**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

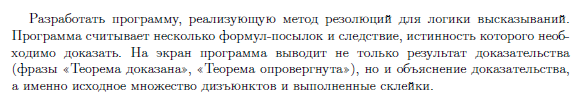
дисциплина: Математическая логика и теория алгоритмов

ФОРМАЛЬНЫЕ ТЕОРИИ

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

Белгород 2017

**ПРАКТИЧЕСКОЕ ЗАДАНИЕ**

****

#include <stdio.h> #include <stdlib.h> #include "stackMas.h" #include <string.h>

typedef struct { int \*\*matr; int numDiz, numCh, mem; }diz;

int Perf\_Expr(char \*s, int \*d,int \*p,int n)

{

unsigned i; char c[strlen(s)];

Stack st; InitStack(&st);

for(i=0;i<n;i++) p[i]=0;

for(i=0;i<=strlen(s);i++) c[i]=s[i];

for(i=0;i<strlen(s);i++)

if((s[i]>=1)&&(s[i]<=26))

if(d[s[i]-1])

{ c[i]='+'; p[s[i]-1]=-1; }

else

{ c[i]='-'; p[s[i]-1]=1; }

for(i=0;i<strlen(c);i++)

{

if((c[i]=='+')||(c[i]=='-')) PutStack(&st,c[i]);

else if(c[i]=='&')

PutStack(&st,(GetStack(&st)=='+'?1:0)&(GetStack(&st)=='+'?1:0)?'+':'-');

else if(c[i]=='|')

PutStack(&st,(GetStack(&st)=='+'?1:0)|(GetStack(&st)=='+'?1:0)?'+':'-');

else if(c[i]=='!')

PutStack(&st,GetStack(&st)=='+'?'-':'+');

}

return GetStack(&st)=='+'?1:0;

}

int IsSame(int \*a, int \*b, int n)

{

int i; for(i=0;i<n;i++) if(a[i]!=b[i]) return 0;

return 1;

}

void Copy(int \*a, int \*b, int n)

{ int i; for(i=0;i<n;i++) a[i]=b[i]; }

void InitDiz(diz \*D,int m, int k)

{

int i; D->numCh=k; D->numDiz=0; D->mem=m;

D->matr=(int\*\*)calloc(D->mem,sizeof(int\*));

for(i=0;i<D->mem;i++) D->matr[i]=(int\*)calloc(D->numCh,sizeof(int));

}

char whichSymbol(int d, char \*ch)

{ int i; for(i=0;i<26;i++) if(ch[i]==d) break; return 'A'+i; }

void PrintDiz(int \*d, int m, char \*ch)

{

int i=0;

while(!d[i]) i++;

if(i<m)

{

if(d[i]==1) printf("%c",whichSymbol(i+1,ch));

else printf("!%c",whichSymbol(i+1,ch));

for(i=i+1;i<m;i++)

{

if(d[i])

if(d[i]==1) printf("|%c",whichSymbol(i+1,ch));

else printf("|!%c",whichSymbol(i+1,ch));

}

}

else

printf("0");

}

int Add(diz \*D, int \*p)

{

int i;

for(i=0;i<D->numDiz;i++) if(IsSame(D->matr[i],p,D->numCh)) return 0;

D->numDiz++;

if(D->numDiz>D->mem)

{

D->mem++; D->matr=(int \*\*)realloc(D->matr,D->mem\*sizeof(int\*));

D->matr[D->mem-1]=(int\*)calloc(D->numCh,sizeof(int));

}

Copy(D->matr[D->numDiz-1],p,D->numCh); return 1;

}

void addDiz(char \*s,int n, diz \*Dizs)

{

int d[n],p[n]; void subs\_in(int i)

{

int x;

for(x=0;x<=1;x++)

{

d[i]=x;

if(i==n-1)

{ if(!Perf\_Expr(s,d,p,n)) Add(Dizs,p); }

else subs\_in(i+1);

}

}

subs\_in(0); }

void ToOznaczenia(char \*s, char \*pol, char \*ch, char \*k)

{

unsigned i=0,j=0;

void Ozn\_in()

{

char a=0,m=0;

Stack st; InitStack(&st);

while(s[i]!=')' && s[i])

{

if((s[i]>='A')&&(s[i]<='Z'))

{

a++;

if(!ch[s[i]-'A']) ch[s[i]-'A']=++(\*k);

pol[j++]=ch[s[i]-'A'];

if(m)

{ pol[j++]='!'; m=0; }

if(a==2)

{ pol[j++]=GetStack(&st); a=1; }

}

else if((s[i]=='&')||(s[i]=='|'))

PutStack(&st,s[i]);

else if(s[i]=='!')

m=1;

else if(s[i]=='(')

{

i++; Ozn\_in();

if(m)

{

pol[j++]='!';

m=0;

}

a++;

}

i++;

}

while(!EmptyStack(&st))

pol[j++]=GetStack(&st);

}

i=0; Ozn\_in(); pol[j]='\0';

}

int Skl(int \*a, int \*b, int \*c, int n)

{

int i,p=1;

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

if(a[i]!=b[i])

if(a[i]==-b[i])

if(p) { c[i]=0; p=0; }

else return 0;

else return 0;

else c[i]=a[i];

return 1;

}

int isNull(int \*a, int n)

{ int i; for(i=0;i<n;i++) if(a[i]) return 0; return 1; }

void Solve(diz \*D,char \*ch)

{

int \*t=calloc(D->numCh,sizeof(int)),p=1,i,j;

while(p)

{

p=0;

for(i=0;i<D->numDiz-1;i++)

for(j=i+1;j<D->numDiz;j++)

if(Skl(D->matr[i],D->matr[j],t,D->numCh))

{

if(Add(D,t))

{

PrintDiz(D->matr[i],D->numCh,ch);

printf(" + ");

PrintDiz(D->matr[j],D->numCh,ch);

printf(" = ");

PrintDiz(t,D->numCh,ch);

if(isNull(t,D->numCh)) return;

printf("\n");

p=1;

}

}

}

}

int main()

{

setlocale(0,""); unsigned m,i; diz D;

char s[300],tmp[50],ch[26],k=0;

for(i=0;i<26;i++) ch[i]=0; // Инициализция массива переменных 0

printf("введите количество посылок\n");

scanf("%i", &m); char pol[m+1][50];

printf("введите посылки\n");

for(i=0;i<m;i++) { scanf("%s",s); ToOznaczenia(s,pol[i],ch,&k); }

s[0]=0;

strcat(s,"!(");

printf("введите результат\n");

scanf("%s",tmp);

strcat(s,tmp);

strcat(s,")");

printf("\n");

ToOznaczenia(s,pol[i],ch,&k);

InitDiz(&D,m+1,k);

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

addDiz(pol[i],k,&D);

for(i=0;i<D.numDiz;i++) { PrintDiz(D.matr[i],D.numCh,ch); printf(" "); }

printf("\n");

Solve(&D,ch);

return 0;

}