

## Условие задачи

Дано правильное арифметическое выражение, состоящее из букв (латинский алфавит A..Z, буквы используются без индексов), цифр, знаков арифметических операций и круглых скобок, записанное в общепринятой форме. Составить программу, удаляющую из выражения лишние пары скобок, не влияющие на порядок выполнения операций.

Критерии корректности введенного выражения:

1. Начало выражение - число, символ числа, "(" либо "-"
2. Конец выражения - число, символ числа, ")"
3. Символьно-числовые одночлены могут вводиться без знака "\*" между числами и символами
4. Между символьно-числовыми многочленами и скобочным выражением знак операции обязателен (знак "\*" можно опускать).
5. Между скобочными выражениями знак операции обязателен (знак "\*" можно опускать)
6. Не допускается сдвоения знаков операций
7. Парность скобок обязательна
8. Каждая пара скобок вводится последовательно - сначала "(" потом ")"
9. Допускается любая глубина вложенности скобок
10. Выражение внутри скобок вводится по правилам описанным для цельного выражения

Согласно условия задачи для удаления скобок дается корректное выражение. Выше определены правила корректности выражения

## Текст решения на КП

```
MODULE Etab2b1;
  IMPORT Log:=StdLog, In:=i21sysIn;
  PROCEDURE skobki*;
    CONST
      MD=100;
    VAR
      V:ARRAY MD OF CHAR;
      N, K, K2:INTEGER;
    PROCEDURE vvod(OUT V:ARRAY OF CHAR; OUT N:INTEGER):BOOLEAN;
      VAR
        e:INTEGER;
    BEGIN
      In.Open;
      N:=-1;
      WHILE (In.Done) DO
        INC(N);
```

```

    IF N>=MD THEN
        Log.String('Слишком много знаков в выражении');
        RETURN FALSE;
    END;
    In.Char(V[N]);
END;
V[N]:='+';
Log.Ln;Log.String('Вводное выражение');Log.Ln;
e:=0;
WHILE e<N DO
    Log.Char(V[e]);
    INC(e);
END;
RETURN TRUE;
END vvod;

PROCEDURE analiz (IN V:ARRAY OF CHAR;N:INTEGER):BOOLEAN;
BEGIN
    RETURN TRUE ;
END analiz;

PROCEDURE proverkaV(IN V:ARRAY OF CHAR;VAR K,K2:INTEGER):BOOLEAN;
VAR
    AD:BOOLEAN;
    UR,e:INTEGER;
BEGIN
    IF V[K+1]='-' THEN
        K2:=K+1;
    ELSE
        K2:=K;
    END;

    UR:=-1;AD:=FALSE;
    WHILE UR<0 DO
        INC(K2);
        IF V[K2]='(' THEN
            DEC(UR);
        ELSIF V[K2]=')' THEN
            INC(UR);
        END;

        IF (UR=-1) & ((V[K2]='+') OR (V[K2]='-')) THEN
            AD:=TRUE;
        END;
    END;

    IF AD THEN
        (*Поиск признака аддитивности в соседнем скобочном выражении (на случай
пустых скобок. *)
        UR:=1;
        IF V[K2+UR]="(" THEN
            INC(UR);

```

```

WHILE (V[K2+UR]='(')OR(V[K2+UR]=')') DO
  INC(UR);
END;
IF V[K2+UR]# '+' THEN
  IF V[K2+UR]# '-' THEN
    UR:=1;
  ELSIF V[K2+UR-1]='(' THEN
    UR:=1;
  END;
END;
END;
(*первый символ за пределами выражение должен быть '+'*)
IF (V[K2+UR]='+')OR(V[K2+UR]='-')OR(V[K2+UR]='(') THEN
  IF K=0 THEN
    RETURN FALSE;
  END;
  IF V[K+1]='-' THEN
    IF V[K-1]='(' THEN
      RETURN FALSE;
    ELSE
      RETURN TRUE;
    END;
  END;
  IF (V[K-1]='(')OR(V[K-1]='+') THEN
    RETURN FALSE;
  END;
END;
RETURN TRUE;

ELSE
  IF V[K+1]='-' THEN
    IF (K#0) & (V[K-1]# '(') THEN
      RETURN TRUE;
    END;
  END;
  RETURN FALSE;
END;
END proverkaV;

PROCEDURE udal_sk(VAR V:ARRAY OF CHAR;K,K2:INTEGER;VAR N:INTEGER);
VAR
  SM:INTEGER; (*Шаг смещения*)
BEGIN
  SM:=1;
  K:=K+SM;
  WHILE K<N DO
    IF K#K2 THEN
      V[K-SM]:=V[K];
    ELSE
      INC(SM);
    END;
  INC(K);

```

```

    END;
    DEC (N, SM) ;
    V[N] := '+' ;
    END udal_sk;
BEGIN
    IF ~vvod(V,N) THEN
        RETURN;
    END;
    K:=0;

    WHILE K<N-1 DO
        IF (V[K]='(' & (~proverkaV(V,K,K2))) THEN
            udal_sk(V,K,K2,N); DEC(K);
            (*При удалении скобок K должен остаться неизменным*)
        END;
        INC(K);
    END;
    K:=0;
    Log.Ln; Log.String('Вывод'); Log.Ln;
    WHILE K<N DO
        Log.Char(V[K]);
        INC(K);
    END;

    END skobki;

END Etap2b1.

```

## Текст на языке Си

```

#include <iostream>
#include <windows.h>

using namespace std;

int proverka(char*, int, int&);
void ud_skobki(char*, int, int, int&);

main ()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    cout<<"Удаление лишних скобок\n\n";
    char V[150];
    int N=0;
    cout<<"Введите правильное выражение\"B\""\n\n\t";
    cin>>V;
    while(V[N]!='\0') N++;
    V[N]='+';
    int k2;

```

```

    for (int k=0;k<N-1;k++)
        if ((V[k]=='(')&&!proverka(V, k, k2))
            { ud_skobki(V, k, k2, N);
              k--;
            }
    cout<<"Вывод\n\n";
    for(int k=0;k<N;k++) cout<<V[k];
    cin.get();
    cin.get();
    return(0);
}
void ud_skobki(char* V,int k,int k2,int & N)
{ int sm=1;//смещение
  k++;
  while(k<N)
  {
    if(k!=k2) V[k-sm]=V[k];
    else sm++;
    k++;
  }
  N-=sm;
}

int proverka(char*V,int k,int& k2)
{ if(V[k+1]=='-') k2=k+1;
  else k2=k;

  int ur=-1;
  int AD=0;
  while(ur<0)
  { k2++;
    if(V[k2]=='(') ur--;
    else if (V[k2]==')') ur++;
    else if (ur==-1&&(V[k2]=='+'||V[k2]=='-')) AD=1;

  }
  if(AD)
  { ur=1;

//////////Если выражение введено согласно правил выше,
//////////то пустые скобки исключены (правило 5,6)
*/
    if(V[k2+ur]=='+'||V[k2+ur]=='-'||V[k2+ur]==')')
        if((k==0)|| (V[k-1]=='(')|| (V[k+1]!='-'&&V[k-1]=='+')) return(0);
        else return(1);
    }
  else if((k>0)&&(V[k+ur]=='-')&&(V[k-ur]=='(')) return(1);
  else return (0);
}

```