

ТОДМ ЛР № 3 Построение минимальной ДНФ логической функции.

1. Построить минимальную ДНФ логической функции F. F равна инвертированному вашему номеру в журнале. Таблица истинности должна содержать 4 переменные. (можно в тетради, в отчет добавляем фото из тетради, можно в word, excel).
Пример построения – «ЛР № 3.xlsx».
Пример оформления отчета – «ЛР №3 ТОДВ.docx».

В результате пункта 1. Вы получаете F, например $F = AB + \bar{A}C$

2. Реализовать алгоритм построения таблицы истинности полученной F. Перебор всех возможных перестановок логических переменных может быть выполнен следующими способами:
 - a. итерационным методом (3 балла)
 - b. методом рекурсивного перебора (4 балла)
 - c. с помощью поразрядного сумматора (3 балла)
 - d. с помощью рекурсивного поразрядного сумматора (5 баллов)
 - e. с использованием рекурсивного алгоритма перевода чисел из 10 в 2 систему счисления (3 балла)

Пример программы, реализующей не рекурсивный поразрядный сумматор:

```
var s1,s2,r: string;
procedure sum1(c1,c2,ppred : char; var r,p : char);
begin
  r := '0';
  p := '0';

  if ppred = '0' then begin
    if (c1=c2) and (c1='1') then begin r := '0'; p := '1'; end;
    if c1 <> c2 then r := '1'
  end
  else begin
    if (c1=c2) and (c1='0') then r := '1';
    if (c1=c2) and (c1='1') then begin r := '1'; p := '1' end;
    if c1 <> c2 then p := '1';
  end;
end;

procedure sumN(s1,s2 : string; var r : string);
var i : integer;
    n : integer;
    ppred : char;
begin
  n := abs(length(s1) - length(s2));
  if length(s1) < length(s2)
  then for i := 1 to n do s1 := '0'+s1
  else for i := 1 to n do s2 := '0'+s2;

  n := length(s1);
  for i := 1 to n do r := r + '0';

  ppred := '0';
  for i := n downto 1 do
    sum1(s1[i],s2[i],ppred, r[i], ppred);
    if ppred = '1' then r := ppred + r
  end;

begin
  write('Введите первое слагаемое: '); readln(s1);
  write('Введите второе слагаемое: '); readln(s2);
  sumN(s1,s2,r);
  writeln('Ответ: ',r);
end.
```

3. Задачи *

- a. Длинная арифметика – реализовать рекурсивный сумматор для системы счисления N
- b. Длинная арифметика – реализовать алгоритмы, выполняющие сложение в дополнительном и обратном кодах
- c. Длинная арифметика – реализовать алгоритм умножения длинных чисел.

Пример выполнения задачи не рекурсивного сложении длинных десятичных чисел без знака:

```
var s1,s2,r: string;
procedure sum1(c1,c2,ppred : byte; var r,p : byte);
begin
  r := 0;
  p := 0;

  r := (c1+c2+ppred) mod 100;
  p := (c1+c2+ppred) div 100;
end;

procedure sumN(s1,s2 : string; var r : string);
var i : integer;
    n : integer;
    ppred : byte;
    ri : byte;
    s1_byte,s2_byte : byte;
begin
  n := abs(length(s1) - length(s2));
  if length(s1) < length(s2)
  then for i := 1 to n do s1 := '0'+s1
  else
    if length(s1) > length(s2) then
      for i := 1 to n do s2 := '0'+s2;

  if length(s1) mod 2 <> 0 then begin s1 := '0'+s1; s2 := '0'+s2; end;

  n := length(s1);

  ppred := 0;
  i := n-1;
  while i > 0 do begin
    s1_byte := StrToInt(s1[i]+s1[i+1]);
    s2_byte := StrToInt(s2[i]+s2[i+1]);

    sum1(s1_byte,s2_byte,ppred, ri, ppred);

    r := inttostr(ri) + r;

    i := i - 2;
  end;

  if ppred <> 0 then r := inttostr(ppred) + r ;

  if r[1] = '0' then r := copy(r,2,length(r)-1);
end;

begin
  write('Введите первое слагаемое: '); readln(s1);
  write('Введите второе слагаемое: '); readln(s2);
  sumN(s1,s2,r);
  write('Ответ: ',r);

end.
```