**МИНОБРНАУКИ РОССИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа № 2

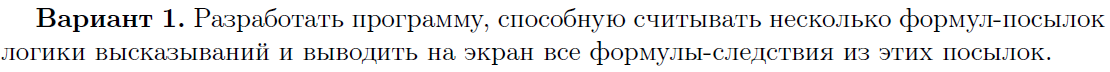
дисциплина: Математическая логика и теория алгоритмов

ЛОГИКА ПРЕДИКАТОВ

|  |  |
| --- | --- |
|  | Выполнил: ст. группы ПВ-21  Браткова Ирина Олеговна |

Белгород 2017

**ПРАКТИЧЕСКОЕ ЗАДАНИЕ**



#include <iostream> #include <set> #include <string> #include <cmath> #include <vector> #include <stack>

std::string opz(std::string s); // перевод строки в обратную польскую запись

void t\_per (std::string s); // кол-во переменных

std::set<char> per; // множество перменных

void virah(std::string s); //создание выражений

std::vector <std::string> vir; // хранит выражения

void output(); // вывод следтвий и з посылок

void soch(int i, int b, int N, int k); // заполняемое место, минимальный элемент, из N по k

void out(); // вывод выражения получено в рез-те сочетания

void inputt (std::string \*s, int n); // ввод массива строк

void kol\_per (std::string \*s, int n); //подсчет количества переменны

void make\_vir (std::string \*s, int n); //создание выражений

std::vector<int>C;

int main()

{

std::cout<<"Strok "; int n;

std::cin >> n; //ввод количества строк

std::string \*vir = new std::string [n]; //выделение памяти под массив строк

inputt(vir, n); // ввод массива сток

kol\_per(vir, n); // подсчет кол-ва переменных

make\_vir(vir, n); //создание выражения

std::cout<< " " << std::endl; std::cout<<"Rez " << std::endl; output(); //вывод результата

}

void inputt (std::string \*s, int n)

{ for (int i=0; i<n; i++) std::cin >> s[i];

}

void kol\_per (std::string \*s, int n)

{ for (int i=0; i<n; i++) t\_per(s[i]); }

void make\_vir (std::string \*s, int n)

{ for (int i=0; i<n; i++) virah(s[i]); }

void t\_per (std::string s) // кол-во переменных

{

for (int i = 0; i<s.length(); i++)

{

if (((s[i] >= 'A') &&(s[i]<='Z'))|| (((s[i] >= 'a') && (s[i] <= 'z')) && s[i]!='v'))

{// 1 если элемент есть в множестве, иначе 0

if ( !per.count(s[i])) per.insert(s[i]);

}

}

}

void virah(std::string s)

{

std::set<char> pod\_mn;

for(int i=0; i<s.length(); i++)

{if (((s[i] >= 'A') &&(s[i]<='Z'))|| (((s[i] >= 'a') && (s[i] <= 'z')) && s[i]!='v')){ pod\_mn.insert(s[i]); } }

if (pod\_mn.size() == per.size())

{ vir.push\_back(s); //расширяет на один элемент и записывает в него s

return; }

int dob = per.size() - pod\_mn.size() ; //сколько добавить?????????

int st = pow(2, dob); std::string \*sn = new std::string [st];

for (int i=0; i<st; i++) { sn[i]=s; }

int m=0; for (std::set<char>::iterator p=per.begin(); p!=per.end(); p++) // итератор по полному множеству

{

if (!pod\_mn.count(\*p))

{ int n = 0; bool f = 1;

for (int j=0; j<st; j++)

{

if(f){ sn[j]+='v'; sn[j]+=\*p; }

else { sn[j]+="v!"; sn[j]+=\*p; }

n++;

if (n==pow(2, m)) { n=0; f=!f; }

m++;

}

}

}

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

{

vir.push\_back(sn[i]);

}

}

void output()

{

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

{

C.push\_back(i); soch(0,0,vir.size(),i);

}

}

void soch(int i, int b, int N, int k)

{

for (int j = b; j < N-k+i; j++)

{

C[i] = j;

if (i == k) { out(); }

else soch(i+1, j+1, N, k);

}

}

void out ()

{

for (unsigned int i = 0; i < C.size(); i++)

{

if(i+1==C.size())

std::cout << vir[C[i]];

else

std::cout << vir[C[i]]<<" & ";

}

std::cout<<std::endl;

}