272 inserted in B idx 269 v 596
273 inserted in A idx 270 v 489
274 inserted in B idx 271 v 953
275 inserted in A idx 272 v 716
276 inserted in B idx 273 v 114
277 inserted in A idx 274 v 362
278 inserted in B idx 275 v 854
279 inserted in A idx 276 v 3
280 inserted in B idx 277 v 990
281 inserted in A idx 278 v 271
282 inserted in B idx 279 v 560
283 inserted in A idx 280 v 866
284 inserted in B idx 281 v 523
285 inserted in A idx 282 v 62
286 inserted in B idx 283 v 646
287 inserted in A idx 284 v 560
288 inserted in B idx 285 v 272
```

```
289 inserted in A idx 286 v 304
290 inserted in B idx 287 v 760
291 inserted in A idx 288 v 315
292 inserted in B idx 289 v 778
293 inserted in A idx 290 v 741
294 inserted in B idx 291 v 176
295 inserted in A idx 292 v 772
296 inserted in B idx 293 v 533
297 inserted in A idx 294 v 675
298 inserted in B idx 295 v 9
299 inserted in A idx 296 v 573
300 inserted in B idx 297 v 14
301 inserted in A idx 298 v 341
302 inserted in B idx 299 v 67
303 inserted in A idx 300 v 29
304 inserted in B idx 301 v 959
305 inserted in A idx 302 v 838
306 inserted in B idx 303 v 813
307 inserted in A idx 304 v 204
308 inserted in B idx 305 v 376
309 inserted in A idx 306 v 270
310 inserted in B idx 307 v 388
311 inserted in A idx 308 v 991
312 inserted in B idx 309 v 791
313 inserted in A idx 310 v 320
314 inserted in B idx 311 v 69
315 inserted in A idx 312 v 180
316 inserted in B idx 313 v 610
317 inserted in A idx 314 v 189
318 inserted in B idx 315 v 236
319 inserted in A idx 316 v 748
320 inserted in B idx 317 v 85
321 inserted in A idx 318 v 662
322 inserted in B idx 319 v 737
323 inserted in A idx 320 v 218
324 inserted in B idx 321 v 647
325 inserted in A idx 322 v 905
326 inserted in B idx 323 v 828
327 inserted in A idx 324 v 422
328 inserted in B idx 325 v 779
329 inserted in A idx 326 v 27
```

```
330 inserted in B idx 327 v 200
331 inserted in A idx 328 v 995
332 inserted in B idx 329 v 308
333 inserted in A idx 330 v 828
334 inserted in B idx 331 v 564
335 inserted in A idx 332 v 908
336 inserted in B idx 333 v 788
337 inserted in A idx 334 v 655
338 inserted in B idx 335 v 68
339 inserted in A idx 336 v 630
340 inserted in B idx 337 v 787
341 inserted in A idx 338 v 999
342 inserted in B idx 339 v 135
343 inserted in A idx 340 v 1
344 inserted in B idx 341 v 899
345 inserted in A idx 342 v 994
346 inserted in B idx 343 v 262
347 inserted in A idx 344 v 408
348 inserted in B idx 345 v 592
349 inserted in A idx 346 v 704
350 inserted in B idx 347 v 708
351 inserted in A idx 348 v 535
352 inserted in B idx 349 v 867
353 inserted in A idx 350 v 70
354 inserted in B idx 351 v 385
355 inserted in A idx 352 v 976
356 inserted in B idx 353 v 831
357 inserted in A idx 354 v 1019
358 inserted in B idx 355 v 257
359 inserted in A idx 356 v 562
360 inserted in B idx 357 v 416
361 inserted in A idx 358 v 868
362 inserted in B idx 359 v 1005
363 inserted in A idx 360 v 799
364 inserted in B idx 361 v 921
365 inserted in A idx 362 v 175
366 inserted in B idx 363 v 669
367 inserted in A idx 364 v 85
368 inserted in B idx 365 v 877
369 inserted in A idx 366 v 760
370 inserted in B idx 367 v 134
```

```
371 inserted in A idx 368 v 704
372 inserted in B idx 369 v 824
373 inserted in A idx 370 v 10
374 inserted in B idx 371 v 809
375 inserted in A idx 372 v 258
376 inserted in B idx 373 v 793
377 inserted in A idx 374 v 692
378 inserted in B idx 375 v 206
379 inserted in A idx 376 v 739
380 inserted in B idx 377 v 273
381 inserted in A idx 378 v 750
382 inserted in B idx 379 v 745
383 inserted in A idx 380 v 867
384 inserted in B idx 381 v 586
385 inserted in A idx 382 v 554
386 inserted in B idx 383 v 242
387 inserted in A idx 384 v 334
388 inserted in B idx 385 v 490
389 inserted in A idx 386 v 696
390 inserted in B idx 387 v 325
391 inserted in A idx 388 v 897
392 inserted in B idx 389 v 455
393 inserted in A idx 390 v 337
394 inserted in B idx 391 v 725
395 inserted in A idx 392 v 163
396 inserted in B idx 393 v 236
397 inserted in A idx 394 v 913
398 inserted in B idx 395 v 987
399 inserted in A idx 396 v 760
400 inserted in B idx 397 v 338
401 inserted in A idx 398 v 863
402 inserted in B idx 399 v 261
403 inserted in A idx 400 v 438
404 inserted in B idx 401 v 680
405 inserted in A idx 402 v 988
406 inserted in B idx 403 v 176
407 inserted in A idx 404 v 547
408 inserted in B idx 405 v 569
409 inserted in A idx 406 v 629
410 inserted in B idx 407 v 851
411 inserted in A idx 408 v 136
```

```
412 inserted in B idx 409 v 709
413 inserted in A idx 410 v 736
414 inserted in B idx 411 v 430
415 inserted in A idx 412 v 198
416 inserted in B idx 413 v 852
417 inserted in A idx 414 v 425
418 inserted in B idx 415 v 289
419 inserted in A idx 416 v 106
420 inserted in B idx 417 v 417
421 inserted in A idx 418 v 631
422 inserted in B idx 419 v 103
423 inserted in A idx 420 v 213
424 inserted in B idx 421 v 91
425 inserted in A idx 422 v 207
426 inserted in B idx 423 v 587
427 inserted in A idx 424 v 159
428 inserted in B idx 425 v 289
429 inserted in A idx 426 v 146
430 inserted in B idx 427 v 39
431 inserted in A idx 428 v 294
432 inserted in B idx 429 v 880
433 inserted in A idx 430 v 109
434 inserted in B idx 431 v 516
435 inserted in A idx 432 v 311
436 inserted in B idx 433 v 717
437 inserted in A idx 434 v 216
438 inserted in B idx 435 v 372
439 inserted in A idx 436 v 915
440 inserted in B idx 437 v 900
441 inserted in A idx 438 v 5
442 inserted in B idx 439 v 587
443 inserted in A idx 440 v 334
444 inserted in B idx 441 v 479
445 inserted in A idx 442 v 856
446 inserted in B idx 443 v 161
447 inserted in A idx 444 v 918
448 inserted in B idx 445 v 444
449 inserted in A idx 446 v 248
450 inserted in B idx 447 v 397
451 inserted in A idx 448 v 722
452 inserted in B idx 449 v 964
```

```
453 inserted in A idx 450 v 965
454 inserted in B idx 451 v 733
455 inserted in A idx 452 v 472
456 inserted in B idx 453 v 149
457 inserted in A idx 454 v 463
458 inserted in B idx 455 v 616
459 inserted in A idx 456 v 92
460 inserted in B idx 457 v 344
461 inserted in A idx 458 v 845
462 inserted in B idx 459 v 843
463 inserted in A idx 460 v 267
464 inserted in B idx 461 v 827
465 inserted in A idx 462 v 591
466 inserted in B idx 463 v 607
467 inserted in A idx 464 v 359
468 inserted in B idx 465 v 1020
469 inserted in A idx 466 v 981
470 inserted in B idx 467 v 940
471 inserted in A idx 468 v 146
472 inserted in B idx 469 v 839
473 inserted in A idx 470 v 276
474 inserted in B idx 471 v 880
475 inserted in A idx 472 v 627
476 inserted in B idx 473 v 17
477 inserted in A idx 474 v 718
478 inserted in B idx 475 v 596
479 inserted in A idx 476 v 427
480 inserted in B idx 477 v 53
481 inserted in A idx 478 v 952
482 inserted in B idx 479 v 555
483 inserted in A idx 480 v 905
484 inserted in B idx 481 v 37
485 inserted in A idx 482 v 583
486 inserted in B idx 483 v 1022
487 inserted in A idx 484 v 774
488 inserted in B idx 485 v 465
489 inserted in A idx 486 v 60
490 inserted in B idx 487 v 829
491 inserted in A idx 488 v 1010
492 inserted in B idx 489 v 787
493 inserted in A idx 490 v 677
```

```
494 inserted in B idx 491 v 139
495 inserted in A idx 492 v 141
496 inserted in B idx 493 v 352
497 inserted in A idx 494 v 698
498 inserted in B idx 495 v 666
499 inserted in A idx 496 v 471
500 inserted in B idx 497 v 924
501 inserted in A idx 498 v 983
502 inserted in B idx 499 v 136
503 inserted in A idx 500 v 512
504 inserted in B idx 501 v 832
505 inserted in A idx 502 v 397
506 inserted in B idx 503 v 798
507 inserted in A idx 504 v 892
508 inserted in B idx 505 v 856
509 inserted in A idx 506 v 1012
510 inserted in B idx 507 v 569
511 inserted in A idx 508 v 696
512 inserted in B idx 509 v 928
513 inserted in A idx 510 v 632
514 inserted in B idx 511 v 238
515 C idx: 0 - C value: 5
516 C idx: 1 - C value: 95
517 C idx: 2 - C value: 300
518 C idx: 3 - C value: 1023
519 C idx: 4 - C value: 444
520 C idx: 5 - C value: 401
521 C idx: 6 - C value: 798
522 C idx: 7 - C value: 191
523 C idx: 8 - C value: 890
524 C idx: 9 - C value: 432
525 C idx: 10 - C value: 23
526 C idx: 11 - C value: 866
527 C idx: 12 - C value: 645
528 C idx: 13 - C value: 21
529 C idx: 14 - C value: 45
530 C idx: 15 - C value: 196
531 C idx: 16 - C value: 471
532 C idx: 17 - C value: 937
533 C idx: 18 - C value: 355
534 C idx: 19 - C value: 37
```

```
C idx: 20 - C value: 941
C idx: 21 - C value: 727
C idx: 22 - C value: 579
C idx: 23 - C value: 27
C idx: 24 - C value: 120
C idx: 25 - C value: 110
C idx: 26 - C value: 942
C idx: 27 - C value: 612
C idx: 28 - C value: 487
C idx: 29 - C value: 161
C idx: 30 - C value: 876
C idx: 31 - C value: 787
C idx: 32 - C value: 440
C idx: 33 - C value: 703
C idx: 34 - C value: 724
C idx: 35 - C value: 245
C idx: 36 - C value: 77
C idx: 37 - C value: 1004
C idx: 38 - C value: 290
C idx: 39 - C value: 389
C idx: 40 - C value: 249
C idx: 41 - C value: 666
C idx: 42 - C value: 759
C idx: 43 - C value: 353
C idx: 44 - C value: 605
C idx: 45 - C value: 76
C idx: 46 - C value: 168
C idx: 47 - C value: 956
C idx: 48 - C value: 516
C idx: 49 - C value: 887
C idx: 50 - C value: 630
C idx: 51 - C value: 357
C idx: 52 - C value: 160
C idx: 53 - C value: 264
C idx: 54 - C value: 646
C idx: 55 - C value: 920
C idx: 56 - C value: 725
C idx: 57 - C value: 554
C idx: 58 - C value: 527
C idx: 59 - C value: 970
C idx: 60 - C value: 758
```

```
C idx: 61 - C value: 603
C idx: 62 - C value: 702
C idx: 63 - C value: 192
C idx: 64 - C value: 400
C idx: 65 - C value: 407
C idx: 66 - C value: 212
C idx: 67 - C value: 733
C idx: 68 - C value: 624
C idx: 69 - C value: 137
C idx: 70 - C value: 73
C idx: 71 - C value: 442
C idx: 72 - C value: 586
C idx: 73 - C value: 964
C idx: 74 - C value: 857
C idx: 75 - C value: 13
C idx: 76 - C value: 404
C idx: 77 - C value: 786
C idx: 78 - C value: 826
C idx: 79 - C value: 257
C idx: 80 - C value: 386
C idx: 81 - C value: 726
C idx: 82 - C value: 899
C idx: 83 - C value: 620
C idx: 84 - C value: 469
C idx: 85 - C value: 771
C idx: 86 - C value: 608
C idx: 87 - C value: 160
C idx: 88 - C value: 902
C idx: 89 - C value: 968
C idx: 90 - C value: 134
C idx: 91 - C value: 587
C idx: 92 - C value: 955
C idx: 93 - C value: 403
C idx: 94 - C value: 221
C idx: 95 - C value: 506
C idx: 96 - C value: 469
C idx: 97 - C value: 372
C idx: 98 - C value: 247
C idx: 99 - C value: 425
C idx: 100 - C value: 407
C idx: 101 - C value: 801
```

```
C idx: 102 - C value: 90
C idx: 103 - C value: 383
C idx: 104 - C value: 561
C idx: 105 - C value: 667
C idx: 106 - C value: 369
C idx: 107 - C value: 602
C idx: 108 - C value: 523
C idx: 109 - C value: 80
C idx: 110 - C value: 38
C idx: 111 - C value: 1018
C idx: 112 - C value: 188
C idx: 113 - C value: 350
C idx: 114 - C value: 762
C idx: 115 - C value: 456
C idx: 116 - C value: 470
C idx: 117 - C value: 823
C idx: 118 - C value: 431
C idx: 119 - C value: 948
C idx: 120 - C value: 101
C idx: 121 - C value: 712
C idx: 122 - C value: 587
C idx: 123 - C value: 8
C idx: 124 - C value: 664
C idx: 125 - C value: 161
C idx: 126 - C value: 801
C idx: 127 - C value: 373
C idx: 128 - C value: 69
C idx: 129 - C value: 90
C idx: 130 - C value: 53
C idx: 131 - C value: 279
C idx: 132 - C value: 447
C idx: 133 - C value: 539
C idx: 134 - C value: 128
C idx: 135 - C value: 334
C idx: 136 - C value: 904
C idx: 137 - C value: 729
C idx: 138 - C value: 1022
C idx: 139 - C value: 614
C idx: 140 - C value: 456
C idx: 141 - C value: 176
C idx: 142 - C value: 927
```

```
C idx: 143 - C value: 215
C idx: 144 - C value: 840
C idx: 145 - C value: 139
C idx: 146 - C value: 945
C idx: 147 - C value: 981
C idx: 148 - C value: 11
C idx: 149 - C value: 560
C idx: 150 - C value: 917
C idx: 151 - C value: 561
C idx: 152 - C value: 152
C idx: 153 - C value: 252
C idx: 154 - C value: 918
C idx: 155 - C value: 429
C idx: 156 - C value: 557
C idx: 157 - C value: 973
C idx: 158 - C value: 43
C idx: 159 - C value: 249
C idx: 160 - C value: 178
C idx: 161 - C value: 466
C idx: 162 - C value: 23
C idx: 163 - C value: 826
C idx: 164 - C value: 394
C idx: 165 - C value: 987
C idx: 166 - C value: 558
C idx: 167 - C value: 497
C idx: 168 - C value: 700
C idx: 169 - C value: 466
C idx: 170 - C value: 90
C idx: 171 - C value: 895
C idx: 172 - C value: 635
C idx: 173 - C value: 593
C idx: 174 - C value: 982
C idx: 175 - C value: 104
C idx: 176 - C value: 281
C idx: 177 - C value: 482
C idx: 178 - C value: 92
C idx: 179 - C value: 848
C idx: 180 - C value: 803
C idx: 181 - C value: 454
C idx: 182 - C value: 915
C idx: 183 - C value: 107
```

```
C idx: 184 - C value: 119
C idx: 185 - C value: 450
C idx: 186 - C value: 49
C idx: 187 - C value: 634
C idx: 188 - C value: 740
C idx: 189 - C value: 849
C idx: 190 - C value: 64
C idx: 191 - C value: 807
C idx: 192 - C value: 1008
C idx: 193 - C value: 775
C idx: 194 - C value: 59
C idx: 195 - C value: 906
C idx: 196 - C value: 564
C idx: 197 - C value: 149
C idx: 198 - C value: 69
C idx: 199 - C value: 253
C idx: 200 - C value: 982
C idx: 201 - C value: 300
C idx: 202 - C value: 831
C idx: 203 - C value: 966
C idx: 204 - C value: 148
C idx: 205 - C value: 377
C idx: 206 - C value: 768
C idx: 207 - C value: 839
C idx: 208 - C value: 657
C idx: 209 - C value: 330
C idx: 210 - C value: 691
C idx: 211 - C value: 551
C idx: 212 - C value: 860
C idx: 213 - C value: 209
C idx: 214 - C value: 671
C idx: 215 - C value: 95
C idx: 216 - C value: 272
C idx: 217 - C value: 770
C idx: 218 - C value: 481
C idx: 219 - C value: 7
C idx: 220 - C value: 19
C idx: 221 - C value: 601
C idx: 222 - C value: 250
C idx: 223 - C value: 82
C idx: 224 - C value: 617
```

```
C idx: 225 - C value: 717
C idx: 226 - C value: 928
C idx: 227 - C value: 461
C idx: 228 - C value: 89
C idx: 229 - C value: 981
C idx: 230 - C value: 454
C idx: 231 - C value: 325
C idx: 232 - C value: 1
C idx: 233 - C value: 600
C idx: 234 - C value: 299
C idx: 235 - C value: 572
C idx: 236 - C value: 612
C idx: 237 - C value: 204
C idx: 238 - C value: 261
C idx: 239 - C value: 270
C idx: 240 - C value: 441
C idx: 241 - C value: 671
C idx: 242 - C value: 895
C idx: 243 - C value: 632
C idx: 244 - C value: 107
C idx: 245 - C value: 757
C idx: 246 - C value: 939
C idx: 247 - C value: 943
C idx: 248 - C value: 26
C idx: 249 - C value: 596
C idx: 250 - C value: 713
C idx: 251 - C value: 540
C idx: 252 - C value: 232
C idx: 253 - C value: 367
C idx: 254 - C value: 296
C idx: 255 - C value: 38
C idx: 256 - C value: 749
C idx: 257 - C value: 507
C idx: 258 - C value: 610
C idx: 259 - C value: 162
C idx: 260 - C value: 902
C idx: 261 - C value: 521
C idx: 262 - C value: 270
C idx: 263 - C value: 750
C idx: 264 - C value: 833
C idx: 265 - C value: 336
```

```
C idx: 266 - C value: 704
C idx: 267 - C value: 99
C idx: 268 - C value: 922
C idx: 269 - C value: 596
C idx: 270 - C value: 489
C idx: 271 - C value: 953
C idx: 272 - C value: 716
C idx: 273 - C value: 114
C idx: 274 - C value: 362
C idx: 275 - C value: 854
C idx: 276 - C value: 3
C idx: 277 - C value: 990
C idx: 278 - C value: 271
C idx: 279 - C value: 560
C idx: 280 - C value: 866
C idx: 281 - C value: 523
C idx: 282 - C value: 62
C idx: 283 - C value: 646
C idx: 284 - C value: 560
C idx: 285 - C value: 272
C idx: 286 - C value: 304
C idx: 287 - C value: 760
C idx: 288 - C value: 315
C idx: 289 - C value: 778
C idx: 290 - C value: 741
C idx: 291 - C value: 176
C idx: 292 - C value: 772
C idx: 293 - C value: 533
C idx: 294 - C value: 675
C idx: 295 - C value: 9
C idx: 296 - C value: 573
C idx: 297 - C value: 14
C idx: 298 - C value: 341
C idx: 299 - C value: 67
C idx: 300 - C value: 29
C idx: 301 - C value: 959
C idx: 302 - C value: 838
C idx: 303 - C value: 813
C idx: 304 - C value: 204
C idx: 305 - C value: 376
C idx: 306 - C value: 270
```

```
C idx: 307 - C value: 388
C idx: 308 - C value: 991
C idx: 309 - C value: 791
C idx: 310 - C value: 320
C idx: 311 - C value: 69
C idx: 312 - C value: 180
C idx: 313 - C value: 610
C idx: 314 - C value: 189
C idx: 315 - C value: 236
C idx: 316 - C value: 748
C idx: 317 - C value: 85
C idx: 318 - C value: 662
C idx: 319 - C value: 737
C idx: 320 - C value: 218
C idx: 321 - C value: 647
C idx: 322 - C value: 905
C idx: 323 - C value: 828
C idx: 324 - C value: 422
C idx: 325 - C value: 779
C idx: 326 - C value: 27
C idx: 327 - C value: 200
C idx: 328 - C value: 995
C idx: 329 - C value: 308
C idx: 330 - C value: 828
C idx: 331 - C value: 564
C idx: 332 - C value: 908
C idx: 333 - C value: 788
C idx: 334 - C value: 655
C idx: 335 - C value: 68
C idx: 336 - C value: 630
C idx: 337 - C value: 787
C idx: 338 - C value: 999
C idx: 339 - C value: 135
C idx: 340 - C value: 1
C idx: 341 - C value: 899
C idx: 342 - C value: 994
C idx: 343 - C value: 262
C idx: 344 - C value: 408
C idx: 345 - C value: 592
C idx: 346 - C value: 704
C idx: 347 - C value: 708
```

```
C idx: 348 - C value: 535
C idx: 349 - C value: 867
C idx: 350 - C value: 70
C idx: 351 - C value: 385
C idx: 352 - C value: 976
C idx: 353 - C value: 831
C idx: 354 - C value: 1019
C idx: 355 - C value: 257
C idx: 356 - C value: 562
C idx: 357 - C value: 416
C idx: 358 - C value: 868
C idx: 359 - C value: 1005
C idx: 360 - C value: 799
C idx: 361 - C value: 921
C idx: 362 - C value: 175
C idx: 363 - C value: 669
C idx: 364 - C value: 85
C idx: 365 - C value: 877
C idx: 366 - C value: 760
C idx: 367 - C value: 134
C idx: 368 - C value: 704
C idx: 369 - C value: 824
C idx: 370 - C value: 10
C idx: 371 - C value: 809
C idx: 372 - C value: 258
C idx: 373 - C value: 793
C idx: 374 - C value: 692
C idx: 375 - C value: 206
C idx: 376 - C value: 739
C idx: 377 - C value: 273
C idx: 378 - C value: 750
C idx: 379 - C value: 745
C idx: 380 - C value: 867
C idx: 381 - C value: 586
C idx: 382 - C value: 554
C idx: 383 - C value: 242
C idx: 384 - C value: 334
C idx: 385 - C value: 490
C idx: 386 - C value: 696
C idx: 387 - C value: 325
C idx: 388 - C value: 897
```

```
C idx: 389 - C value: 455
C idx: 390 - C value: 337
C idx: 391 - C value: 725
C idx: 392 - C value: 163
C idx: 393 - C value: 236
C idx: 394 - C value: 913
C idx: 395 - C value: 987
C idx: 396 - C value: 760
C idx: 397 - C value: 338
C idx: 398 - C value: 863
C idx: 399 - C value: 261
C idx: 400 - C value: 438
C idx: 401 - C value: 680
C idx: 402 - C value: 988
C idx: 403 - C value: 176
C idx: 404 - C value: 547
C idx: 405 - C value: 569
C idx: 406 - C value: 629
C idx: 407 - C value: 851
C idx: 408 - C value: 136
C idx: 409 - C value: 709
C idx: 410 - C value: 736
C idx: 411 - C value: 430
C idx: 412 - C value: 198
C idx: 413 - C value: 852
C idx: 414 - C value: 425
C idx: 415 - C value: 289
C idx: 416 - C value: 106
C idx: 417 - C value: 417
C idx: 418 - C value: 631
C idx: 419 - C value: 103
C idx: 420 - C value: 213
C idx: 421 - C value: 91
C idx: 422 - C value: 207
C idx: 423 - C value: 587
C idx: 424 - C value: 159
C idx: 425 - C value: 289
C idx: 426 - C value: 146
C idx: 427 - C value: 39
C idx: 428 - C value: 294
C idx: 429 - C value: 880
```

```
C idx: 430 - C value: 109
C idx: 431 - C value: 516
C idx: 432 - C value: 311
C idx: 433 - C value: 717
C idx: 434 - C value: 216
C idx: 435 - C value: 372
C idx: 436 - C value: 915
C idx: 437 - C value: 900
C idx: 438 - C value: 5
C idx: 439 - C value: 587
C idx: 440 - C value: 334
C idx: 441 - C value: 479
C idx: 442 - C value: 856
C idx: 443 - C value: 161
C idx: 444 - C value: 918
C idx: 445 - C value: 444
C idx: 446 - C value: 248
C idx: 447 - C value: 397
C idx: 448 - C value: 722
C idx: 449 - C value: 964
C idx: 450 - C value: 965
C idx: 451 - C value: 733
C idx: 452 - C value: 472
C idx: 453 - C value: 149
C idx: 454 - C value: 463
C idx: 455 - C value: 616
C idx: 456 - C value: 92
C idx: 457 - C value: 344
C idx: 458 - C value: 845
C idx: 459 - C value: 843
C idx: 460 - C value: 267
C idx: 461 - C value: 827
C idx: 462 - C value: 591
C idx: 463 - C value: 607
C idx: 464 - C value: 359
C idx: 465 - C value: 1020
C idx: 466 - C value: 981
C idx: 467 - C value: 940
C idx: 468 - C value: 146
C idx: 469 - C value: 839
C idx: 470 - C value: 276
```

```
 986 C idx: 471 - C value: 880
 987 C idx: 472 - C value: 627
 988 C idx: 473 - C value: 17
 989 C idx: 474 - C value: 718
 990 C idx: 475 - C value: 596
 991 C idx: 476 - C value: 427
 992 C idx: 477 - C value: 53
 993 C idx: 478 - C value: 952
 994 C idx: 479 - C value: 555
 995 C idx: 480 - C value: 905
 996 C idx: 481 - C value: 37
 997 C idx: 482 - C value: 583
 998 C idx: 483 - C value: 1022
 999 C idx: 484 - C value: 774
1000 C idx: 485 - C value: 465
1001 C idx: 486 - C value: 60
1002 C idx: 487 - C value: 829
1003 C idx: 488 - C value: 1010
1004 C idx: 489 - C value: 787
1005 C idx: 490 - C value: 677
1006 C idx: 491 - C value: 139
1007 C idx: 492 - C value: 141
1008 C idx: 493 - C value: 352
1009 C idx: 494 - C value: 698
1010 C idx: 495 - C value: 666
1011 C idx: 496 - C value: 471
1012 C idx: 497 - C value: 924
1013 C idx: 498 - C value: 983
1014 C idx: 499 - C value: 136
1015 C idx: 500 - C value: 512
1016 C idx: 501 - C value: 832
1017 C idx: 502 - C value: 397
1018 C idx: 503 - C value: 798
1019 C idx: 504 - C value: 892
1020 C idx: 505 - C value: 856
1021 C idx: 506 - C value: 1012
1022 C idx: 507 - C value: 569
1023 C idx: 508 - C value: 696
1024 C idx: 509 - C value: 928
1025 C idx: 510 - C value: 632
1026 C idx: 511 - C value: 238
```

```
1027 done
1028 Выход!
```

ЗАКЛЮЧЕНИЕ

В результате работы изучены принципы работы вычислительного комплекса Тераграф и получены практические навыки решения задач обработки множеств на основе гетерогенной вычислительной структуры.