

*Областное государственное бюджетное  
образовательное учреждение  
среднего профессионального образования  
«Иркутский авиационный техникум»*

УТВЕРЖДАЮ

Директор ОГБОУ СПО «ИАТ»

\_\_\_\_\_ В.Г. Семенов

**Комплект методических указаний по выполнению  
практических работ по дисциплине  
ОП.08 Дискретная математика**

образовательной программы (ОП)  
по специальности СПО

230113 Компьютерные системы и комплексы

базовой подготовки

**Иркутск 2013**

## Перечень практических (лабораторных) работ

№ работы	Название работы	Объём часов на выполнение работы	Страница
1	Множества и операции над ними	2	
2	Перестановки и сочетания с повторением	2	
3	Алгоритмическое перечисление (генерирование) некоторых комбинаторных объектов	4	
4	Составление программ шифрования	4	
5	Работа с вычетами на паскале	4	
6	Булевы функции. Способы их задания и основные свойства	4	
7	Представление графов в памяти ЭВМ	4	
8	Нахождение Эйлера цикла	4	
		28	

# Практическая работа №1

## “Множества и операции над ними”

### Цель работы:

На конкретных примерах разобраться с конечными множествами и операциями над ними; научиться находить число всех  $k$ -элементных подмножеств множества из  $n$  элементов и число перестановок множества, состоящего из  $n$  элементов. Закрепить полученные знания по данной тематике при решении занимательных задач. Научиться использовать в своей работе компьютер как инструмент.

### Содержание работы:

**Задача 1:**  $A$  – количество студентов;  $B$  – посещают математический кружок;  $C$  – физический;  $D$  – не посещают ни одного из кружков. Сколько студентов посещают и математический, и физический кружок? Сколько студентов посещают только математический кружок?  $|A|=37$ ;  $|B|=20$ ;

### Решение:

$$|C|=16; |D|=17.$$

$$|B \cup C| = |B| + |C| - |B \cap C|$$

$$\overline{|B \cup C|} = 17$$

$$\overline{|B \cup C|} = |A| - |B \cup C| = 37 - 17 = 20$$

$$20 = 20 + 16 - |B \cap C|$$

$$|B \cap C| = 16$$

$$|B \setminus C| = |B| - |B \cap C| = 20 - 16 = 4$$

**Ответ:** 16; 4

**Задача 2:** Сколькими способами из  $A$  студентов можно выбрать делегацию, если делегация состоит из  $B$  студентов?  $|A|=42$ ;  $|B|=8$ .

### Решение:

$$C_n^k = \frac{n!}{k!(n-k)!}$$

$$k=8; n=42$$

$$C_n^k = \frac{42!}{8!34!}$$

**Ответ:**  $\frac{42!}{8!34!}$

**Задача 3:** Сколько имеется четырехзначных чисел, у которых каждая следующая цифра меньше предыдущей?

### Решение:

$$C_n^k = \frac{n!}{k!(n-k)!}$$

$$n=10; k=4;$$

$$C_n^k = \frac{10!}{4!(10-4)!} = 210$$

**Ответ:** 210

**Задача 4:** На собрании должны выступить 4 человека А,В,С,Д. Сколькими способами их можно разместить в списке ораторов, если В не может выступить до того момента, пока не выступит А.

**Решение:**

$$\frac{4!}{2!}=12$$

CDAB DCAB CADB CABD DACB DABC ABCD ABDC ACBD ACDB ADBC ADCB

**Ответ:** 12

**Задача 5:** сколькими способами могут разместиться А покупателей в очереди в кассу. |А|=15

**Решение:**

$$P_n^k = \frac{n!}{(n-k)!}$$

$$n=15; k=15$$

$$P_n^k = \frac{15!}{0!} = 15!$$

**Ответ:** 15!

## Практическая работа №2 “Перестановки и сочетания с повторением”

### Цель работы:

На конкретных примерах научиться находить число перестановок и сочетаний с повторениями. Закрепить полученные знания по данной тематике при решении занимательных задач.

### Содержание работы:

**Задача 1:** Сколько различных слов можно составить, переставляя буквы слова КОМБИНАТОРИКА.

### Решение:

$$P_n\{k_1, k_2, \dots, k_m\} = \frac{n!}{k_1! \cdot k_2! \cdot \dots \cdot k_m!}$$

$$n=13 \{2, 2, 1, 1, 2, 1, 2, 1, 1\}$$

$$k_1=2, k_2=2, k_3=1, k_4=1, k_5=2, k_6=1, k_7=2, k_8=1, k_9=1$$

$$P_n = \frac{13!}{2!2!!1!1!2!!1!!1!} = 389188800$$

**Ответ:** 389188800

**Задача 2:** Сколькими способами можно выбрать А одинаковых или разных пирожных в кондитерской, где есть В разных сортов пирожных. А=5; В=12

### Решение:

$$U_n^k = n^k$$

$$n=12; k=5$$

$$U_n^k = 12^5 = 248832$$

**Ответ:** 248832

**Задача 3:** Пусть  $A = \{a, b, c, d, e\}$  и  $B = \{a, b, c, f, m, k\}$ . Указать все элементы множества  $A \times B$  – декартова произведения множеств А и В.

### Решение:

$$A \times B = \{(a, a), (a, b), (a, c), (a, f), (a, m), (a, k), \\ (b, a), (b, b), (b, c), (b, f), (b, m), (b, k), \\ (c, a), (c, b), (c, c), (c, f), (c, m), (c, k), \\ (d, a), (d, b), (d, c), (d, f), (d, m), (d, k), \\ (e, a), (e, b), (e, c), (e, f), (e, m), (e, k)\}$$

## Практическая работа №3

### “Алгоритмическое перечисление(генерирование) некоторых комбинаторных объектов”

**Цель работы:** приобретение навыков практического использования алгоритмов над объектами дискретной математики

**Оборудование:** ПК IBM PC

**Содержание работы:**

1. Составить блок-схему алгоритма по своему варианту
2. Составить тестовый пример по своему варианту
3. Ответить на контрольные вопросы
4. Оформить отчет и распечатать

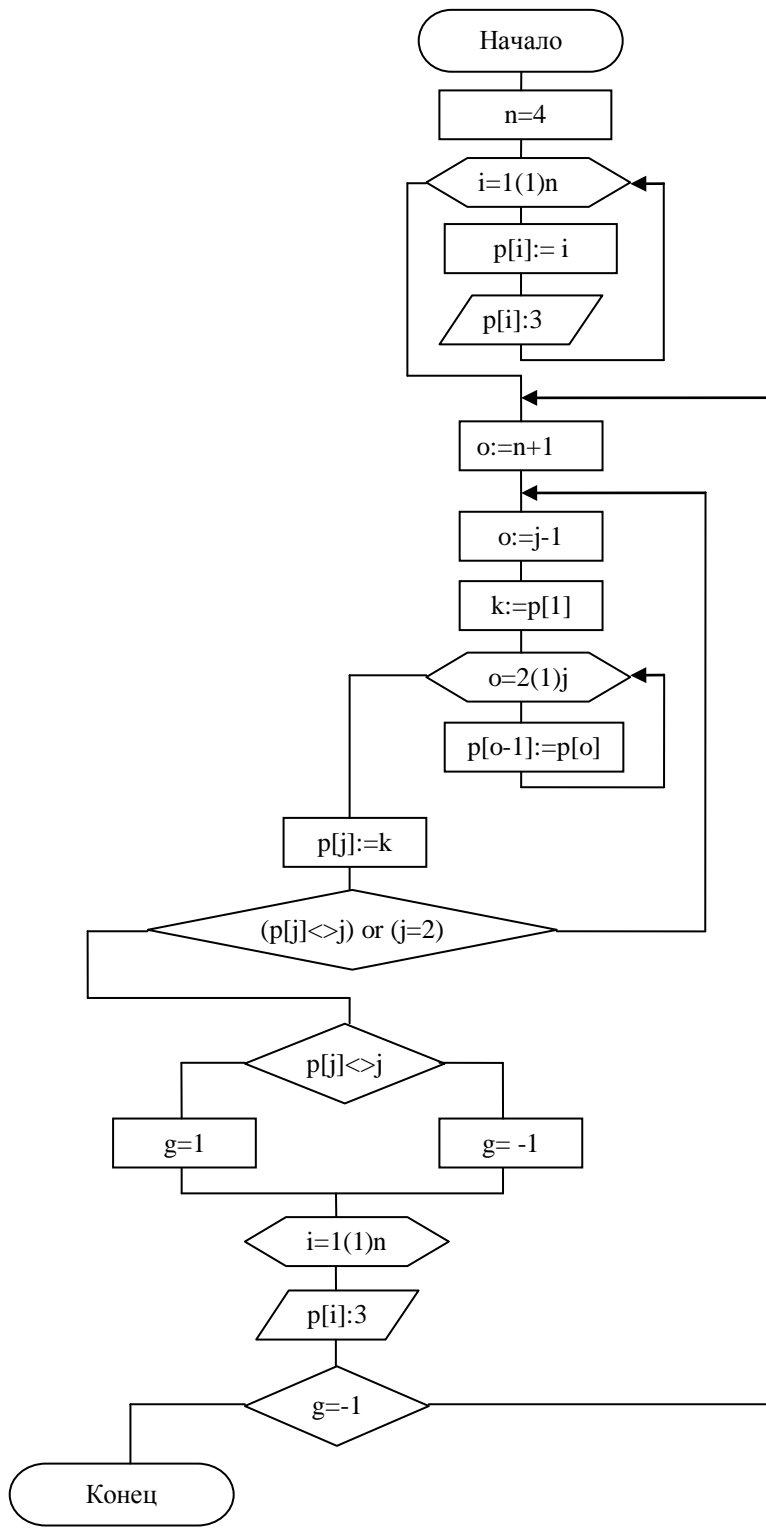
**Контрольные вопросы:**

1. Что такое "алгоритмическое перечисление"?
2. С какого вектора начинают работать оба генератора?
3. По какому признаку заканчивает работу генератор перестановок?
4. По какому признаку заканчивает работу генератор сочетаний?

```
program perest;
uses crt;
const n=4;
type mas=array [1..n] of integer;
var p:mas; i,g:integer;
procedure genper (n:word; var p:mas; var g:integer);
var o,j,k:word;
begin
o:=n+1;
repeat
o:=j-1;
k:=p[1];
for o:=2 to j do
p[o-1]:=p[o];
p[j]:=k;
until (p[j]<>j) or (j=2);
if (p[j]<>j) then g:=1 else g:=-1 end;

begin
clrscr;
for i:= 1 to n do
begin
p[i]:= i;
write (p[i]:3)
end;
writeln;
repeat genper(n,p,g);
for i:=1 to n do write (p[i]:3);
writeln;
until g=-1
end.
```

Разработал		Подпись	Дата	ПР 2203-ОПД.ОИ-03-02	Лист	Листов
			22.11.02		1	2
Проверил						



**Ответы:**

1. Алгоритмическое перечисление (выборка) – способ задания множества при помощи некоторого алгоритма.
2. Оба генератора начинают работать с вектора 123...n.
3. Генератор перестановок заканчивает работу, когда все элементы встают на свои первоначальные места.
4. Генератор сочетаний заканчивает работу, когда на первом месте стоит число, равное (n-k+1), где n – максимальное значение элемента, а k – количество элементов вектора.

Разработал		Подпись	Дата	ПР 2203-ОПД.ОИ-03-02	Лист	Листов
			22.11.02		2	2
Проверил						





## Практическая работа №4 “Составление программ шифрования”

**Цель работы:** приобретение умений кодирования и декодирования на примере шифров замены и перестановки

**Оборудование:** ПК IBM PC

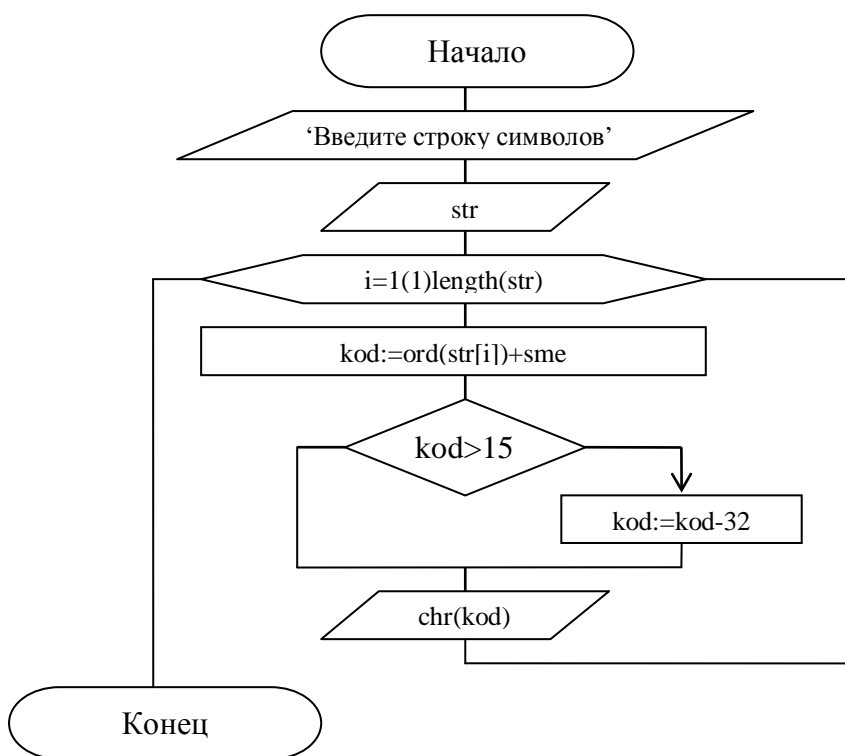
**Содержание работы:**

1. Составить блок-схему алгоритма по своему варианту
2. Составить тестовый пример по своему варианту
3. Составить программу по блок-схеме
4. Отладить программу и протестировать на контрольном примере
5. Ответить на контрольные вопросы
6. Оформить отчет и распечатать

**Контрольные вопросы:**

1. Перечислить задачи теории шифрования
2. В чем заключается проблема абсолютно стойкого шифра?
3. В чем разница между понятием “шифрование” и “кодирование”?
4. Какие методы шифрования вы знаете кроме шифра замены и шифра перестановки?

**Задание:** составить программу кодирования/декодирования сообщений с помощью шифра замены со сдвигом на 5 позиций.



```

program diskretn4; {Кодирование}
uses crt;
const sme=5;
var kod,i:integer;
str:string;
begin
  clrscr;
  writeln ('Vvedite stroku simvolov');
  readln(str);
  for i:=1 to length(str) do
  begin
    kod:=ord(str[i])+sme;
    if kod>159 then kod:=kod-32;
    write(chr(kod));
  end;
  readkey;
end.
  
```

**Пример:** denis → ijsnx

Разработал		Подпись	Дата	ПР 2203-ОПД.ОИ-04-02	Лист	Листов
			11.11.02		1	2
Проверил						

```

program diskretn4a;                                {Декодирование}
const sme=5;
var kod,i:integer; str:string;
begin
  writeln ('Vvedite stroku simvolov'); readln(str);
  for i:=1 to length(str) do
    begin
      kod:=ord(str[i])-sme;
      if kod>159 then kod:=kod-32;
      write(chr(kod));
    end;
  end.

```

**Ответы:**

1. Основной задачей теории шифрования в данное время является защита компьютерных данных от несанкционированного доступа, искажения и уничтожения.
2. Проблема создания абсолютно стойкого шифра заключается в том, что сейчас нет способа создать такой алгоритм, который при относительно небольших затратах обеспечивал бы надежную защиту данных от злоумышленника на довольно долгий срок. Причина этого в сверх быстром развитии вычислительных средств. Например, 22-го октября 1997 года за 212 дней был взломан методом полного перебора 56-битный ключ широко известной в криптографии компании RSA. Для этого участникам проекта (под названием **RC5-56**) пришлось перебрать порядка  $2^{56}$  (около  $7.2 \cdot 10^{16}$ ) ключей. Сейчас этот ключ был бы сломан за 2-3 дня. Так, завершившийся недавно проект по взлому 64-битного ключа (18,446,744,073,709,551,616 комбинаций: в 256 раз выше, чем 56-битного) занял 1,757 дней и объединил 331,252 человек с общей производительностью 45,998 процессоров класса AMD AthlonXP 2Ghz. Средняя скорость 103,883,000,840 комбинаций/сек. Поэтому простого увеличения длины ключа становится недостаточно, т.к. сейчас рассматривается не отдельно взятый компьютер, а целые сети, объединяющие в себе новейшие достижения вычислительной техники.
3. Кодирование – это процесс, переводящий одни символы в другие по определенным алгоритмам. Шифрование – это кодирование данных с целью защиты от несанкционированного доступа.
4. Методы шифрования: гаммирование (заключается в наложении на исходный текст некоторой псевдослучайной последовательности, генерируемой на основе ключа), системы с открытым ключом.

Разработал		Подпись	Дата	ПР 2203-ОПД.ОИ-04-02	Лист	Листов
			11.11.02		2	2
Проверил						



## Практическая работа №5 “Работа с вычетами на паскале”

**Цель работы:** приобретение умений использования вычетов в прикладных программах

**Оборудование:** ПК IBM PC

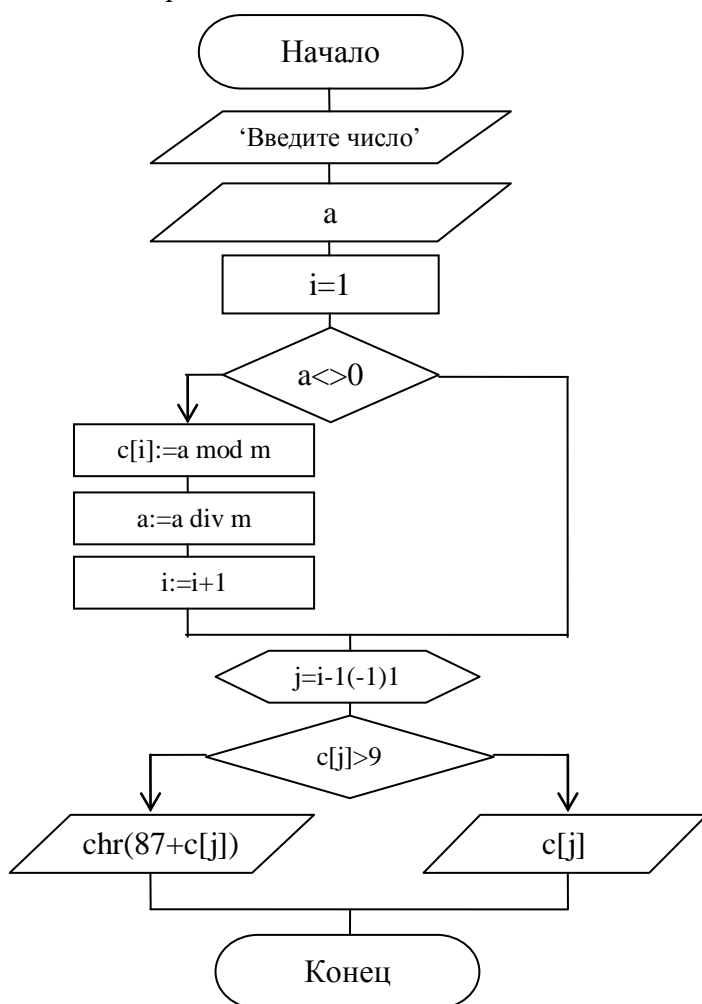
**Содержание работы:**

1. Составить блок-схему алгоритма по своему варианту
2. Составить тестовый пример по своему варианту
3. Составить программу по блок-схеме
4. Отладить программу и протестировать на контрольном примере
5. Ответить на контрольные вопросы
6. Оформить отчет и распечатать

**Контрольные вопросы:**

1. Определение классов вычетов
2. Различимы ли по  $\text{mod } n$  элементы внутри одного класса вычетов
3. Сколько классов вычетов образуется в результате операции сравнения по модулю  $n$
4. Назвать представителей каждого класса вычетов по модулю  $n$

**Задание:** перевести число из 10-ой системы счисления в 16-ю и вывести его на экран.



```
program vichet;
uses crt;
const m=16;
var a,i,j:integer;
    c:array [1..10] of integer;
begin
  clrscr;
  writeln('Vvedite chislo');
  readln(a);
  writeln;
  i:=1;
  while (a<>0) do
  begin
    c[i]:=a mod m;
    a:=a div m;
    i:=i+1;
  end;
  for j:=i-1 downto 1 do
  if c[j]>9 then write (chr(87+c[j]))
  else write (c[j]);
  readkey;
end.
```

**Тест:**  $1534_{10} \rightarrow 5fe_{16}$

**Ответы:**

1. Классами вычетов по модулю  $n$  называются подмножества, элементами которых являются все числа, дающие при делении на  $n$  один и тот же остаток.
2. Элементы внутри одного класса вычетов не различимы по  $\text{mod } n$ , т.к. при делении на  $n$  будут давать один и тот же остаток.
3. В результате операции сравнения по модулю  $n$  образуется  $n$  классов вычетов.
4. Представителями каждого класса вычетов по модулю  $n$  будут остатки:  $0, 1, 2, \dots, n-1$ .

## Практическая работа №6

### “Булевы функции. Способы их задания и основные свойства”

**Цель работы:** на конкретных примерах разобраться с работой элементарных булевых функций и освоить методики получения совершенной дизъюнктивной нормальной формы; методики представления функции в виде полинома Жегалкина.

**Содержание работы:**

- задана функция  $f$ ;
- найти значения функции  $f$  и свести полученные результаты в таблицу
- используя полученную таблицу, построить таблицу для двойственной функции  $f^*$
- построить СДНФ для функции  $f$
- выразить функцию  $f$  в виде полинома Жегалкина

$$f = [(x_1 \wedge x_2) \vee (x_3 \oplus (x_4 \rightarrow x_1))] / x_2$$

x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	$x_1 \wedge x_2$	$x_4 \rightarrow x_1$	$x_3 \oplus (x_4 \rightarrow x_1)$	$(x_1 \wedge x_2) \vee (x_3 \oplus (x_4 \rightarrow x_1))$	f	f*
0	0	0	0	0	1	1	1	1	1
0	0	0	1	0	0	0	0	1	1
0	0	1	0	0	1	0	0	1	1
0	0	1	1	0	0	1	1	1	1
0	1	0	0	0	1	1	1	0	0
0	1	0	1	0	0	0	0	1	0
0	1	1	0	0	1	0	0	1	0
0	1	1	1	0	0	1	1	0	0
1	0	0	0	0	1	1	1	1	1
1	0	0	1	0	1	1	1	1	0
1	0	1	0	0	1	0	0	1	0
1	0	1	1	0	1	0	0	1	1
1	1	0	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1	0	0
1	1	1	0	1	1	0	1	0	0
1	1	1	1	1	1	0	1	0	0

СДНФ:

$$f = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 x_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 x_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_3 x_4$$

ДНФ:

$$\bar{x}_2 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 x_3 \bar{x}_4$$

Полином Жегалкина:

$$x_2 \oplus 1 \oplus x_2 x_4 \oplus x_1 x_2 x_4 \oplus x_1 x_2 x_3 \oplus x_2 x_3$$

1. Функция  $f(x_1, x_2, \dots, x_n)$  называется булевой функцией, если она имеет тип:  $f(x_1, x_2, \dots, x_n): \{0;1\}^n \rightarrow \{0;1\}$  ( $x_i \in \{0;1\} \forall i$ )
2. Функция  $f^*(x_1, x_2, \dots, x_n)$  называется двойственной к  $f(x_1, x_2, \dots, x_n)$ , если она принимает противоположные значения на противоположных значениях векторов переменных.
3. СДНФ функции  $n$  переменных представляет собой дизъюнкцию, слагаемые которой – элементарные конъюнкции  $n$ -го ранга.

4. Важнейшие замкнутые классы:  $T_0$ (класс функций, сохраняющих 0),  $T_1$ (класс функций, сохраняющих 1),  $S$ (класс самодвойственных функций),  $L$ (класс линейных функций),  $M$ (класс монотонных функций).
5. Основные элементарные функции: конъюнкция, дизъюнкция, отрицание, импликация, эквиваленция.
6. Свойства элементарных функций:
  - закон коммутативности  
 $A \cup B = B \cup A \quad A \cap B = B \cap A$
  - закон ассоциативности  
 $A \cup (B \cup C) = (A \cup B) \cup C \quad A \cap (B \cap C) = (A \cap B) \cap C$
  - дистрибутивность  
 $A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \quad A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
  - закон де Моргана  
 $\overline{A \cap B} = \overline{A} \cup \overline{B} \quad \overline{A \cup B} = \overline{A} \cap \overline{B}$
  - правило поглощения  
 $(A \cap B) \cup A = A \quad (A \cup B) \cap A = A$
  - закон идемпотентности  
 $A \cup A = A \quad A \cap A = A$
  - двойного отрицания  
 $\overline{\overline{A}} = A$

**Практическая работа №8**  
**“Представление графов в памяти ЭВМ ”**  
 Вариант №12

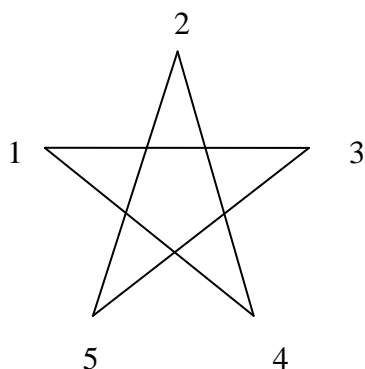
**Цель работы:** получить первичные навыки работы с графами программно: научиться представлять графы в памяти ЭВМ на языке программирования Паскаль.

**Содержание работы:**

- построить граф указанным способом;
- составить алгоритм, который будет представлять граф другим способом;
- запрограммировать свой вариант;
- вывести граф, представленный тем и другим способом на печать.

**Оборудование:** ПК, среда Turbo Pascal

**Задание:** матрицу инцидентности вывести в виде перечня ребер



Матрица инцидентности:

	1	2	3	4	5
I	1	0	1	0	0
II	0	0	1	0	1
III	0	1	0	0	1
IV	0	1	0	1	0
V	1	0	0	1	0

Список ребер: (1,3) (3,5) (2,5) (2,4) (1,4)

1. Орграф – граф, содержащий направленные ребра
2. Мультиграф – граф, содержащий кратные ребра
3. Ребра, инцидентные одной и той же паре вершин называются кратными
4. Степень вершин равна числу ребер, инцидентных ей
5. Смежными называются вершины, которым инцидентно одно и то же ребро, соединяющее их
6. Два ребра называются смежными если они имеют общую инцидентную вершину
7. Отношение инцидентности – отношение, при котором каждое ребро графа инцидентно ровно двум вершинам графа.

Разработал		Подпись	Дата	<b>ПР 2203-ОПД.04-08-03</b>	Лист	Листов
			04.05.03		1	2
Проверил						

```

uses crt;
const n=5; m=5;
var a:array [1..n,1..m] of integer;  {matr. incid.}
    b:array [1..2,1..n] of integer;  {spisok reber}
    i,j,k,l:integer;
begin
clrscr;
writeln('Vvedite matr. incidentnosti');
for i:=1 to n do
for j:=1 to m do
begin
write('a['i',';',j,']=');
read(a[i,j]);
end;
writeln;
k:=0;
l:=0;
for i:=1 to n do
for j:=1 to m do
if a[i,j]<>0 then
case 1 of
0: begin
inc(k);
b[1,k]:=j;
inc(l);
end;
1: begin
b[2,k]:=j;
l:=0;
end;
end;
for i:=1 to n do  {vyvod matr. incid.}
begin
for j:=1 to m do
write (a[i,j], ' ');
writeln;
end;
for i:=1 to k do  {vyvod spiska reber}
writeln (('b[1,i],',b[2,i]'));
readkey;
end.

```

Разработал		Подпись	Дата	ПР 2203-ОПД.04-08-03	Лист	Листов
			04.05.03		2	2
Проверил						





## Практическая работа №9 "Нахождение Эйлера цикла"

**Цель:** разобраться со структурой данных типа дерево. Научиться составлять программы на языке Паскаль с использованием структурных данных типа дерево, закрепить навыки по составлению алгоритма с графами.

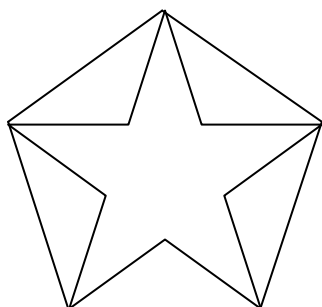
**Задание:** составить алгоритм и программу, которая определяет является ли заданный граф Эйлеровым, вывести граф на печать, проверить результат с заранее составленным тестовым примером.

### Ответы на контрольные вопросы:

1. Графы имеющие Эйлера циклы (простые циклы, содержащие все ребра графа) называются Эйлера графами.
2. Гамильтоновым циклом называется простой цикл, проходящий через все вершины графа.
3. Граф называется циклически связным, если через любые две вершины рассматриваемого графа может проходить простой цикл.
4. Граф называется связным, если у него нет изолированных вершин.
5. Маршрут называется цепью, если каждое ребро встречается в нем не более раза.
6. Маршрутом называется такая последовательность ребер  $(e_0, e_1, \dots, e_{n-1}, e_n)$ , что каждые 2 соседних ребра смежные, т.е.  $\forall i e_{i-1}$  и  $e_i$  имеют общую инцидентную вершину.
7. Простым циклом называют циклический маршрут, если он является простой цепью.

1	2	5	4	1	6	1	4	7	2	8	9	9	10	10
2	3	6	3	5	1	7	5	2	8	3	3	4	4	5

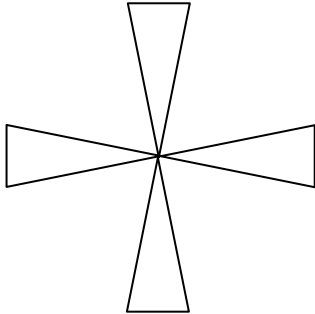
1	7	2	8	3	9	4	10	5	1	6	5	4	3	2	1
---	---	---	---	---	---	---	----	---	---	---	---	---	---	---	---



Разработал		Подпись	Дата	<b>ПР 2203-ОПД.04-09-03</b>	Лист	Листов
			12.05.03		1	3
Проверил						

9	9	9	9	9	9	9	9	1	3	5	7
1	2	3	4	5	6	7	8	2	4	6	8

9	2	1	9	4	3	9	6	5	9	8	7	9
---	---	---	---	---	---	---	---	---	---	---	---	---



```

uses crt;
var v_n,v_t:byte;
    mass1,mass2:array [1..100,1..2] of byte;
    stek1,stek2:array [1..100] of byte;
    num1,num2,i,j,k,m:integer;
begin
  clrscr;
  write ('Vvedite kol-vo reber: ');
  read (k);
  {-----vvod reber-----}
  for i:=1 to k do
    begin
      write ('Vvedite ',i,' paru vershin: ');
      readln (mass1[i,1],mass1[i,2]);
      mass2[i,1]:=mass1[i,1];
      mass2[i,2]:=mass1[i,2];
    end;
  {-----main-----}
  v_n:=mass1[1,1];
  v_t:=mass1[1,2];
  stek1[1]:=v_n;
  stek1[2]:=v_t;
  num1:=3;
  num2:=1;
  mass1[1,1]:=0;
  mass1[1,2]:=0;

```

Разработал		Подпись	Дата	ПР 2203-ОПД.04-09-03	Лист	Листов
			12.05.03		2	3
Проверил						

```

repeat
  m:=0;
  for i:=2 to k do
    begin
      if mass1[i,1]=v_t then
        begin
          if mass1[i,2]=v_n then
            begin
              stek2[num2]:=v_n;
              inc(num2);
              v_n:=v_t;
            end else
            begin
              stek1[num1]:=mass1[i,2];
              inc(num1);
              v_t:=mass1[i,2];
            end;
          mass1[i,1]:=0;
          mass1[i,2]:=0;
          i:=k;
          m:=1;
        end;
      if mass1[i,2]=v_t then
        begin
          if mass1[i,1]=v_n then
            begin
              stek2[num2]:=v_n;
              inc(num2);
              v_n:=v_t;
            end else
            begin
              stek1[num1]:=mass1[i,1];
              inc(num1);
              v_t:=mass1[i,1];
            end;
          mass1[i,1]:=0;
          mass1[i,2]:=0;
          i:=k;
          m:=1;
        end;
      end;
    if m=0 then
      begin
        dec(num1);
        stek2[num2]:=stek1[num1];
        stek1[num1]:=0;
        v_n:=stek1[num1-1];
        v_t:=stek1[num1-1];
        inc(num2);
      end;
    until num2>k+1;
  {-----vyvod spiska reber-----}
  clrscr;
  for i:=1 to k do write (mass2[i,1],' ');
  writeln;
  for i:=1 to k do write (mass2[i,2],' ');
  writeln;
  writeln;
  {-----vyvod vershin grafa-----}
  for i:=1 to k+1 do write (stek2[i],' ');
  readkey;
end.

```

Разработал	.	Подпись	Дата	ПР 2203-ОПД.04-09-03	Лист	Листов
			12.05.03		3	3
Проверил						

