МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

Теоретико-множественные основы интеллектуальных систем

ОТЧЁТ

По лабораторной работе №1

Выполнил:

Студент группы ИИ-22

Копанчук Евгений

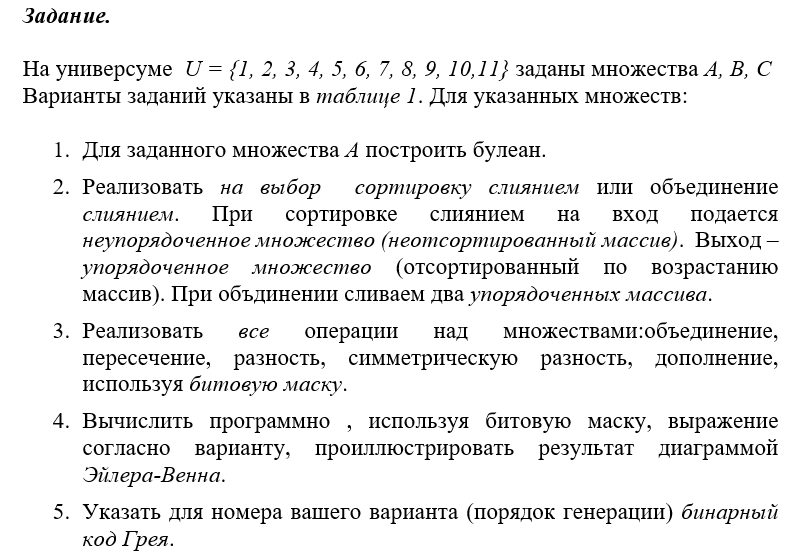
Романович

Проверил:

Глущенко Т. А.

Брест - 2022

***Вариант 10***



**

**Отчёт по программе:**

|  |  |
| --- | --- |
|  | Меню программы |
|  | При выборе первого пункта меню нам предлагают выбрать множество для построения булеана. Результат на следующем скриншоте. |
|  | При выборе второго пункта мы вводим два множества, из которых строится слияние |
|  | Третий пункт. Можем считать различные операции между множествами. |
|  | Четвёртый пункт меню – подсчёт выражения, заданного по варианту. |
|  | Пятый – Код грея. Вводим с клавиатуры число, до которого будут сгенерированы значения. |

***Код программы:***

#include <iostream>

#include <ctime>

#include <cmath>

#include <cstdlib>

using namespace std;

// Прототипы функций

void menu();

bool find(int mass[], int size, int el);

void bit(int mass[], int size, int bitmass[], int bitsize);

void print(int mass[], int size);

int ten2two(int n);

void bul(int mass[], int size);

void merge(int mass1[], int size1, int mass2[], int size2);

void intersection(int mass1[], int size1, int mass2[], int size2);

void subtraction(int mass1[], int size1, int mass2[], int size2);

void simsubtraction(int mass1[], int size1, int mass2[], int size2);

void confluence(int mass1[], int size1, int mass2[], int size2);

// Множества

int U[] = { 1,2,3,4,5,6,7,8,9,10,11 }, A[] = { 1,3,5,6,7,8 }, B[] = { 2,3,4,5,8 }, C[] = { 1,2,3,5,6 };

// Размеры множеств

int const size\_\_U = sizeof(U) / sizeof(int);

int const size\_\_A = sizeof(A) / sizeof(int);

int const size\_\_B = sizeof(B) / sizeof(int);

int const size\_\_C = sizeof(C) / sizeof(int);

int result[size\_\_U];

int size\_\_result;

int main()

{

setlocale(LC\_ALL, "Russian");

menu();

return 0;

}

////////////////////////////////////////////////////////////////////////////////

// Меню

void menu() {

int n;

cout << " Универсум:" << endl;

cout << " ----------------------------------" << endl;

cout << " U = { 1 2 3 4 5 6 7 8 9 10 11 }" << endl;

cout << " ----------------------------------" << endl;

cout << " Доступные множества:" << endl;

cout << " ----------------------------------" << endl;

cout << " A = { 1 3 5 6 7 8 }" << endl;

cout << " B = { 2 3 4 5 8 }" << endl;

cout << " C = { 1 2 3 5 6 }" << endl;

cout << " ----------------------------------" << endl;

cout << " 1. Построить буллеан для множества" << endl;

cout << " 2. Слияние множеств" << endl;

cout << " 3. Операции над множествами" << endl;

cout << " 4. Подсчёт выражения" << endl;

cout << " 5. Бинарный код Грея" << endl;

cout << " ----------------------------------" << endl;

cin >> n;

system("cls");

switch (n) {

case 1:

cout << " Введите желаемое множество (A, B или С):" << endl;

char chose;

cin >> chose;

switch (chose) {

case 'A':

system("cls");

cout << " Буллеан множества А:" << endl;

cout << " --------------------" << endl;

bul(A, size\_\_A);

cout << " --------------------" << endl;

system("pause");

system("cls");

menu();

case 'B':

system("cls");

cout << " Буллеан множества B:" << endl;

cout << " --------------------" << endl;

bul(B, size\_\_B);

cout << " --------------------" << endl;

system("pause");

system("cls");

menu();

case 'C':

system("cls");

cout << " Буллеан множества C:" << endl;

cout << " --------------------" << endl;

bul(C, size\_\_C);

cout << " --------------------" << endl;

system("pause");

system("cls");

menu();

default:

system("cls");

cout << " ------------------------------" << endl;

cout << " !Error: non-existent operation" << endl;

cout << " ------------------------------" << endl;

system("pause");

system("cls");

menu();

}

break;

case 2:

cout << " Введите два множества для слияния (A, B или С):" << endl;

char chose1, chose2;

cin >> chose1 >> chose2;

if (chose1 == 'A' && chose2 == 'B' || chose2 == 'A' && chose1 == 'B') {

system("cls");

cout << " Слияние множеств А и В:" << endl;

cout << " -----------------------" << endl;

confluence(A, size\_\_A, B, size\_\_B);

cout << " -----------------------" << endl;

system("pause");

system("cls");

menu();

}

if (chose1 == 'A' && chose2 == 'C' || chose2 == 'A' && chose1 == 'C') {

system("cls");

cout << " Слияние множеств A и C:" << endl;

cout << " -----------------------" << endl;

confluence(A, size\_\_A, C, size\_\_C);

cout << " -----------------------" << endl;

system("pause");

system("cls");

menu();

}

if (chose1 == 'B' && chose2 == 'C' || chose2 == 'B' && chose1 == 'C') {

system("cls");

cout << " Слияние множеств B и C:" << endl;

cout << " -----------------------" << endl;

confluence(B, size\_\_B, C, size\_\_C);

cout << " -----------------------" << endl;

system("pause");

system("cls");

menu();

}

case 3:

int k;

cout << " Выберете операцию" << endl;

cout << " -----------------------------------" << endl;

cout << " 1. Объединение множеств" << endl;

cout << " 2. Пересечение множеств" << endl;

cout << " 3. Разность множеств" << endl;

cout << " 4. Симметрическая разность множеств" << endl;

cout << " -----------------------------------" << endl;

cin >> k;

switch (k) {

case 1:

system("cls");

cout << " Объединение множеств A ^ B" << endl;

cout << " -----------------------" << endl;

merge(A, size\_\_A, B, size\_\_B);

cout << " -----------------------" << endl;

system("pause");

system("cls");

menu();

case 2:

system("cls");

cout << " Пересечение множеств A ^ B" << endl;

cout << " -----------------------" << endl;

intersection(A, size\_\_A, B, size\_\_B);

cout << " -----------------------" << endl;

system("pause");

system("cls");

menu();

case 3:

system("cls");

cout << " Разность множеств A ^ B" << endl;

cout << " -----------------------" << endl;

subtraction(A, size\_\_A, B, size\_\_B);

print(result, size\_\_result);

cout << " -----------------------" << endl;

system("pause");

system("cls");

menu();

case 4:

system("cls");

cout << " Симметрическая разность множеств A ^ B" << endl;

cout << " -----------------------" << endl;

simsubtraction(A, size\_\_A, B, size\_\_B);

print(result, size\_\_result);

cout << " -----------------------" << endl;

system("pause");

system("cls");

menu();

default:

system("cls");

cout << " ------------------------------" << endl;

cout << " !Error: non-existent operation" << endl;

cout << " ------------------------------" << endl;

system("pause");

system("cls");

menu();

}

case 4:

system("cls");

cout << " Подсчитать выражение (A / B) / C " << endl;

cout << " --------------------------------" << endl;

cout << " A / B --> ";

subtraction(A, size\_\_A, B, size\_\_B);

cout << " (A / B) / C --> ";

subtraction(result, size\_\_result, C, size\_\_C);

cout << " --------------------------------" << endl;

system("pause");

system("cls");

menu();

case 5: {

// Бинарный код Грея

int size;

system("cls");

cout << " Введите n для генерации кода Грея 2^n: " << endl;

cin >> size;

cout << " --------------------------------------" << endl;

int\* a = new int[21];

for (int i = 0; i < size + 1; i++) a[i] = 0;

int j = 0;

for (;;) {

int\* mass = new int[20];

for (int i = size-1, l = 0; i >= 0; i--) {

l++;

mass[l] = a[i];

cout << mass[l] << " ";

}

cout << endl;

a[size] = 1 - a[size];

if (a[size] == 1) j = 0;

else {

for (j = 0; j < size; j++) {

if (a[j] == 1) {

j += 1;

break;

}

}

}

if (j == size) {

break;

}

a[j] = 1 - a[j];

delete[] mass;

}

delete[] a;

cout << " --------------------------------------" << endl;

system("pause");

system("cls");

menu();

}

default:

system("cls");

cout << " ------------------------------" << endl;

cout << " !Error: non-existent operation" << endl;

cout << " ------------------------------" << endl;

system("pause");

system("cls");

menu();

}

}

// Поиск элемента el в множестве mass размером size

bool find(int mass[], int size, int el) {

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

if (mass[i] == el) {

return true;

}

}

return false;

}

// Битовая маска bitmass размером bitsize(универсум) множества mass размером size

void bit(int mass[], int size, int bitmass[], int bitsize) {

for (int i = 1; i <= bitsize; i++) {

if (find(mass, size, i)) bitmass[i - 1] = true;

else bitmass[i - 1] = false;

}

}

// Вывод множества mass размером size

void print(int mass[], int size) {

cout << "{ ";

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

cout << mass[i] << " ";

}

cout << "}";

cout << endl;

}

// Перевод числа n из десятичной в двоичную систему

int ten2two(int n) {

int k = 0;

for (int i = sizeof(n) \* 8 - 1; i >= -1; --i)

{

k += (int) pow(10, i) \* ((n >> i) & 1);

}

return k;

}

// Булеан множества

void bul(int mass[], int size) {

for (int i = 0, count = (int) pow(2, size); i < count; i++) {

int num = ten2two(i);

cout << "{ ";

for (int j = 0; j < size; j++) {

if (num % 10) {

cout << mass[j] << " ";

}

num /= 10;

}

cout << "}" << endl;

}

}

// Слияние множеств

void confluence(int mass1[], int size1, int mass2[], int size2) {

int i = 0, j = 0;

cout << "{ ";

while (i < size1 || j < size2) {

if (mass1[i] < mass2[j]) {

cout << mass1[i] << " ";

i++;

}

else

if (mass1[i] > mass2[j]) {

cout << mass2[j] << " ";

j++;

}

else

if (mass1[i] == mass2[j]) {

cout << mass1[i] << " ";

cout << mass2[j] << " ";

i++;

j++;

}

}

cout << "}" << endl;

}

// Объединение множеств

void merge(int mass1[], int size1, int mass2[], int size2) {

int bitmass1[size\_\_U], bitmass2[size\_\_U];

bit(mass1, size1, bitmass1, size\_\_U);

bit(mass2, size2, bitmass2, size\_\_U);

cout << "{ ";

int l = 0;

for (int i = 0; i < size\_\_U; i++) {

if (bitmass1[i] | bitmass2[i]) {

result[l] = U[i];

l++;

cout << U[i] << " ";

}

else result[i] = 0;

}

size\_\_result = l;

cout << "}" << endl;

}

// Пересечение множеств

void intersection(int mass1[], int size1, int mass2[], int size2) {

int bitmass1[size\_\_U], bitmass2[size\_\_U];

bit(mass1, size1, bitmass1, size\_\_U);

bit(mass2, size2, bitmass2, size\_\_U);

cout << "{ ";

int l = 0;

for (int i = 0; i < size\_\_U; i++) {

if (bitmass1[i] & bitmass2[i]) {

result[l] = U[i];

l++;

cout << U[i] << " ";

}

}

size\_\_result = l;

cout << "}" << endl;

}

// Разность множеств

void subtraction(int mass1[], int size1, int mass2[], int size2) {

int bitmass1[size\_\_U], bitmass2[size\_\_U];

bit(mass1, size1, bitmass1, size\_\_U);

bit(mass2, size2, bitmass2, size\_\_U);

cout << "{ ";

int l = 0;

for (int i = 0; i < size\_\_U; i++) {

if (bitmass2[i] != 1 && bitmass1[i] == 1) {

result[l] = U[i];

l++;

cout << U[i] << " ";

}

}

size\_\_result = l;

cout << "}" << endl;

}

// Симметрическая разность множеств

void simsubtraction(int mass1[], int size1, int mass2[], int size2) {

int bitmass1[size\_\_U], bitmass2[size\_\_U];

bit(mass1, size1, bitmass1, size\_\_U);

bit(mass2, size2, bitmass2, size\_\_U);

cout << "{ ";

int l = 0;

for (int i = 0; i < size\_\_U; i++) {

if (bitmass1[i] != bitmass2[i]) {

result[l] = U[i];

l++;

cout << U[i] << " ";

}

}

size\_\_result = l;

cout << "}" << endl;

}