**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 №3

Дисциплина: Математическая логика и теория алгоритмов

по теме «Формальные теории»

Выполнил: ст. группы ВТ-22  
Макаров Даниил Сергеевич

Проверил: Куценко Д.А.

Белгород 2018

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

int \*\*matr\_init(int str, int tab) {

int \*\*matr = (int \*\*)malloc(str \* sizeof(int \*));

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

matr[i] = (int \*)malloc(tab \* sizeof(int));

return matr;

}

//ввод КНФ пользователем в заданном программой виде

int \*\*input\_knf(int \*m, int \*n) {

int t1=0,t2=0;

char str[50];

printf("KNF - ");

scanf("%s",str );

int str\_size=strlen(str);

int max=0;

//подсчет количества строк и столбцев в таблице которая будет содержать кнф

for(int i=0;i<str\_size;i++){

if(str[i]=='+') t1++;

if(str[i]==')') {

if(t1>max) max=t1;

t1=0;

}

if(str[i]=='\*') t2++;

}

\*n=max+1;

\*m=t2+1;

int\*\* matr=matr\_init(\*m,\*n);

//считываем кнф из строки и записываем значения в таблицу

int i=0,j=0;// i - строка, j - столбец

for(int k=0;k<str\_size;k++){

switch (str[k]) {

case '!':

matr[i][j]=-1;

j++;

k++;

break;

case '\*':

i++;

j=0;

break;

case '0':

matr[i][j]=0;

j++;

break;

case '(':

break;

case ')':

break;

case '+':

break;

default:

matr[i][j]=1;

j++;

break;

}

}

return matr;

}

//вывод КНФ на экран

void output\_knf(int \*\*a, int m, int n) {

int i = 0, j = 0;

for (i = 0; i < m; i++) {

printf("(");

j = 0;

while ((a[i][j] == 0) && (j < n))

j++;

if (j < n) {

if (a[i][j] == -1)

printf("!%c", 'A'+j);

else

printf("%c", 'A'+j);

}

j++;

for (j; j < n; j++) {

if (a[i][j] == -1)

printf("|!%c", 'A'+j);

if (a[i][j] == 1)

printf("|%c", 'A'+j);

}

printf(")");

if (i + 1 < m) printf("&");

}

printf("\n");

}

int next(int \*a, int n, int ind) {

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

if (a[i])

return 1;

return 0;

}

int scobka(int\* a, int n) {

int i, f = 1;

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

if ((a[i] == 1)) {

printf("%c", 'A' + i);

if (next(a, n, i + 1)) printf("|");

f = 0;

}

if ((+a[i] == -1)) {

printf("!%c", 'A' + i);

if (next(a, n, i + 1)) printf("|");

f = 0;

}

}

if (f) {

printf("%c", '0');

return 0;

}

return 1;

}

int one\_dif(int \*a, int \*b, int n) {

int flag = 0, i;

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

if (a[i] == -b[i] && a[i] != 0) flag++;

return (flag == 1) ? 1 : 0;

}

void skleika(int \*a, int \*b, int \*rez, int n) {

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

if (a[i] \* b[i] == -1 || (!a[i] && !b[i]))

rez[i] = 0;

else if ((a[i] \* b[i] == 1 && a[i] == 1) || (a[i] \* b[i] == 0 && ((a[i] == 0 && b[i] == 1) || (a[i] == 1 && b[i] == 0))))

rez[i] = 1;

else

rez[i] = -1;

}

}

void rez(int \*\*a, int m, int n) {

int\* rez1 =(int\*)malloc(sizeof(int)\*n);

int i, j, f = 1;

for (j = 0; j < n; j++)

rez1[j] = a[0][j];

for (i = 0; i < m && f; i++) {

if (a[i][0] != 5) {

for (j = 0; j < m && f; j++)

if (one\_dif(rez1, a[j], n) && a[j][0] != 5) {

printf("(");

scobka(rez1, n);

printf(")");

printf("&");

printf("(");

scobka(a[j], n);

printf(")");

printf("=>");

skleika(rez1, a[j], rez1, n);

f = scobka(rez1, n);

printf("\n");

}

}

}

}

int main() {

int n=0,m=0,i = 0;

int\*\* a=input\_knf(&m ,&n );

printf("\n");

rez(a, m, n);

return 0;

}