

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Лабораторный практикум по дисциплине
ПРОЕКТИРОВАНИЕ СИСТЕМ ОБРАБОТКИ ДАННЫХ»
ТЕСТИРОВАНИЕ СТРУКТУРЫ МОДУЛЯ

для студентов специальности
«Автоматизированные системы обработки информации»

Данный лабораторный практикум предназначен для использования студентами в процессе выполнения лабораторной работы по дисциплине «Проектирование систем обработки информации» по теме «Тестирование структуры программных модулей». Лабораторный практикум включает постановку задач на выполнение лабораторной работы, основные понятия и методику для разработки тестов для проверки структуры программных модулей. Приведен пример разработки тестов. В приложениях лабораторного практикума приведены варианты исходных данных и схем алгоритмов для индивидуального выполнения лабораторной работы студентами.

Табл. 6., список лит. 2 назв., 56 рис.

Составители: В.И. Хвещук, профессор, к.т.н.
Г.Л.Муравьев, профессор, к.т.н.

Рецензенты: доцент, к.п.н. Брестского государственного университета им. А.С. Пушкина

СОДЕРЖАНИЕ

1. ПОСТАНОВКА ЗАДАЧ	4
2. ТЕСТИРОВАНИЕ СТРУКТУРЫ ПРОГРАММНЫХ МОДУЛЕЙ.....	5
2.1. ОСНОВНЫЕ ПОНЯТИЯ.....	5
2.2. МЕТОДИКА РАЗРАБОТКИ ТЕСТОВ ДЛЯ ПРОВЕРКИ СТРУКТУРЫ МОДУЛЯ	6
2.3. ПРИМЕР РАЗРАБОТКИ ТЕСТОВ ДЛЯ ПРОВЕРКИ СТРУКТУРЫ МОДУЛЯ.....	7
ПРИЛОЖЕНИЕ П1. ТАБЛИЦЫ ВАРИАНТОВ ИСХОДНЫХ ДАННЫХ	10
ПРИЛОЖЕНИЕ П2. ВАРИАНТЫ СХЕМ АЛГОРИТМОВ МОДУЛЕЙ.....	14

1. ПОСТАНОВКА ЗАДАЧ

ЦЕЛЬ:

Сформировать знания и практические умения, необходимые для разработки тестов для проверки структуры отдельных программных модулей системы обработки данных (СОД).

ЗАДАЧИ:

1. Изучить постановки задач и исходные данные на лабораторную работу;
2. Разработать набор маршрутов для проверки структуры модулей на основе критерия минимального покрытия графа программы;
3. Разработать набор маршрутов для проверки структуры модулей на основе критерия максимального покрытия графа программы;
4. Разработать тесты для созданных маршрутов проверки;
5. Проанализировать маршруты проверки и определить нереализуемые маршруты;
6. Выбрать критерий и упорядочить наборы разработанных тестов;
7. Оформить и защитить работу у преподавателя.

ИСХОДНЫЕ ДАННЫЕ:

Индивидуальный вариант значений входных параметров для данной лабораторной работы, который выдает преподаватель, включает следующие параметры:

1. Вариант схемы алгоритма модуля для разработки тестов;
2. Исходные данные для схемы алгоритма модуля (см. прил.1);
3. Критерии для разработки наборов тестов (минимального покрытия, максимального покрытия);
4. Критерии для упорядочения тестов (по времени выполнения, по количеству логических операторов).

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:

1. Изучить теоретическую часть лабораторной работы (см.п.2).
2. Ответить на контрольные вопросы.
3. Решить задачи, перечисленные выше.
4. Оформить результаты выполнения лабораторной работы.
5. Защитить у преподавателя результаты выполнения лабораторной работы.

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ РАБОТЫ:

Отчет по лабораторной работе должен содержать следующие результаты:

1. Постановка задач на лабораторную работу.
2. Индивидуальный вариант значений входных параметров и схема алгоритма модуля для разработки тестов;
3. Описание тестов (пример см. табл.2.1);
4. Результаты анализа и упорядочения тестов (см. табл.2.2).

ПРИМЕЧАНИЕ:

В качестве основы для разработки тестов для проверки структуры модуля используется схема алгоритма модуля, которая представляется в виде совокупности блоков (начальный, конечный, логический, вычислительный) и линий, которые представляют передачи управления. Например, см. рис.2.1.

2. ТЕСТИРОВАНИЕ СТРУКТУРЫ ПРОГРАММНЫХ МОДУЛЕЙ

2.1. ОСНОВНЫЕ ПОНЯТИЯ

Тестирование программы отдельного модуля предполагает выполнение следующей последовательности проверок [1,2]:

1. **Проверка интерфейса модуля.** Интерфейс модуля тестируется для проверки правильности ввода-вывода тестовой информации. Если нет уверенности в правильном вводе-выводе данных, нет смысла проводить другие тесты.
2. **Проверка внутренних структур данных.** Проверка внутренних структур данных гарантирует целостность сохраняемых данных.
3. **Проверка структуры программного модуля** предназначена для выявления ошибок в схеме принятия решений и логике функционирования модуля. Должна быть проверена корректность структуры модуля и их основные конструктивные компоненты: циклы, логические операторы и т.д. Проверке подлежат маршруты обработки информации в модуле и правильность их реализации в зависимости от исходных данных. Полнота проверки определяется критериями выделения маршрутов для тестирования и степенью покрытия маршрутов исполнения программы.
4. **Проверка вычислений и преобразований данных программным модулем** служит для обнаружения ошибок в вычислительной части программы. Для этого выделяются компоненты программного модуля и маршруты обработки данных, содержащие вычисления и преобразования переменных. В состав теста входят наборы значений исходных переменных во всей области их определения. Особо выделяются комбинации переменных в критических и особых точках, а также сочетания противоречивых и искаженных данных. Эталоном служат результаты предварительных расчетов по формулам, заложенным в модуле.
5. **Проверка полноты функций,** выполняемых модулем, предназначена для выявления ошибок в разработанной программе относительно функций, заданных в программной спецификации. Тестированию подлежат реализация всех функций, определенных спецификацией, и характеристики интерфейса. Для проверки функций используются результаты тестирования структуры и вычислений, которые дополняются тестами для проверки неконтролировавшихся функций.

В рамках данной лабораторной работы рассматривается только тестирование структуры отдельного модуля. Необходимые понятия и определения следующие.

Оценку качества структуры программы можно проводить на основе показателя **корректности структуры программы** [2]. Для оценки этого показателя наиболее часто используются несколько частных показателей – критериев выделения маршрутов программы для тестирования – критерии K1, K2 или K3 [2], которые различаются степенью охвата тестами структуры программы и необходимыми ресурсами для их реализации.

Критерий выделения маршрутов программы для тестирования – это критерий, который определяет полноту покрытия графа программы маршрутами проверки. Например [2]:

- **Критерий K1** – покрытие графа программы минимальным количеством маршрутов, охватывающих каждую дугу хотя бы один раз (минимальное покрытие). Согласно критерию K1 должны быть разработаны тесты, которые обеспечивают покрытие всех блоков решений в программе. При этом каждое решение должно принять значение «истина» и «ложь», а также должна исполниться каждая логическая ветвь программы.
- **Критерий K3** – покрытие графа программы маршрутами при всех возможных комбинациях дуг (максимальное или полное покрытие). Согласно критерию K2 должны быть разработаны тесты, обеспечивающие выполнение всех возможных комбинаций результатов условий в каждом решении. Этот критерий является наиболее полным.

Критерии приведены в порядке углубления тестовых проверок увеличения объема тестов и достигаемой степени корректности программ.

Проверку корректности структуры программы будем проводить статически (ручным способом) на основе схемы алгоритма программы модуля, которая задается в качестве исходных данных для лабораторной работы.

Граф программы. Для удобства рассмотрения схем алгоритмов программ используется их представление в виде графа программы. Для этого необходимо заменить блоки (операторы) схемы на вершины графа.

Маршрут проверки – это совокупность конкретных вершин и дуг на графе программы, которые выполняются в рамках данного маршрута. Некоторые маршруты могут оказаться нереализуемыми из-за несовместимости условий, которые последовательно анализируются в разных вершинах маршрута. Эти маршруты отбраковываются.

Тест – это отдельная проверка программы модуля, которая определяется следующими компонентами: маршрутом проверки; совокупностью значений входных данных модуля (**входной набор теста**), определяющих выполнение программы по данному маршруту; совокупностью значений выходных данных модуля (**эталон** для проверки – **выходной набор теста**), рассчитанных для данного маршрута на основе входного набора.

Упорядочение тестов. Для упорядочения разработанных тестов для проверки модуля можно использовать следующие критерии: по длительности исполнения или по числу команд в маршруте; по количеству условных операторов в маршруте; по вероятности исполнения маршрута и другие. По первому критерию первичной проверке подвергаются маршруты, наиболее длинные либо по числу команд, либо по времени исполнения. По второму критерию приоритет отдается маршрутам наиболее сложным по числу условий. По третьему критерию в первую очередь выполняются маршруты с наибольшей вероятностью их реализации.

2.2. МЕТОДИКА РАЗРАБОТКИ ТЕСТОВ ДЛЯ ПРОВЕРКИ СТРУКТУРЫ МОДУЛЯ

Методика разработки тестов для проверки структуры модуля включает последовательность следующих этапов:

1. **Выбор критерия выделения маршрутов для проверки.** Эти критерии определяют полноту проверки структуры модуля. Тестирование структуры программы предполагает разработку групп тестов в соответствии с выбранным критерием полноты проверки структуры программы, который позволяет в той или иной степени проверить прохождение различных маршрутов программы.
2. **Разработка набора маршрутов.** В соответствии с выбранным критерием разрабатывается соответствующее множество маршрутов для проверки структуры программы. Каждый маршрут определяется в виде последовательности проверяемых вершин и дуг графа программы.
3. **Разработка тестов.** Для каждого маршрута разрабатывается свой тест: подбирается конкретный набор значений входных данных для данного маршрута; рассчитывается соответствующий набор выходных данных (эталон), которые соответствуют заданному маршруту.
4. **Анализ маршрутов и отбраковка нереализуемых маршрутов.** Каждый маршрут анализируется на предмет его реализуемости. Для этой цели рассматриваются логические выражения в логических операторах анализируемого маршрута. В случае,

если обнаруживаются несовместимые логические выражения в разных вершинах маршрута, например, $X > 2$ и $X < 1$ (X не может одновременно быть больше двух и меньше единицы), то такой маршрут является нереализуемым и выбрасывается из дальнейшего рассмотрения.

- 5. Выбор критерии для упорядочения тестов.** Критерий (критерии) выбирается на основе анализа результатов разработки тестов и наличия необходимой информации для их применения. При отсутствии упорядоченного анализа структуры модуля некоторое число маршрутов в модуле может оказаться пропущенным при проверке.
- 6. Упорядочение тестов.** Оно выполняется для тестов в соответствии с выбранным критерием упорядочения маршрутов.

2.3. ПРИМЕР РАЗРАБОТКИ ТЕСТОВ ДЛЯ ПРОВЕРКИ СТРУКТУРЫ МОДУЛЯ

Процесс разработки набора тестов для проверки структуры программы отдельного модуля рассмотрим для модуля, схема алгоритма которого представлена на рис.2.1. Обозначения блоков в результирующих таблицах (табл.2.1. и табл.2.2) приведены с использованием их представления на схеме алгоритма.

Этап 1. Выбор критериев выделения маршрутов для покрытия графа программы. Процесс разработки маршрутов рассмотрим для двух вариантов критериев – минимального и максимального покрытия графа программы, которые определены в п.2.1.

Этап 2. Разработка маршрутов. В соответствии с перечисленными выше критериями разработано два набора маршрутов проверки, которые представлены в табл.2.1. Первая группа маршрутов (критерий K1) включает 4 маршрута. Эти маршруты минимальным образом покрывают граф программы. Для этого критерия повторная проверка дуг не оценивается и считается избыточной.

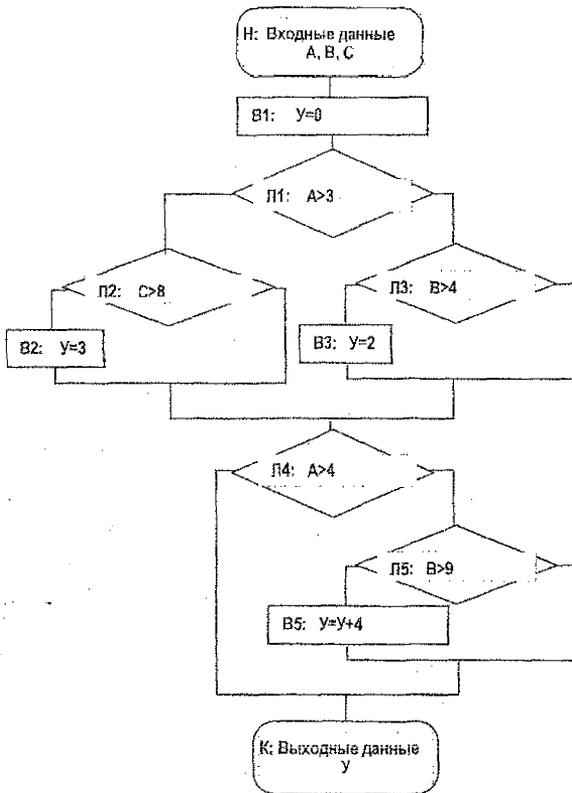
Вторая группа представлена 12 маршрутами (критерий K2). В этой группе представлены все возможные комбинации дуг в маршрутах.

Для каждого маршрута из рассмотренных групп в таблице описания тестов приводится:

1. Определение элементов маршрута проверки в виде последовательности блоков (операторов) схемы алгоритма, которые вошли в маршрут. Рекомендуется его представлять в виде наименований блоков из схемы алгоритма, через которые он проходит. Например, В1-Л1-Л3-Л4-Л5-К и т.д. При этом использованы следующие обозначения:
 - ВN – символ В обозначает вычислительный блок, а N – порядковый номер этого блока;
 - ЛN – символ Л обозначает логический блок, а N – порядковый номер этого блока;
 - Н и К – соответственно начальный и конечный блоки схемы алгоритма.
2. Документирование логических условий по маршруту, которые используются в логических блоках. Эта информация используется для определения нереализуемых маршрутов.

Этап 3. Разработка тестов. Для каждого маршрута разрабатывается отдельный тест, который включает определение следующей информации:

1. Присвоение тесту идентификатора (каждому тесту присваивается порядковый номер).
2. Подбор набора значений для входных данных модуля, которые определяют реализацию данного маршрута;
3. Расчет значений выходных данных модуля (эталон) для данного набора входных данных.



Примечание:

В качестве исходных данных модуля использованы переменные А, В, С. В качестве выходных данных определена переменная У. На рис.2.2 использованы следующие обозначения:

- В1, ..., В5 – обозначения блоков (операторов), в которых выполняются расчеты. Для каждого отдельного блока представлены расчетные формулы для выполнения вычислений;
- Л1, ..., Л5 – обозначения логических блоков (операторов), в которых выполняются логические операции. Логическое выражение задается в виде логического выражения над входными данными

Рисунок 2.1 – Пример схемы алгоритма программы модуля

Этап 4. Анализ и отбраковка нереализуемых маршрутов. В первой группе тестов (критерий К1) нереализуемых маршрутов не выявлено. Для второго набора тестов было выявлено четыре маршрута (см.табл.2.1), 3, 4, 7 и 8 для которых выявлены – несовместимые логические условия $A \leq 3$ и $A > 4$. Перечисленные маршруты отбракованы, т.е. выброшены из дальнейшего рассмотрения.

Этап 5. Выбор критерия для упорядочения тестов. Для упорядочения маршрутов использованы следующие критерии: упорядочение по количеству логических операторов и по длительности выполнения маршрута. Значения этих критериев (количество логических и вычислительных операторов для каждого маршрута) приведены в табл.2.2.

Этап 6. Упорядочение тестов. Результаты упорядочения разработанного набора тестов приведены в табл.2.2. Как представлено в табл.2.2, количество логических блоков для всех вариантов тестов равно четырем. Поэтому упорядочение тестов по этому параметру не имеет смысла. Упорядочение выполнено по второму критерию (количество вычислительных блоков) и представлено в графе «Последовательность выполнения тестов» (см. табл.2.2).

Результат разработки тестов для двух групп маршрутов представлен в табл.2.1.

Таблица 2.1 – Описание тестов для проверки структуры модуля

Критерий покрытия графа программы	Входные данные (входной набор теста)			Выходные данные (эталон)	Проверяемые условия на маршруте	Определение маршрута проверки	Номер теста
	A	B	C				
K1	5	10		0	$A>3, B>4, A>4, B>9$	Н-В1-Л1-Л3-Л4-Л5-К	1
	5	4		6	$A>3, B=<4, A>4, B=<9$	Н-В1-Л1-Л3-В3-Л4-Л5-В5-К	2
	3		9	0	$A=<3, C>8, A=<4$	Н-В1-Л1-Л2-Л4-К	3
	3		9	3	$A=<3, C=<8, A=<4$	Н-В1-Л1-Л2-В2-Л4-К	4
K2	5	5		0	$A>3, B>4, A>4, B>9$	Н-В1-Л1-Л3-Л4-Л5-К	1
	5	4		2	$A>3, B=<4, A>4, B>9$	Н-В1-Л1-Л3-В3-Л4-Л5-К	2
				0	$A=<3, C>8, A>4, B>9$	Н-В1-Л1-Л2-Л4-Л5-К	3
				0	$A=<3, C=<8, A>4, B>9$	Н-В1-Л1-Л2-В2-Л4-Л5-К	4
	2	5		4	$A>3, B>4, A>4, B=<9$	Н-В1-Л1-Л3-Л4-Л5-В5-К	5
	5	4		0	$A>3, B=<4, A>4, B=<9$	Н-В1-Л1-Л3-В3-Л4-Л5-В5-К	6
				4	$A=<3, C>8, A>4, B=<9$	Н-В1-Л1-Л2-Л4-Л5-В5-К	7
				6	$A=<3, C=<8, A>4, B=<9$	Н-В1-Л1-Л2-В2-Л4-Л5-В5-К	8
	4	5		0	$A>3, B>4, A=<4$	Н-В1-Л1-Л3-Л4-Л5-В5-К	9
	4	4	9	2	$A>3, B=<4, A=<4$	Н-В1-Л1-Л3-В3-Л4-Л5-В5-К	10
	2		9	0	$A=<3, C>8, A=<4$	Н-В1-Л1-Л3-Л4-Л5-В5-К	11
	3		8	3	$A=<3, C=<8, A=<4$	Н-В1-Л1-Л3-В3-Л4-Л5-В5-К	12

Таблица 2.2 – Результаты анализа и упорядочения тестов

Номер теста из табл.2.1	Маршрут проверки	Количество логических блоков в маршруте	Количество вычислительных блоков в маршруте	Последовательность выполнения теста
1	Н-В1-Л1-Л3-Л4-Л5-К	4	1	8
2	Н-В1-Л1-Л3-В3-Л4-Л5-К	4	2	4
5	Н-В1-Л1-Л3-Л4-Л5-В5-К	4	2	5
6	Н-В1-Л1-Л3-В3-Л4-Л5-В5-К	4	3	1
9	Н-В1-Л1-Л3-Л4-Л5-В5-К	4	2	6
10	Н-В1-Л1-Л3-В3-Л4-Л5-В5-К	4	3	2
11	Н-В1-Л1-Л3-Л4-Л5-В5-К	4	2	7
12	Н-В1-Л1-Л3-В3-Л4-Л5-В5-К	4	3	3

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- Орлов, С.А. Технология разработки программного обеспечения: учебник для вузов. / С.А. Орлов – СПб.: Питер, 2004. – 527 с.
- Липаев, В.В. Тестирование программ. / В.В. Липаев. – М.: Статистика, 1995. – 254 с.

Таблица П.1.2 – Таблица исходных данных для заполнения схем алгоритмов модулей
(для вариантов В1-В14)

Вариант	Входные данные	ЛОГИЧЕСКИЕ ОПЕРАТОРЫ (обозначения блоков и логических выражений)							ОПЕРАТОРЫ ВЫЧИСЛЕНИЯ (обозначения блоков и формул для расчета)							Выходные данные (этапы)																												
		П1	П2	П3	П4	П5	П6	П7	В1	В2	В3	В4	В5	В6	В7																													
В1	A	A>1	A=2	C<3	D<4	B<5	A<3	B<6	A<3	B<4	D<5	C<6	D<7	A<8	B<9	A<10	A>11	D<12	B<13	C<2	D<3	C<4	D<4	B<5	B<6	C<5	D<6	B<7	A<8	B<9	A<10	A<11	A<12	B<13	C<14									
В2	A	B=4	C<3	C<4	D<5	B<6	A<3	B<4	D<5	C<6	D<7	A<8	B<9	A<10	A<11	A<12	B<13	C<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	B<22	C<23	D<24	A<25	B<26	C<27	D<28	A<29	B<30	C<31	D<32								
В3	A	C<5	D<6	B<7	A<8	B<9	C<10	D<11	A<12	B<13	C<14	D<15	A<16	B<17	C<18	D<19	A<20	B<21	C<22	D<23	A<24	B<25	C<26	D<27	A<28	B<29	C<30	D<31	A<32	B<33	C<34	D<35	A<36	B<37	C<38	D<39	A<40	B<41	C<42	D<43				
В4	A	D<4	B<5	A<6	C<7	D<8	B<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45	
В5	A	B<3	C<4	D<5	A<6	B<7	C<8	D<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45
В6	A	C<3	D<4	B<5	A<6	B<7	C<8	D<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45
В7	A	D<4	B<5	A<6	C<7	D<8	B<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45	
В8	A	B<3	C<4	D<5	A<6	B<7	C<8	D<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45
В9	A	C<3	D<4	B<5	A<6	B<7	C<8	D<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45
В10	A	D<4	B<5	A<6	C<7	D<8	B<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45	
В11	A	B<3	C<4	D<5	A<6	B<7	C<8	D<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45
В12	A	C<3	D<4	B<5	A<6	B<7	C<8	D<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45
В13	A	D<4	B<5	A<6	C<7	D<8	B<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45	
В14	A	B<3	C<4	D<5	A<6	B<7	C<8	D<9	A<10	B<11	C<12	D<13	A<14	B<15	C<16	D<17	A<18	B<19	C<20	D<21	A<22	B<23	C<24	D<25	A<26	B<27	C<28	D<29	A<30	B<31	C<32	D<33	A<34	B<35	C<36	D<37	A<38	B<39	C<40	D<41	A<42	B<43	C<44	D<45

ПРИЛОЖЕНИЕ П2. ВАРИАНТЫ СХЕМ АЛГОРИТМОВ МОДУЛЕЙ

В данном приложении приведены схемы алгоритмов модулей для выполнения лабораторной работы на тему «Тестирование структуры программы модуля». Каждая отдельная схема идентифицируется уникальным номером (вариантом). Варианты схем пронумерованы следующим образом: А1, А2, ..., А13 (13 вариантов); В1, В2, ..., В13 (13 вариантов); С1, С2, ..., С13 (13 вариантов); D1, D2, ..., D13 (13 вариантов).

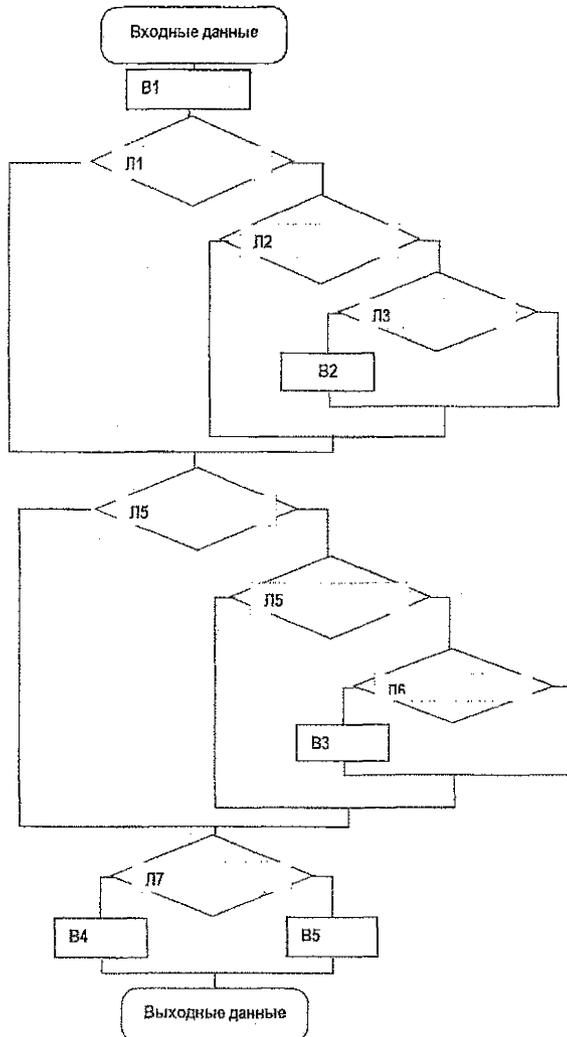


Рисунок 2.1 – Вариант А1

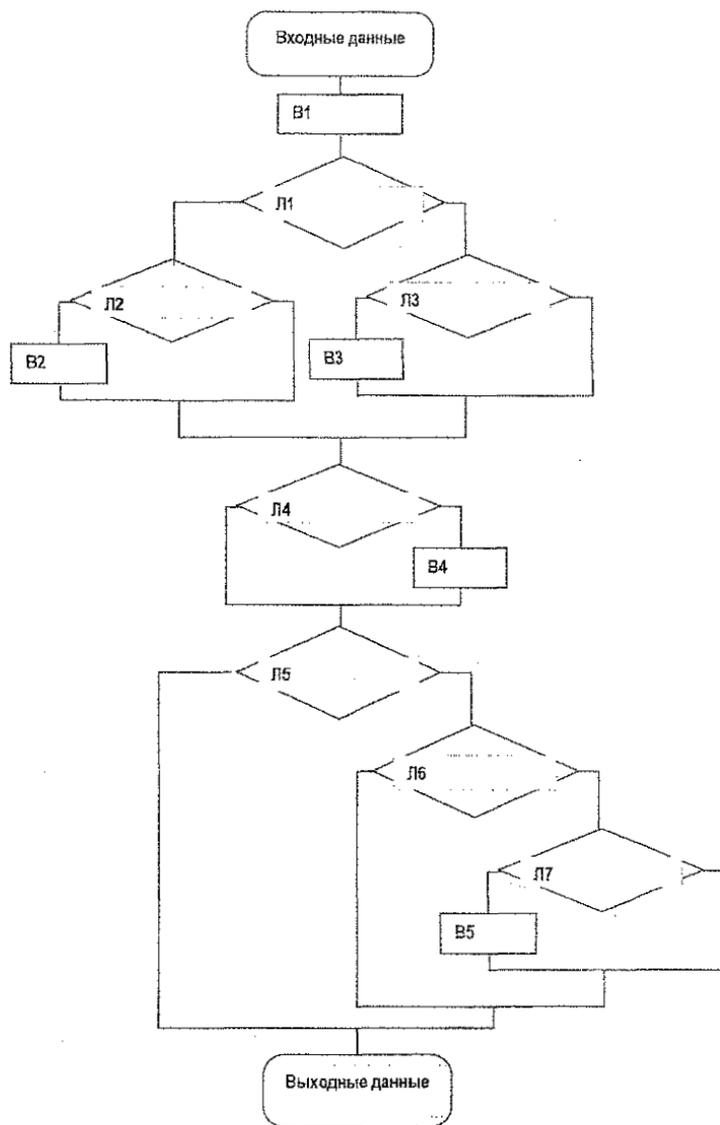


Рисунок 2.2 – Вариант А2

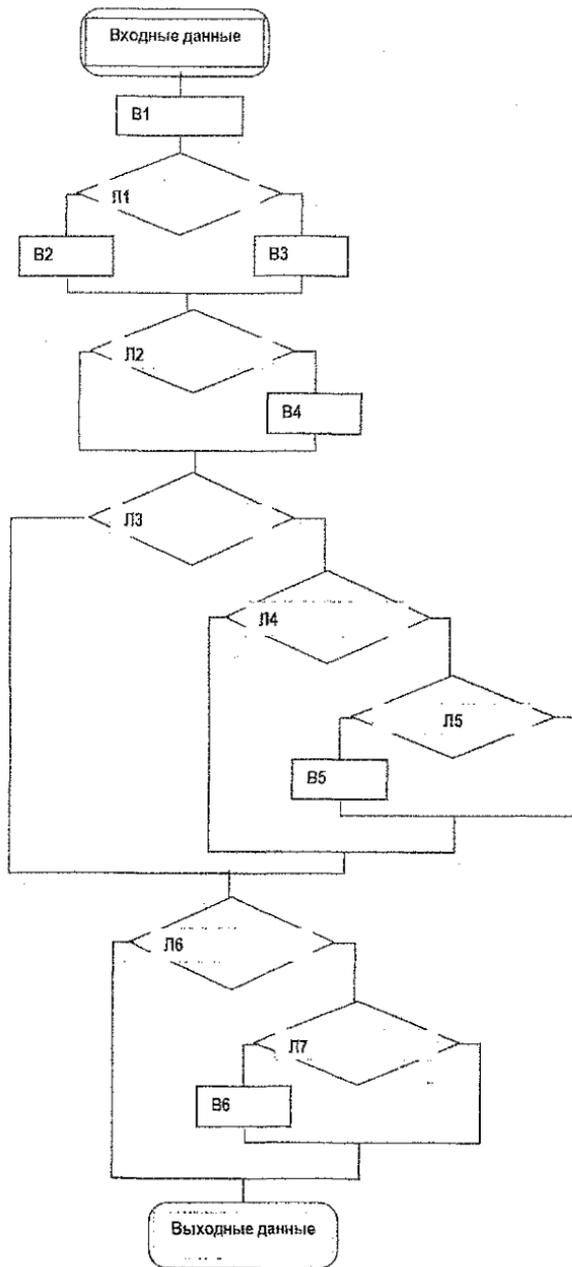


Рисунок 2.3 – Вариант А3

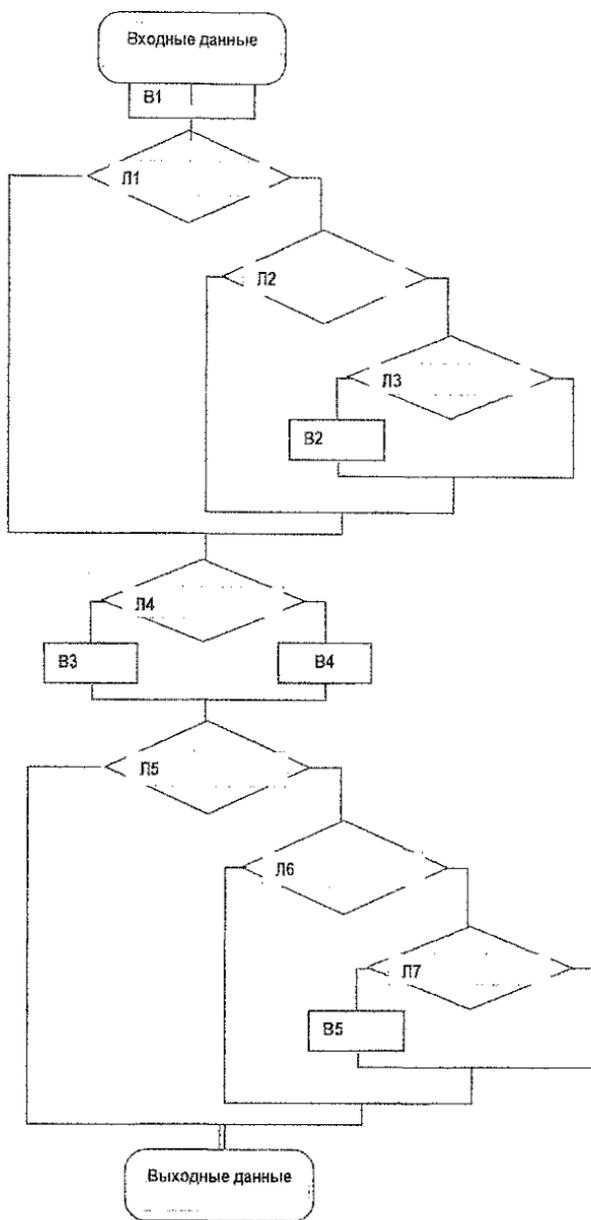


Рисунок 2.4 – Вариант А4

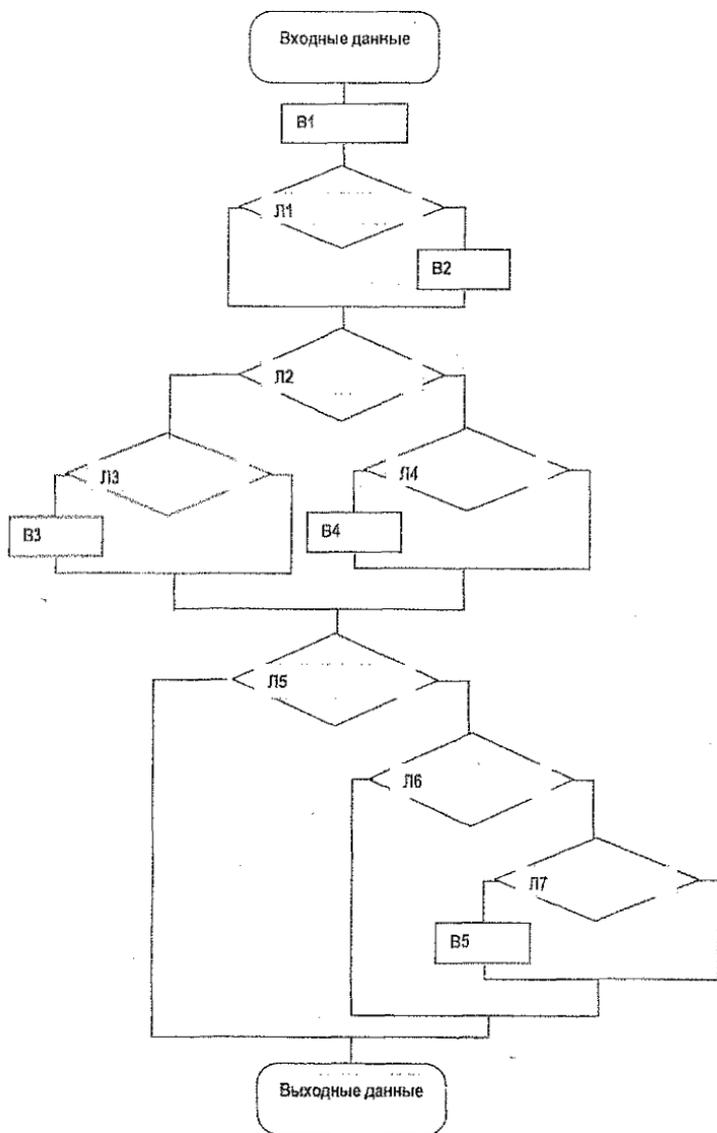


Рисунок 2.5 – Вариант А5

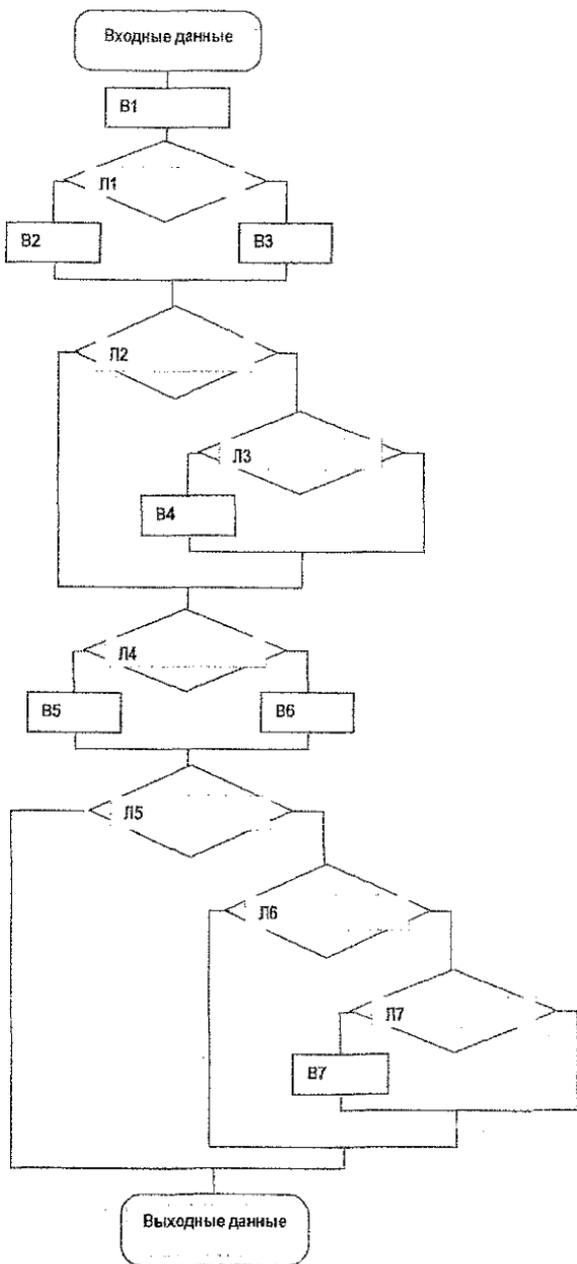


Рисунок 2.6 – Вариант А6

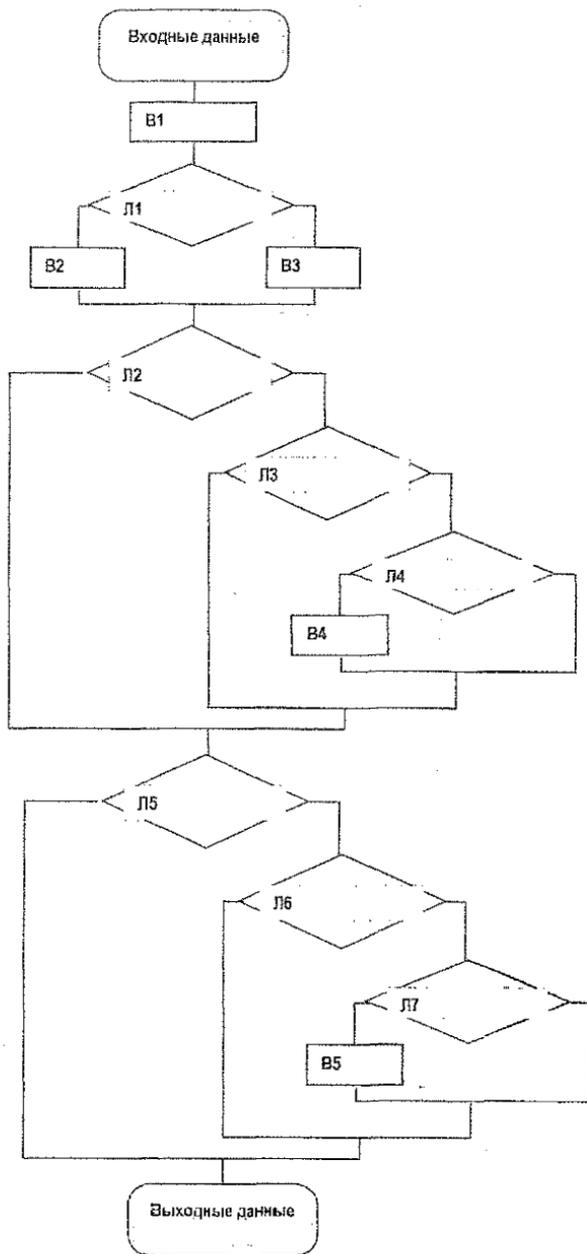


Рисунок 2.7 – Вариант А7

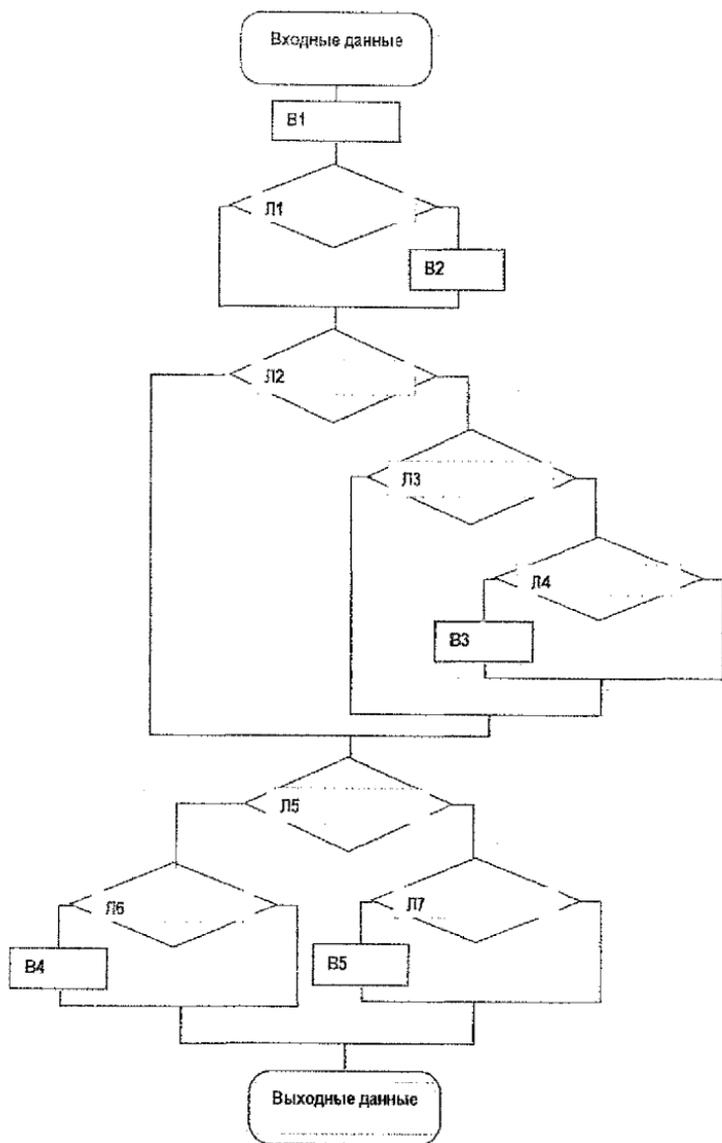


Рисунок 2.8 – Вариант А8

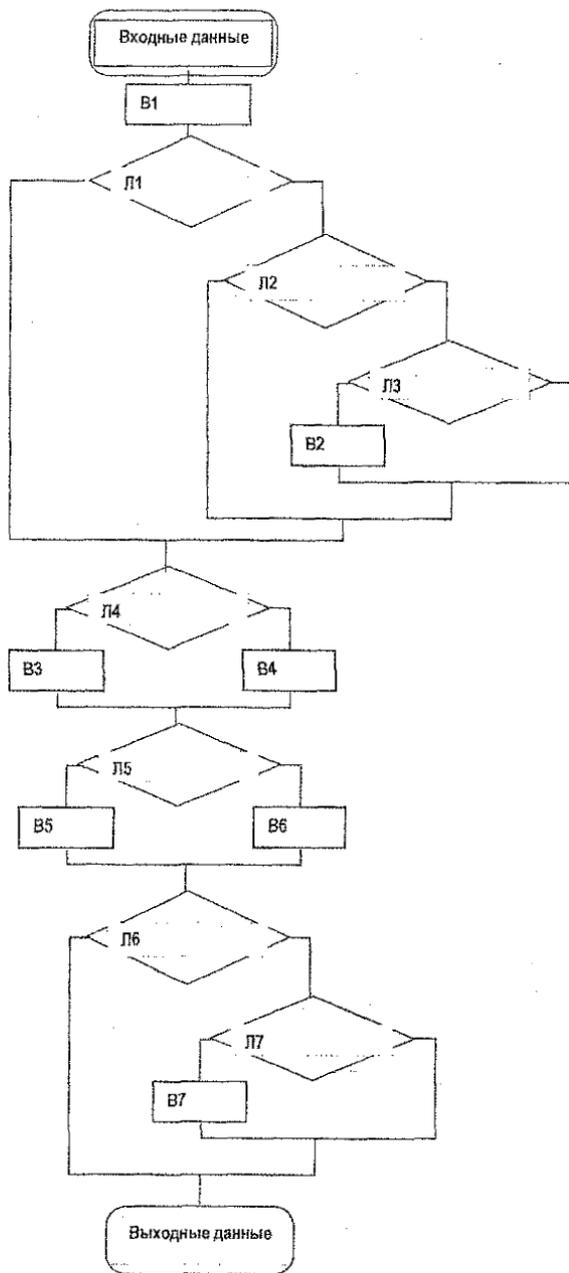


Рисунок 2.9 – Вариант А9

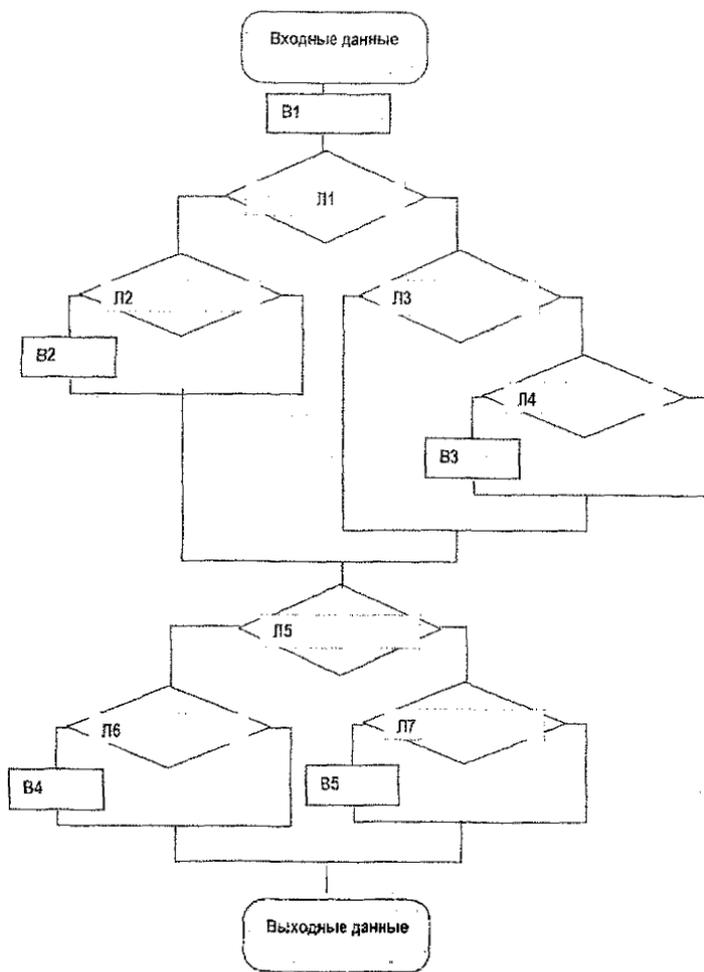


Рисунок 2.10 – Вариант А10

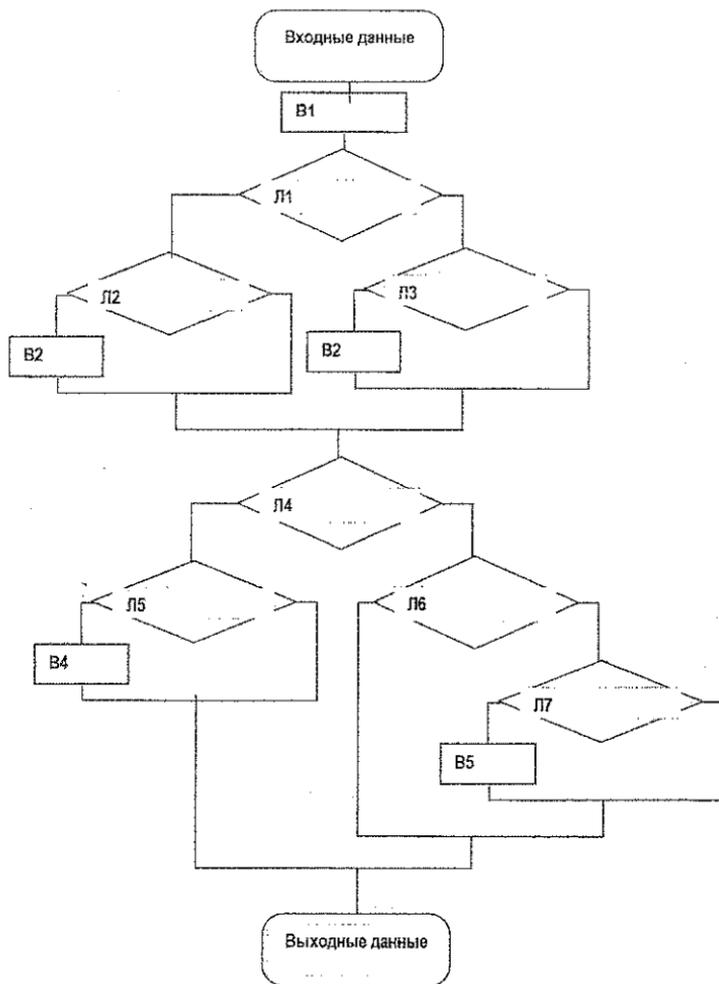


Рисунок 2.11 – Вариант А11

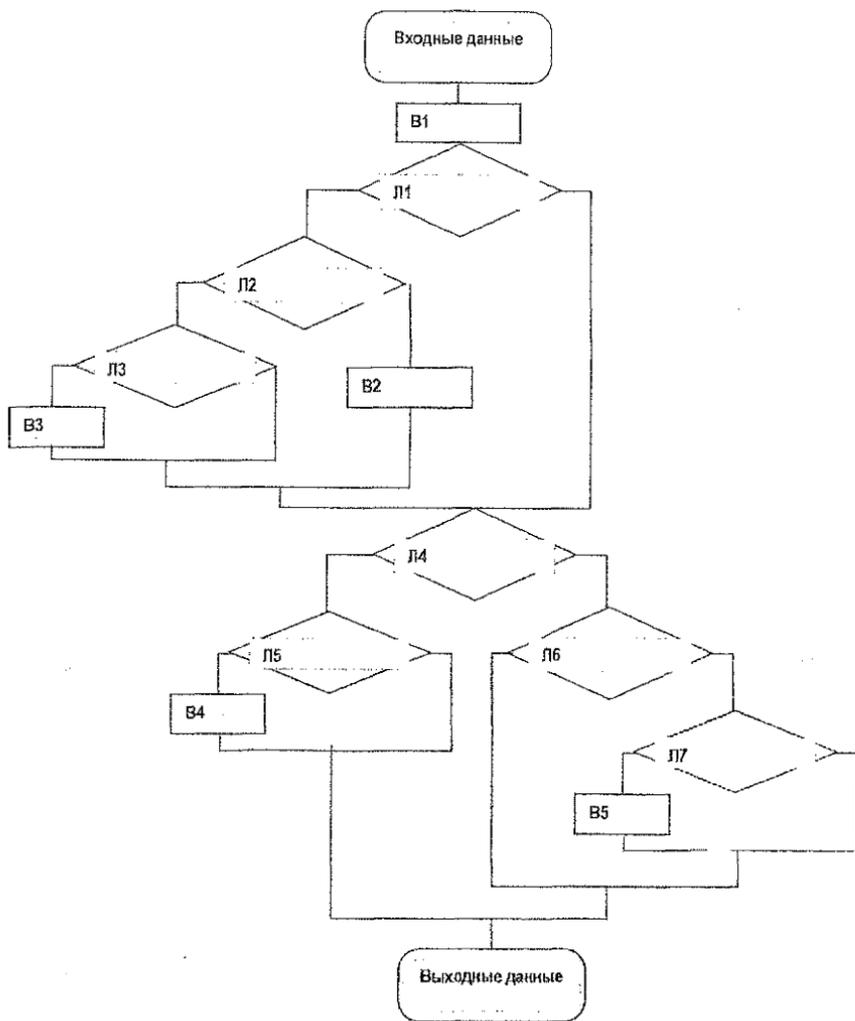


Рисунок 2.12 – Вариант А12

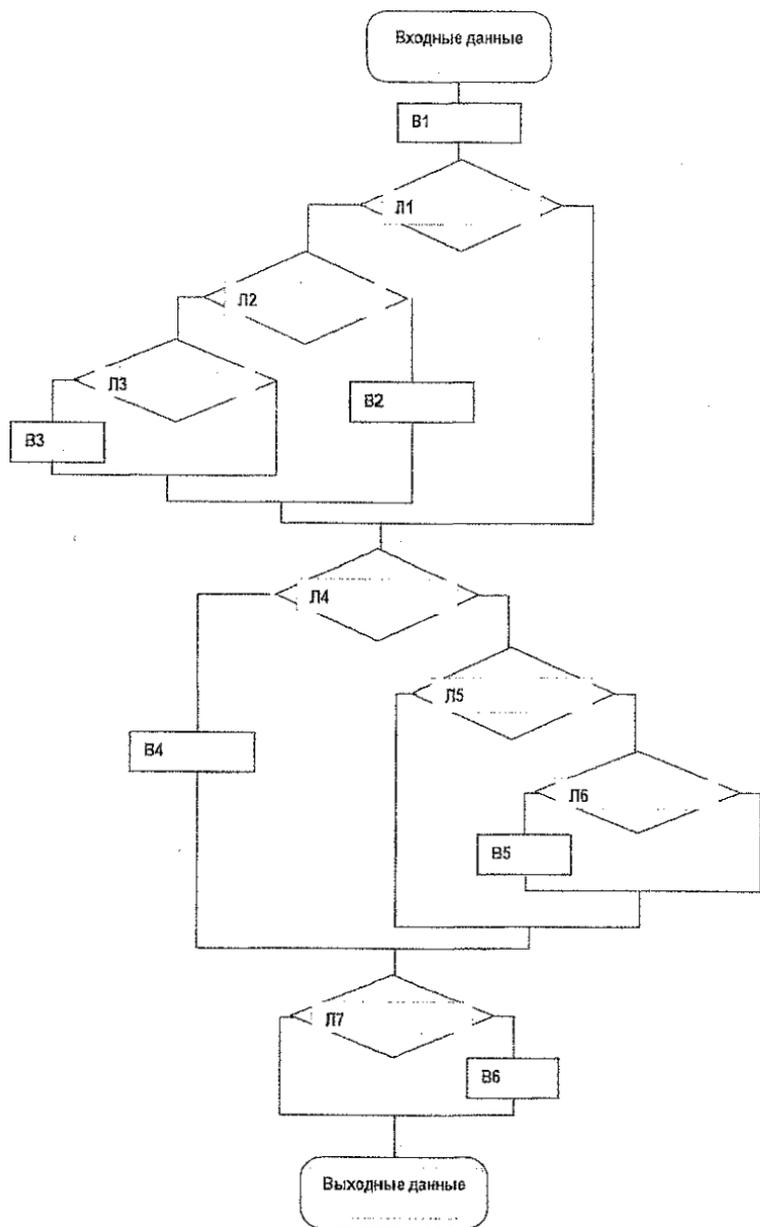


Рисунок 2.14 – Вариант В1

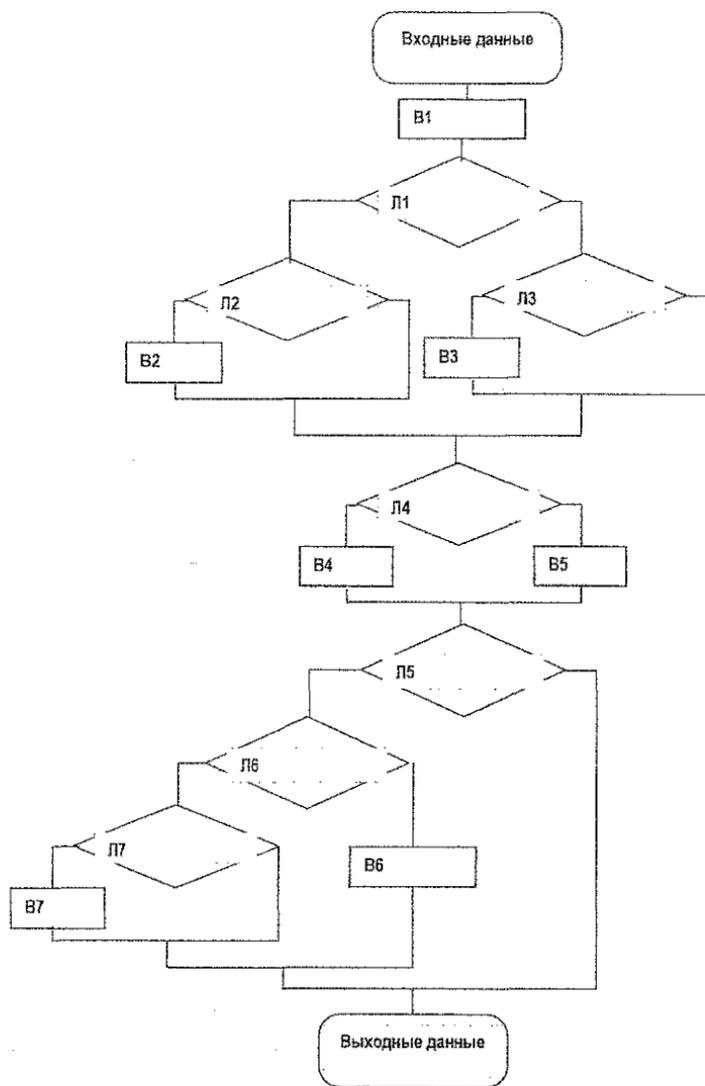


Рис 2.15 – Вариант В2

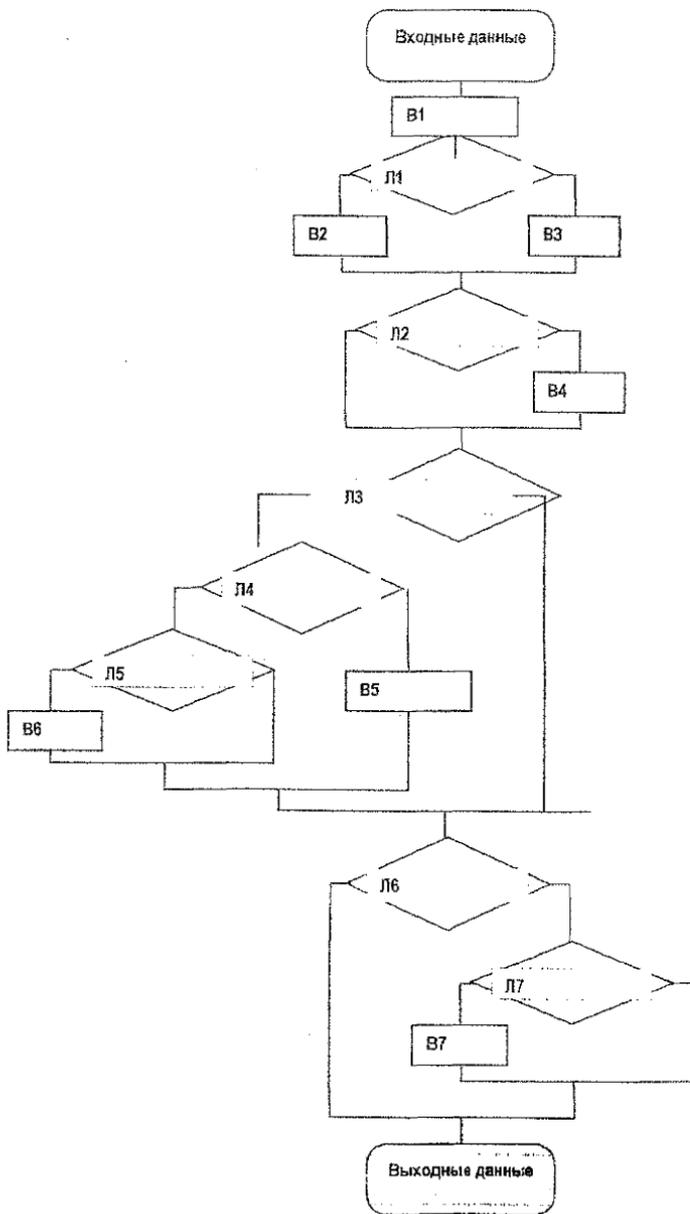


Рисунок 2.16 – Вариант В3

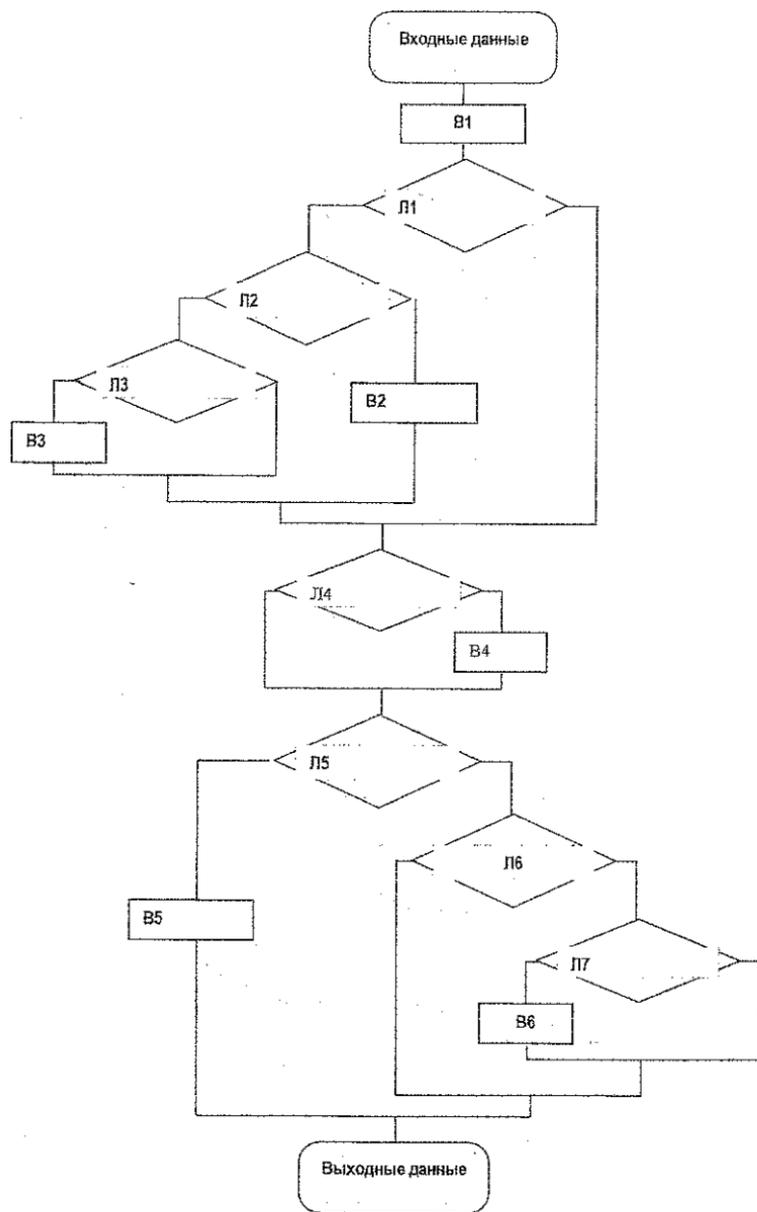


Рисунок 2.17 – Вариант В4

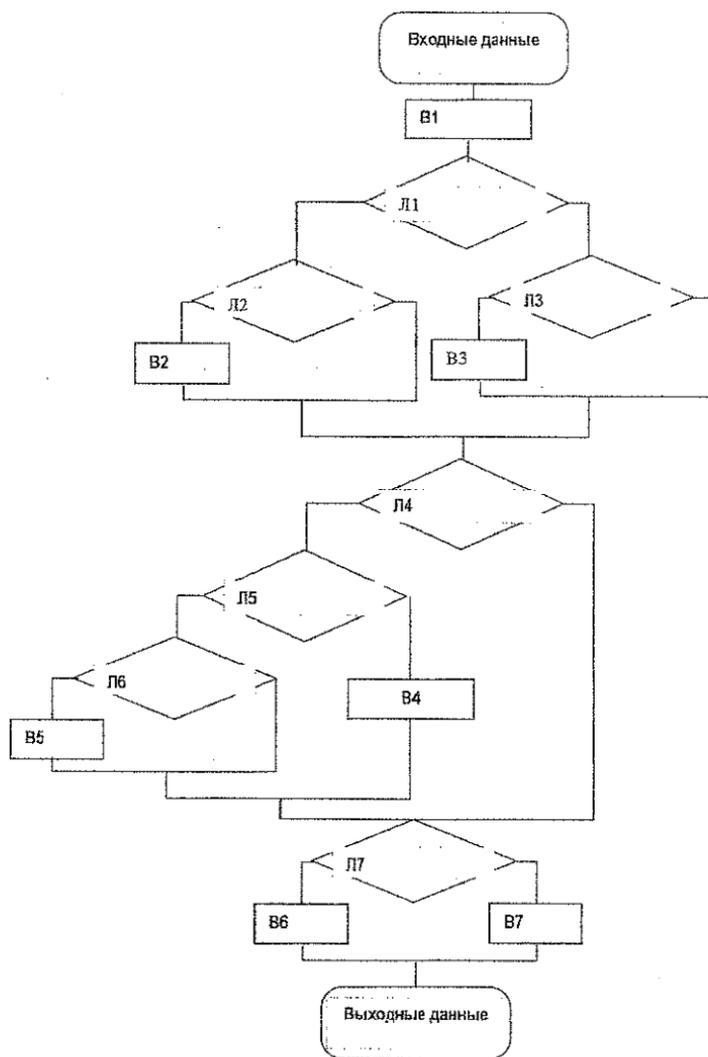


Рисунок 2.18 – Вариант В5

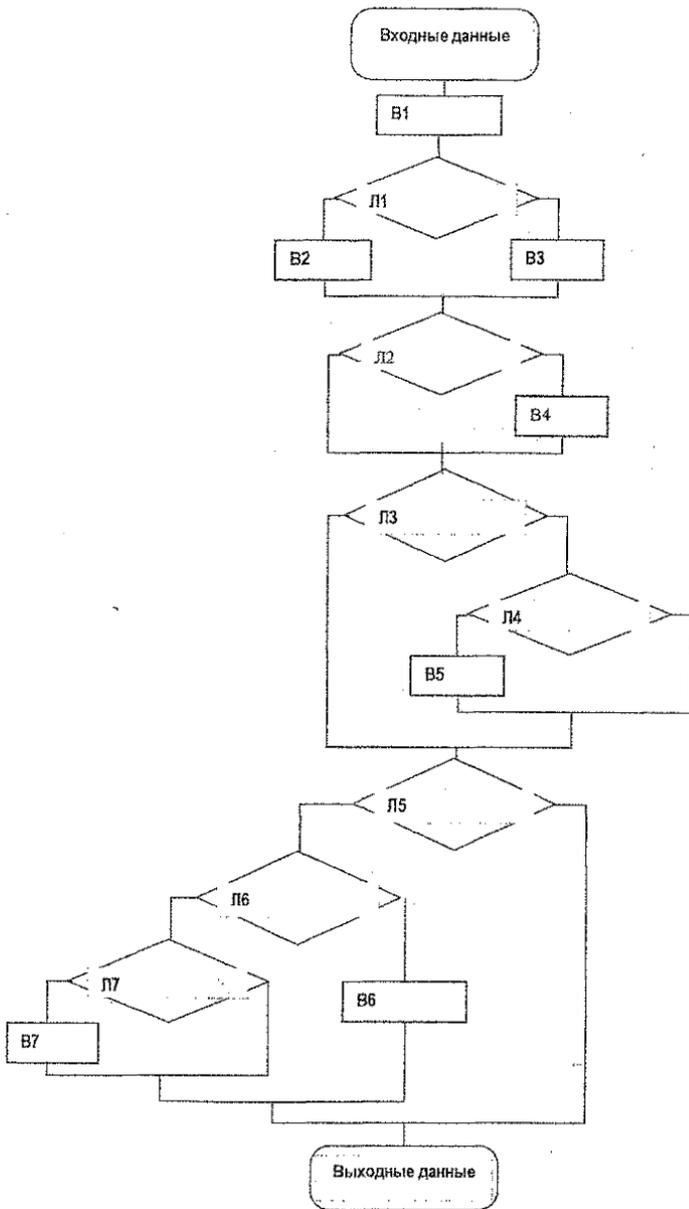


Рисунок 2.19 – Вариант В6

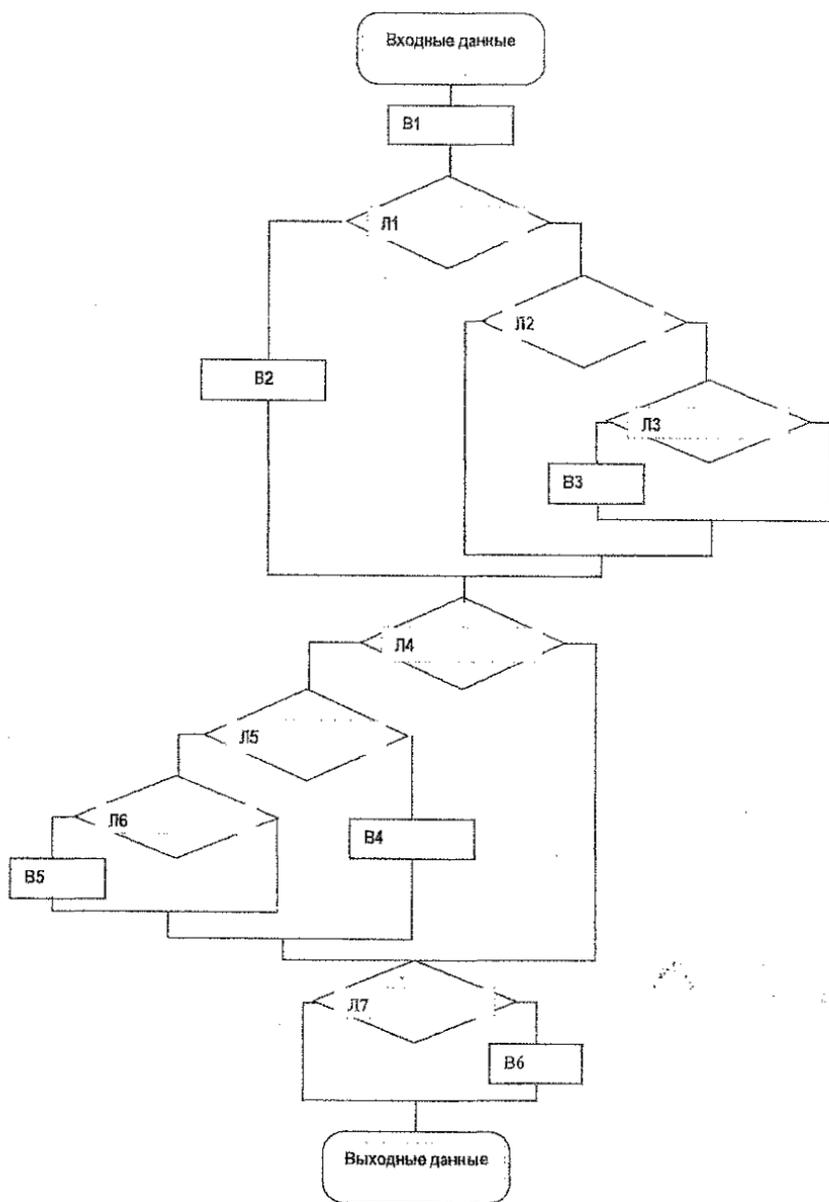


Рисунок 2.20 -- Вариант В7

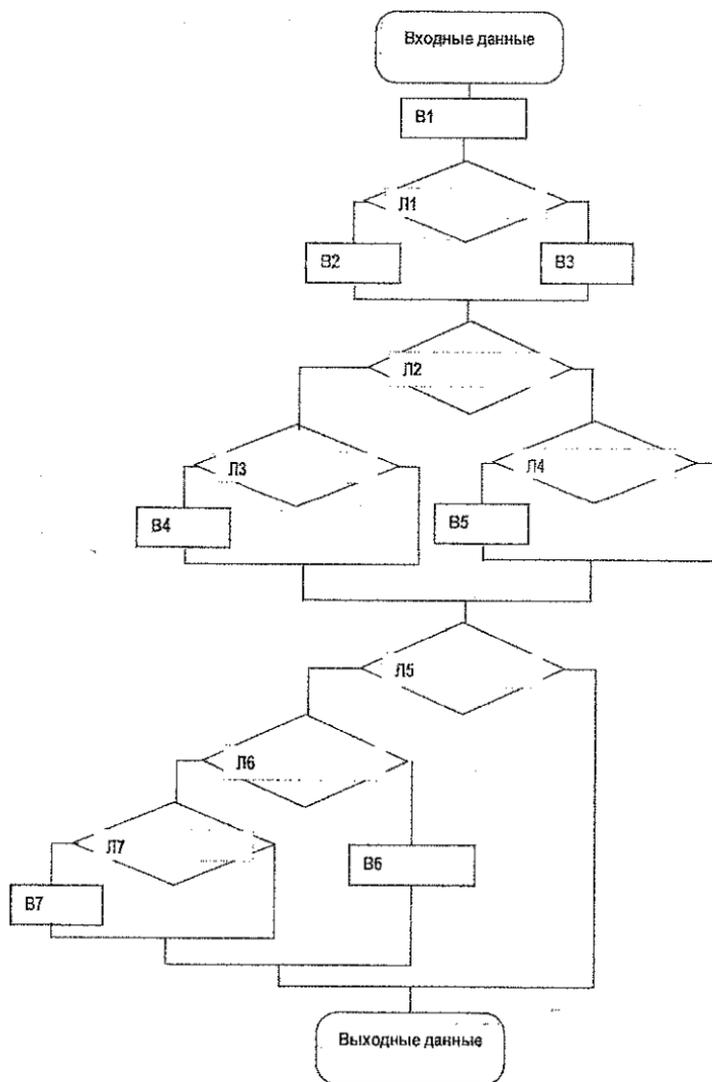


Рисунок 2.21 – Вариант В8

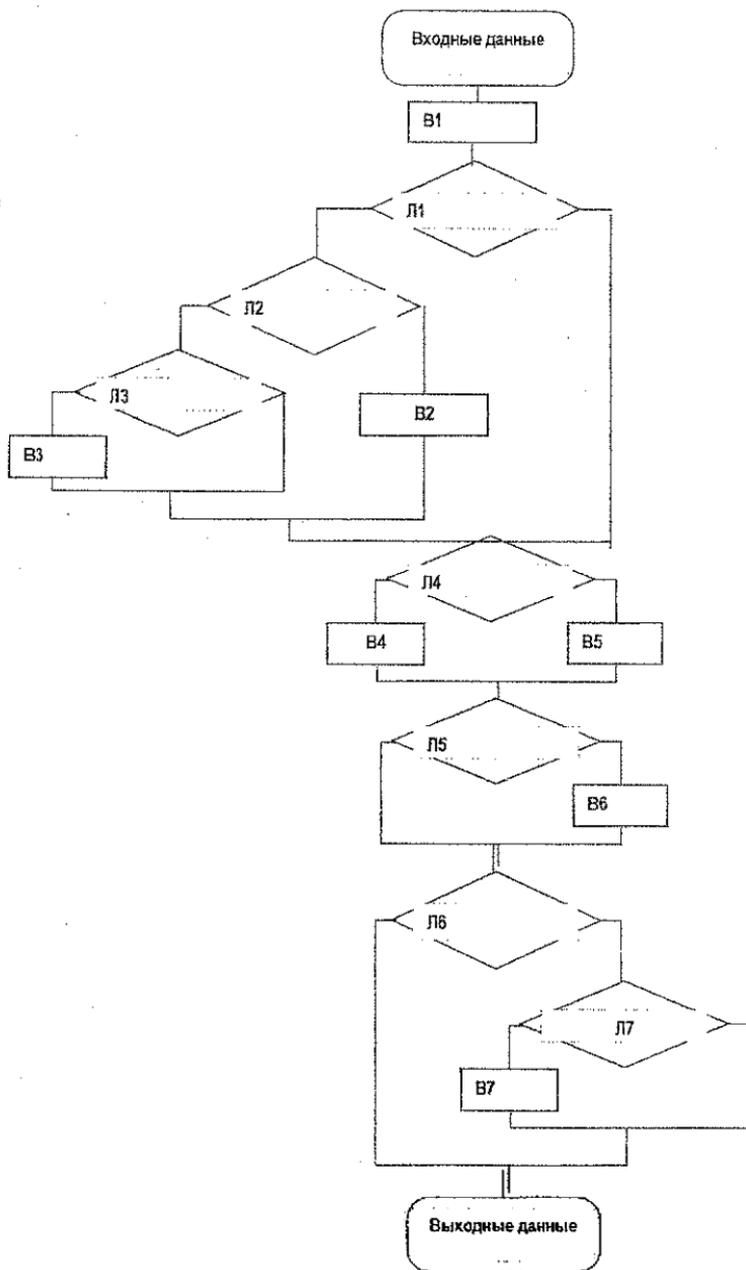


Рисунок 2.22 – Вариант В9

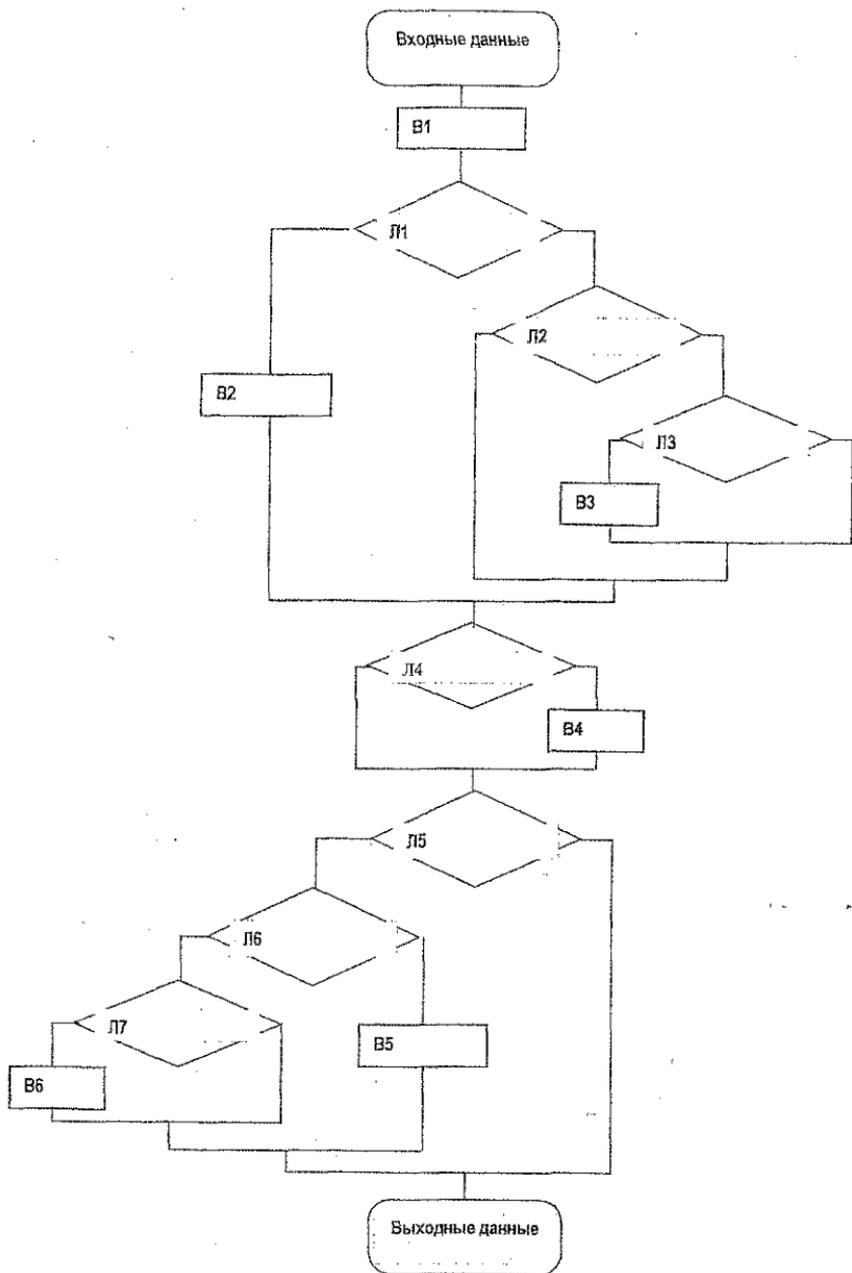


Рисунок 2.23 – Вариант В10

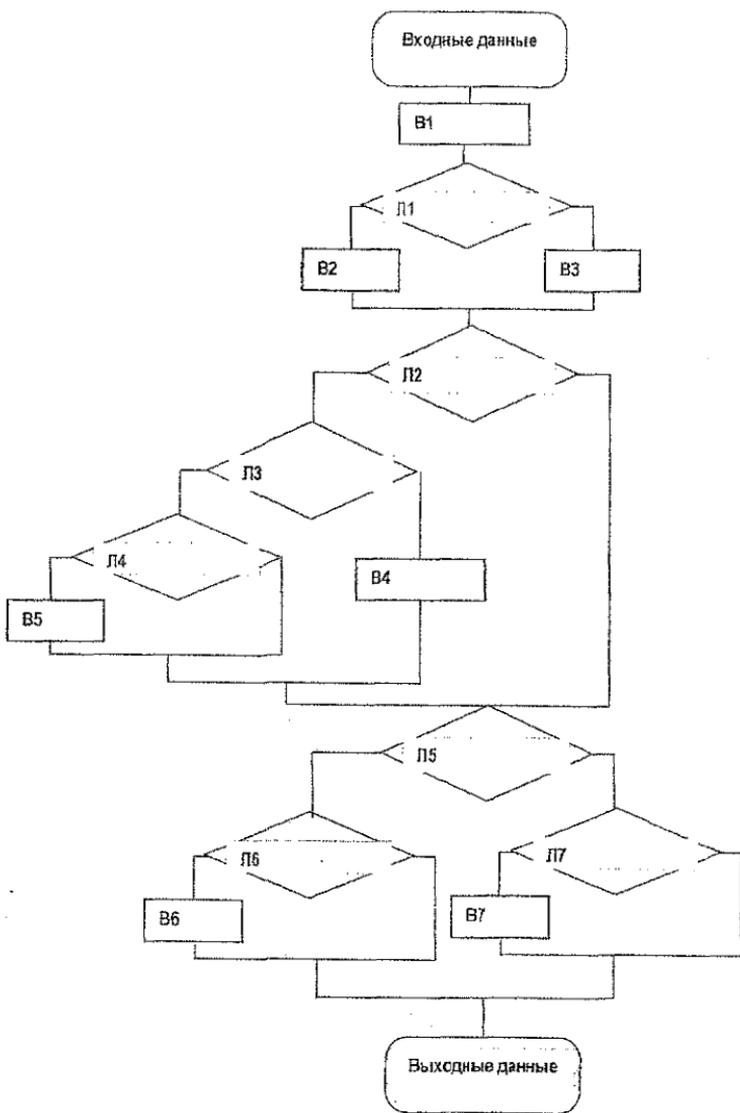


Рисунок 2.24 – Вариант В11

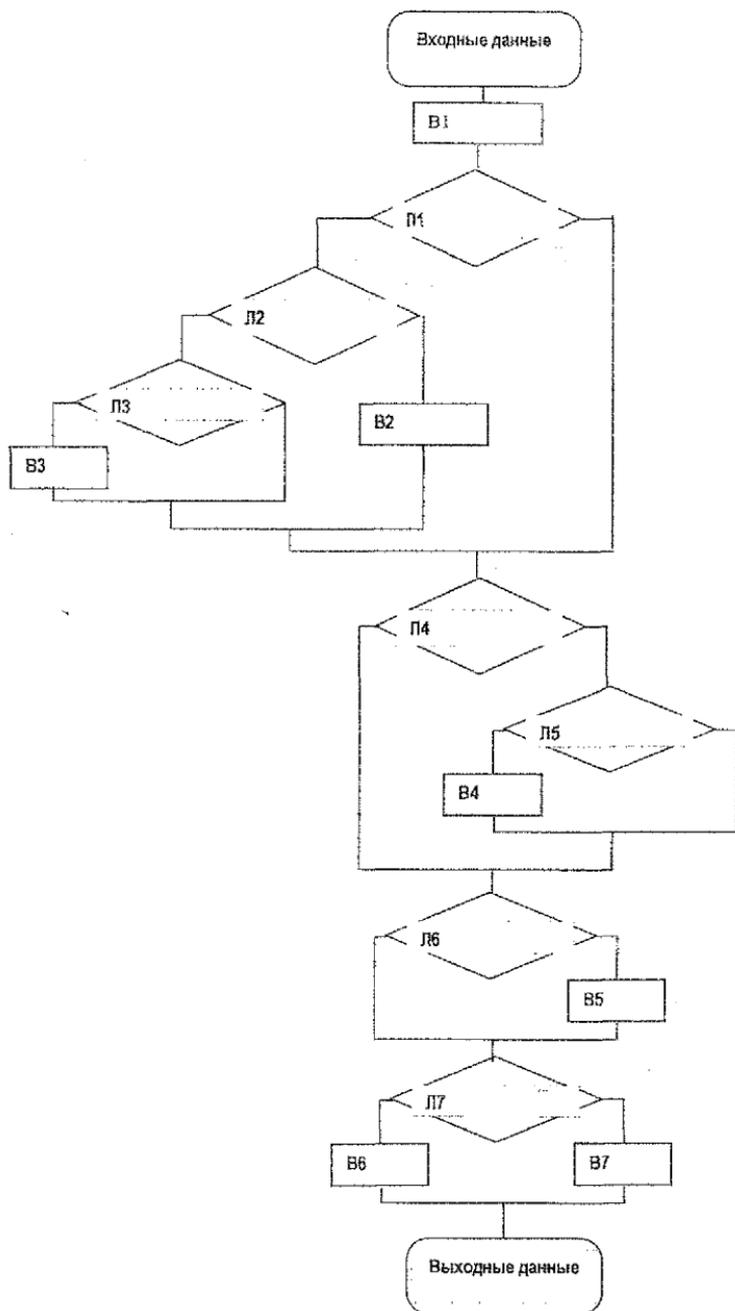


Рисунок 2.25 – Вариант В12

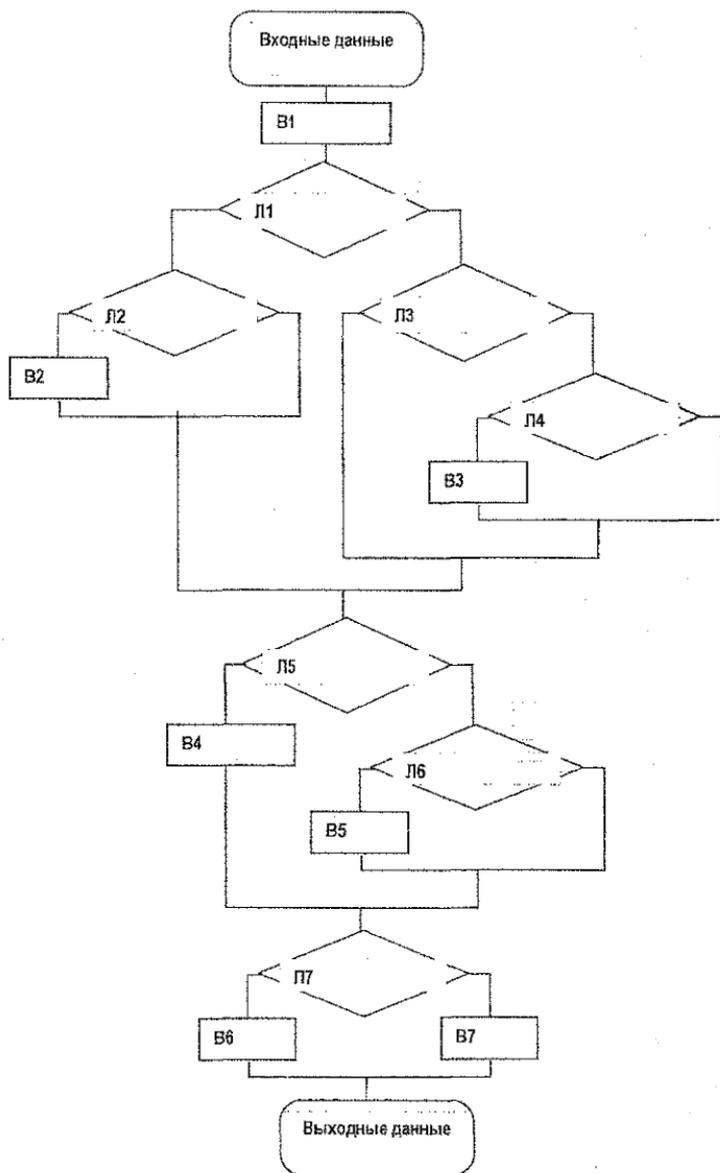


Рисунок 2.26 – Вариант В13

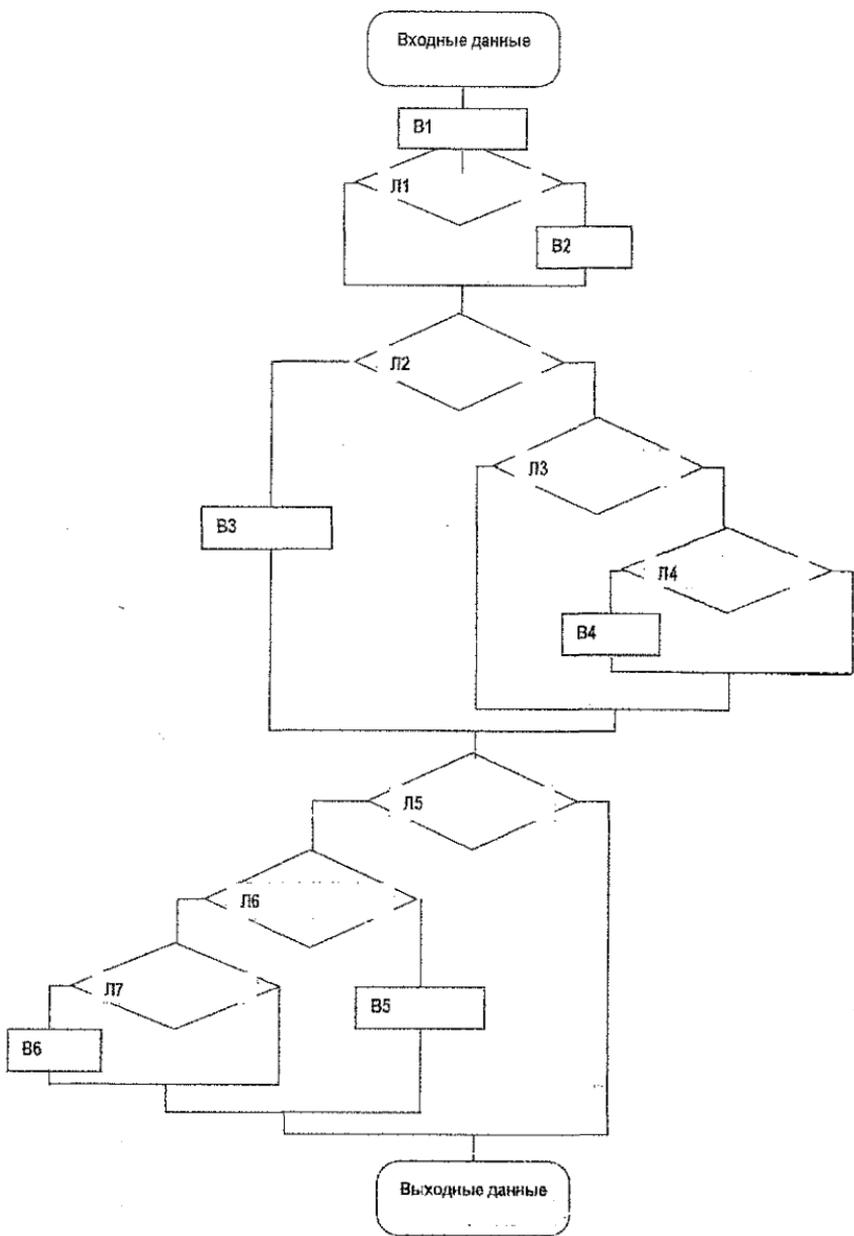


Рисунок 2.27 – Вариант С1

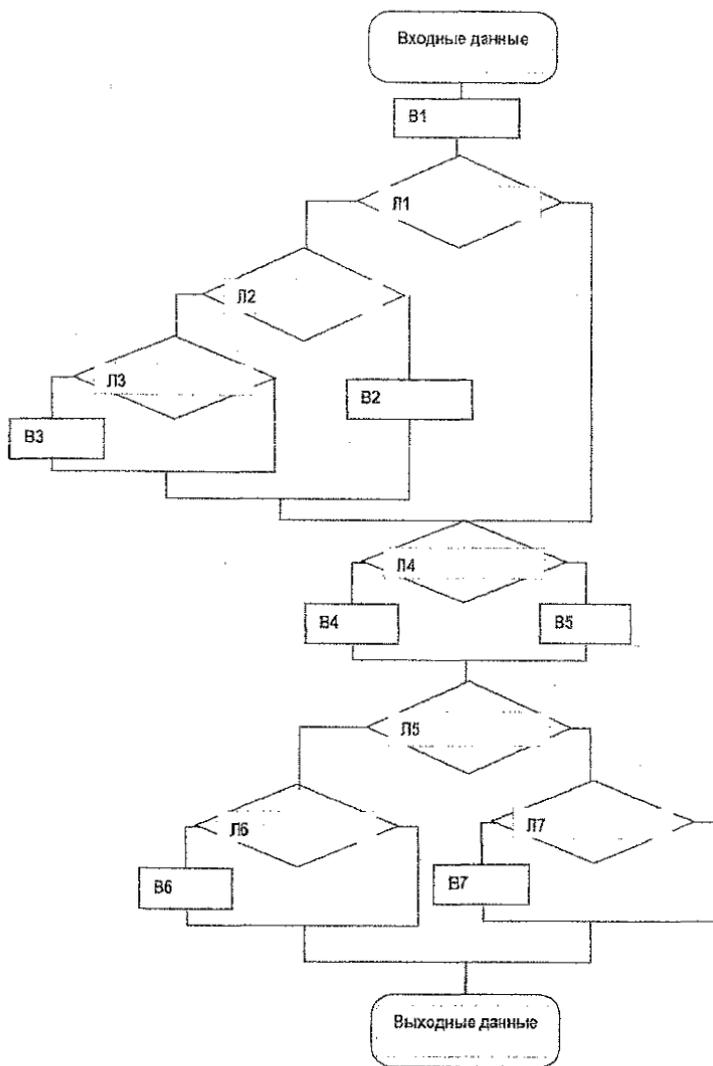


Рисунок 2.28 – Вариант С2

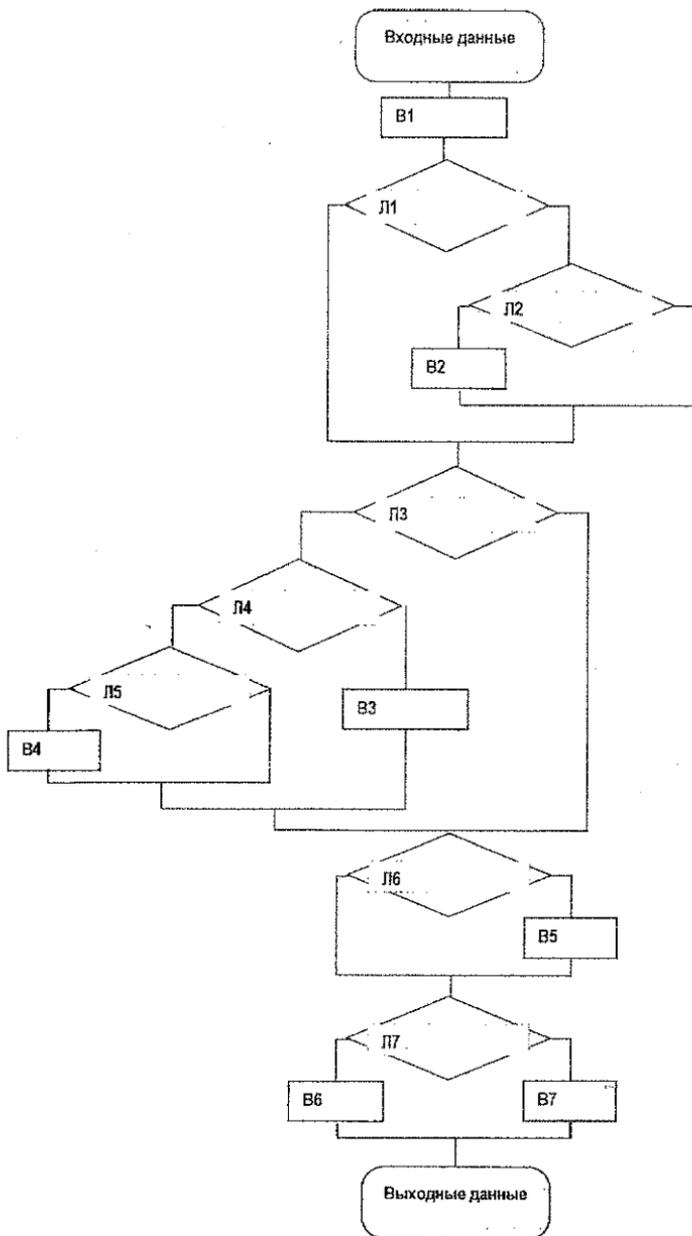


Рисунок 2.29 – Вариант С3

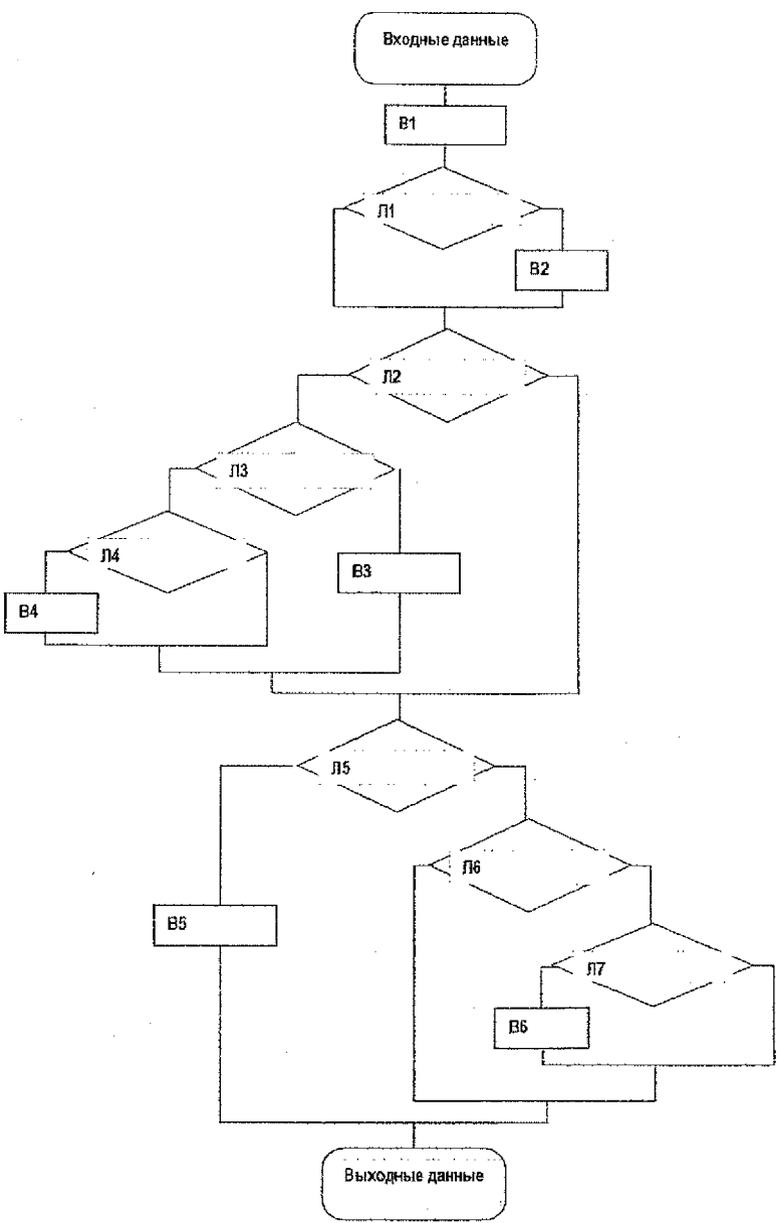


Рисунок 2.30 – Вариант С4

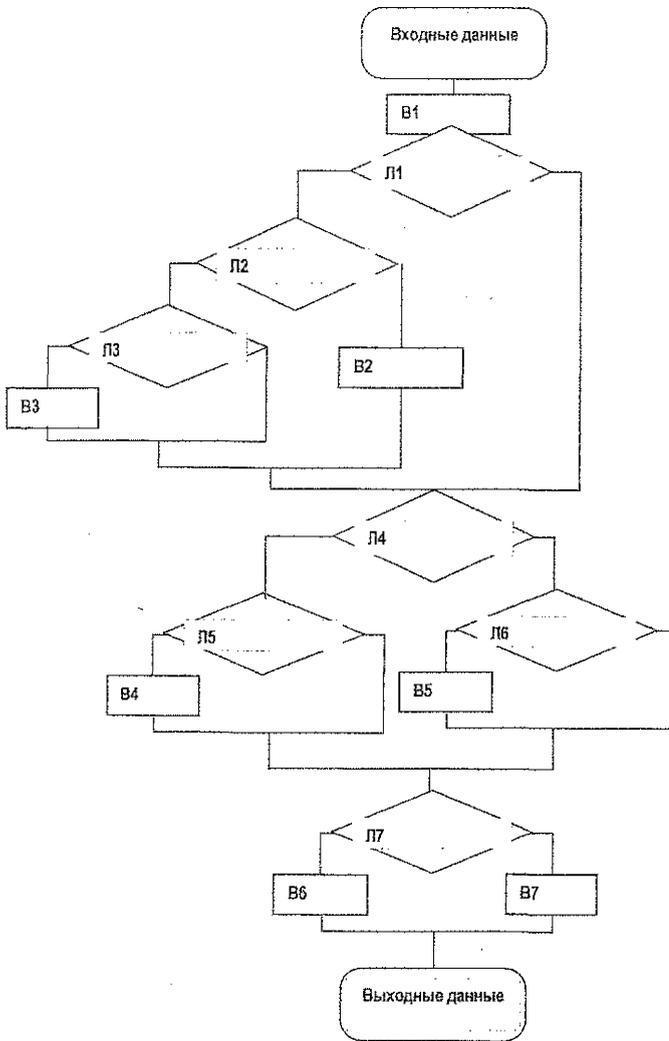


Рисунок 2.31 – Вариант С5

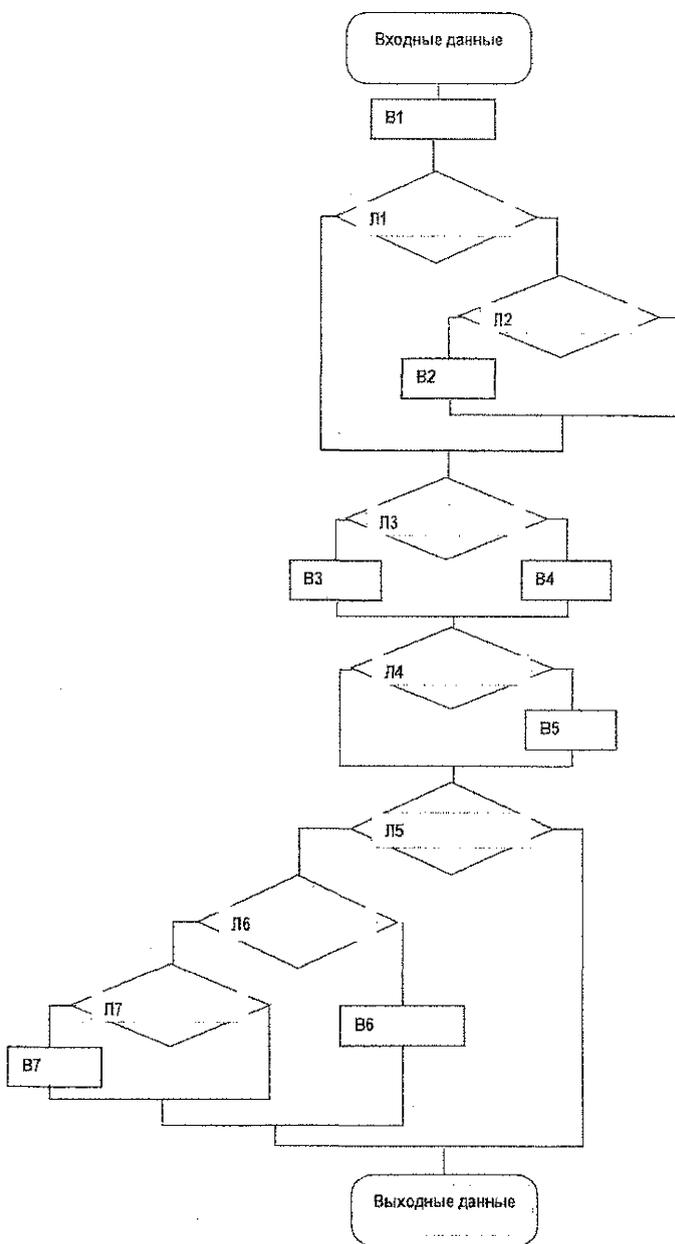


Рисунок 2.32 – Вариант С6

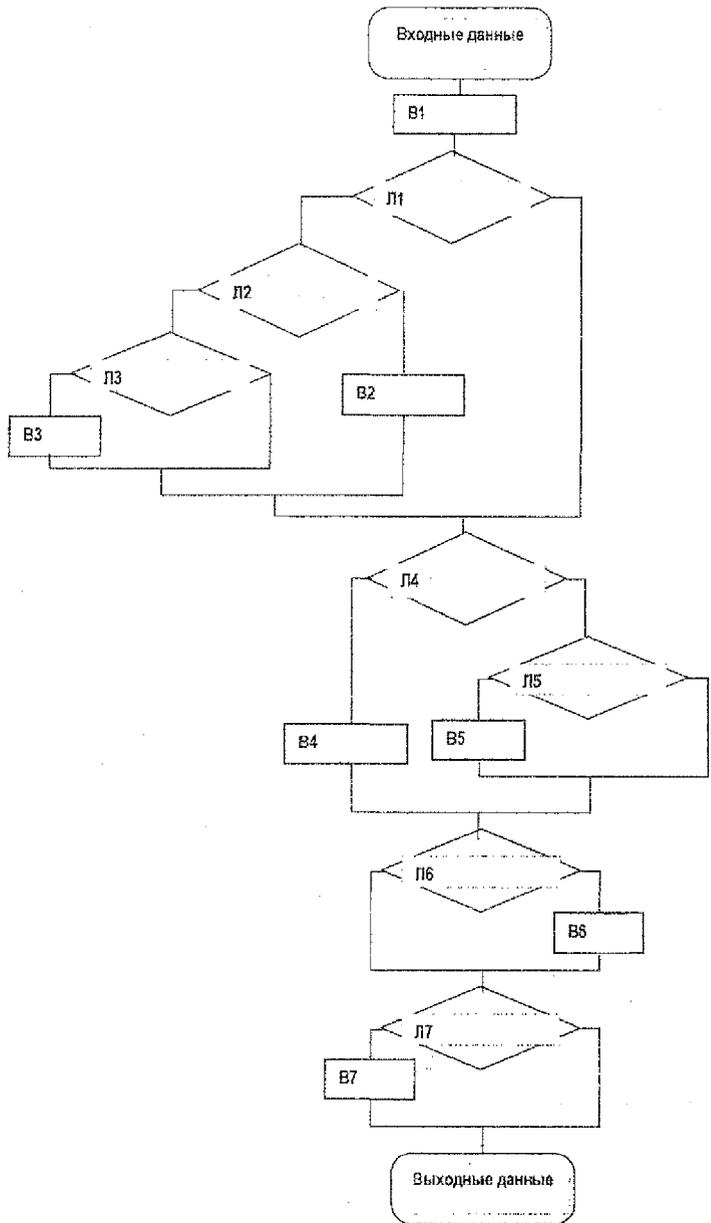


Рисунок 2.33 – Вариант С7

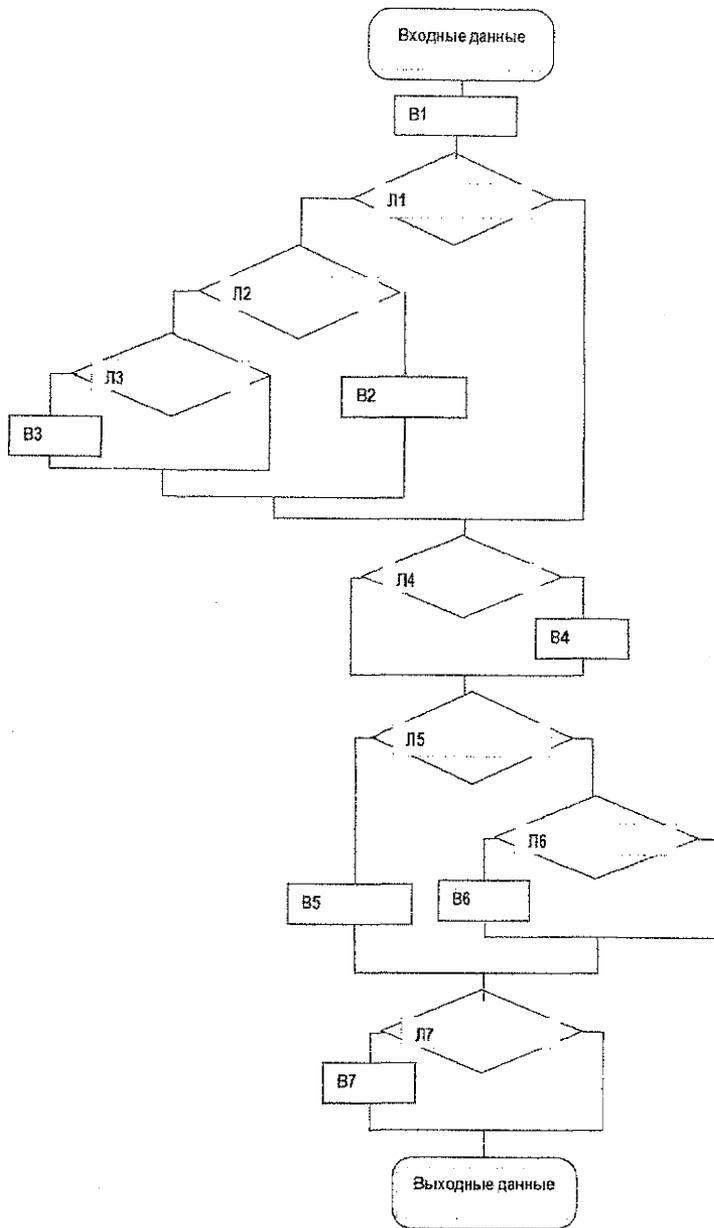


Рисунок 2.34 – Вариант С8

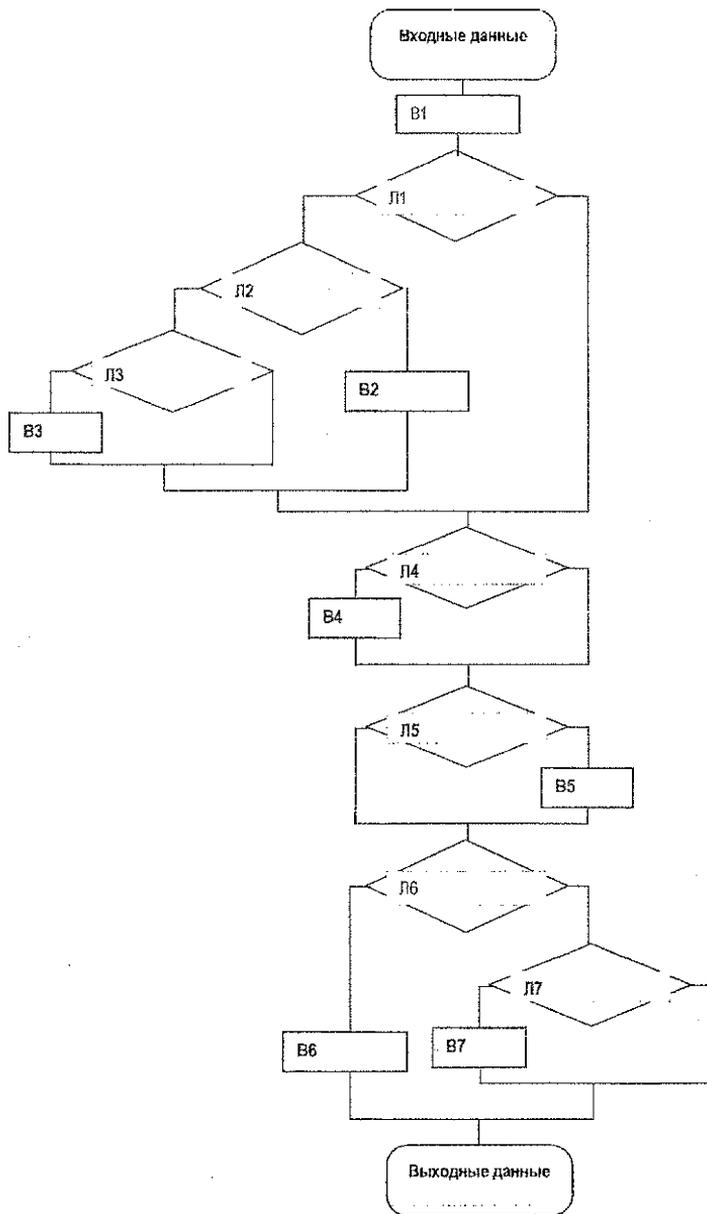


Рисунок 2.35 – Вариант С9

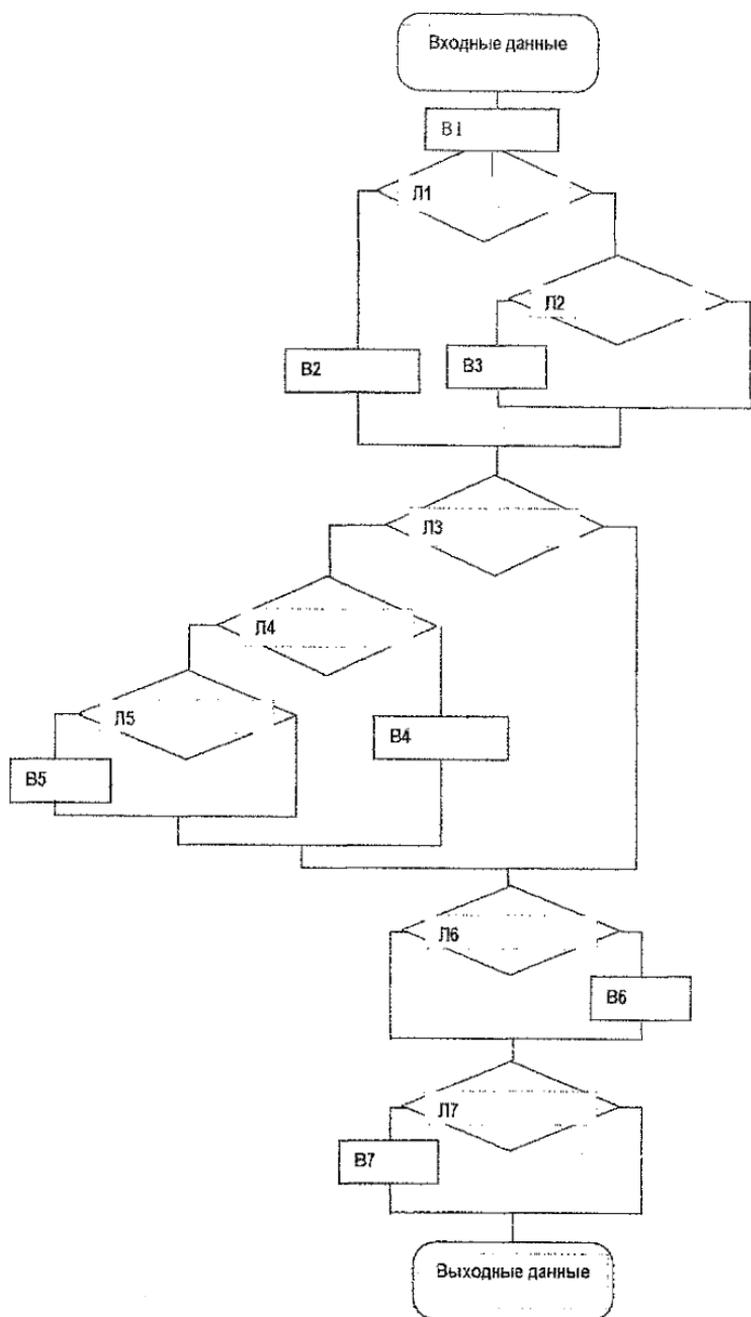


Рисунок 2.36 – Вариант С10

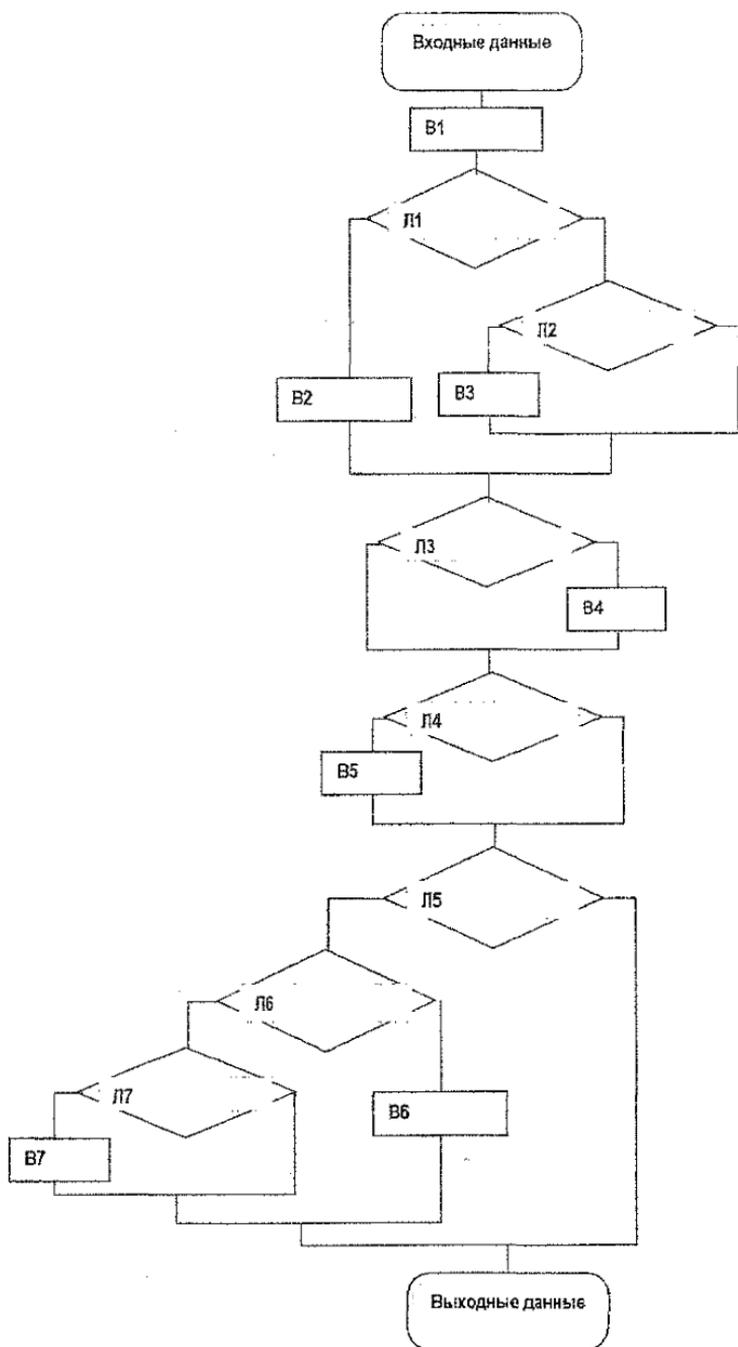


Рисунок 2.37 – Вариант С11

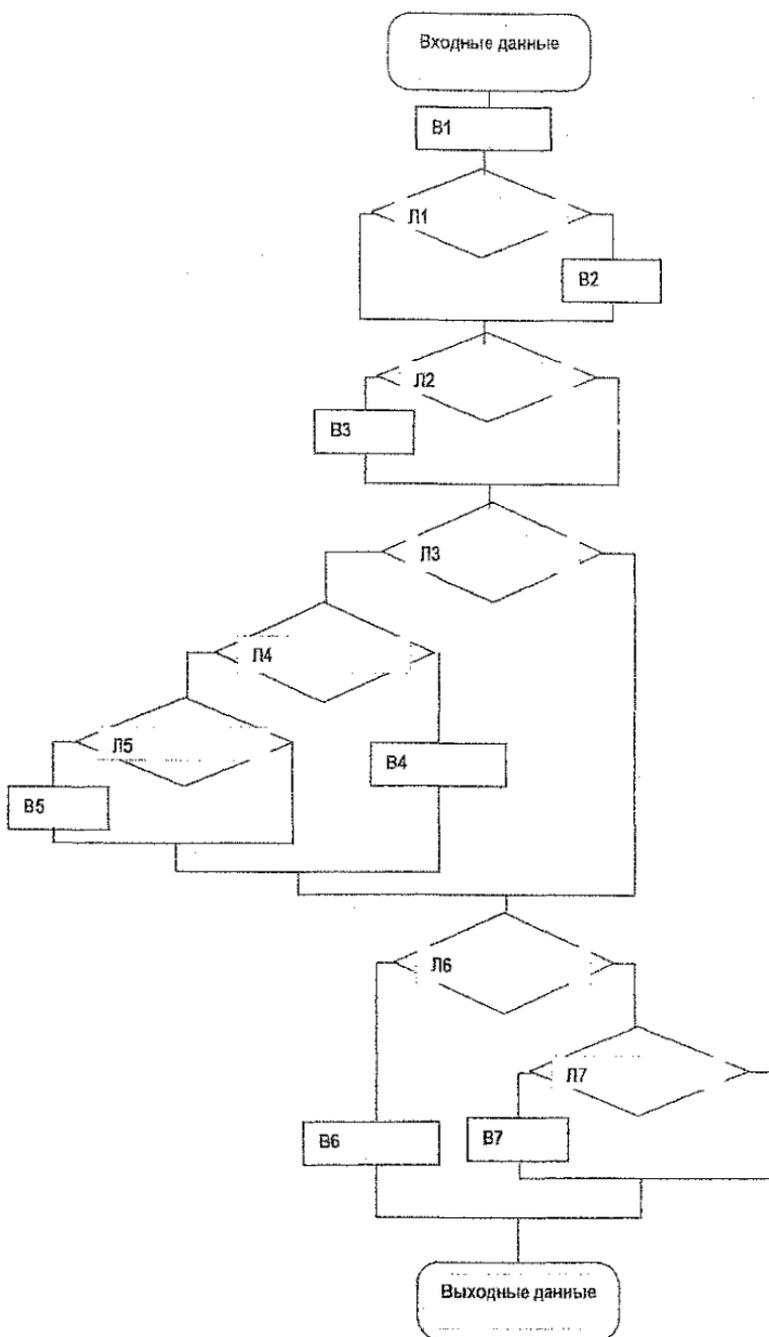


Рисунок 2.38 – Вариант С12

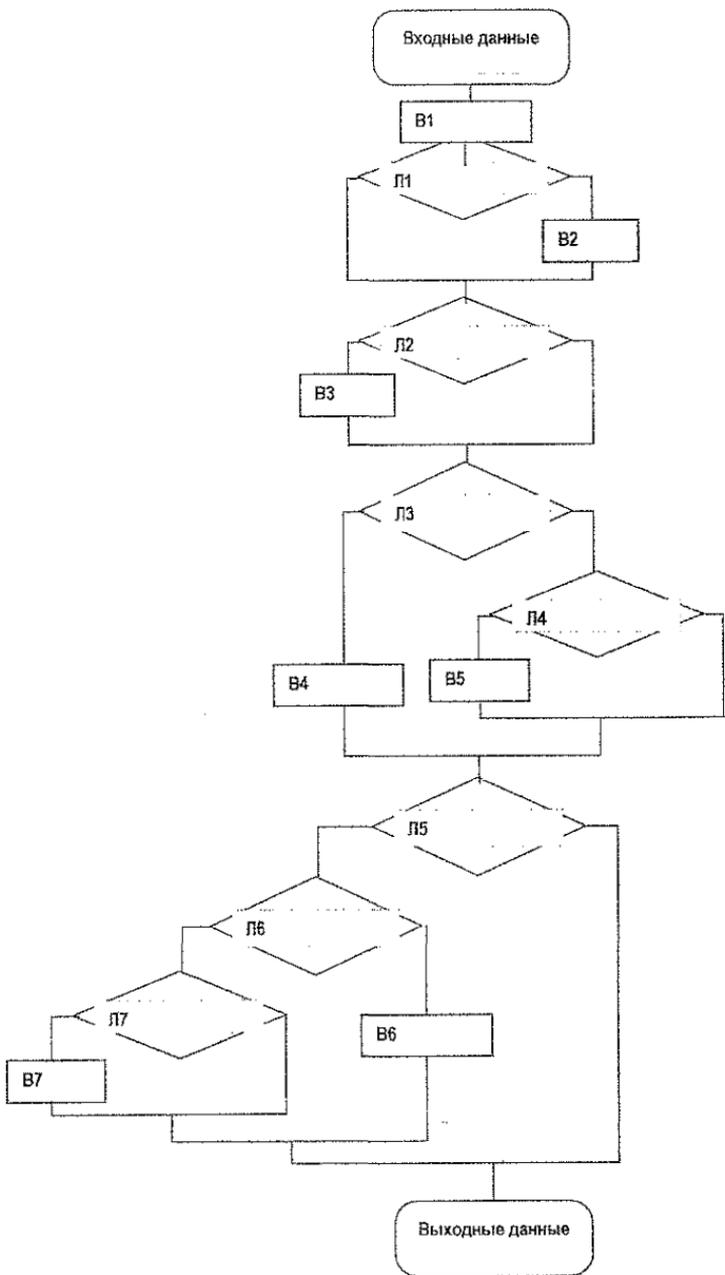


Рисунок 2.39 – Вариант С13

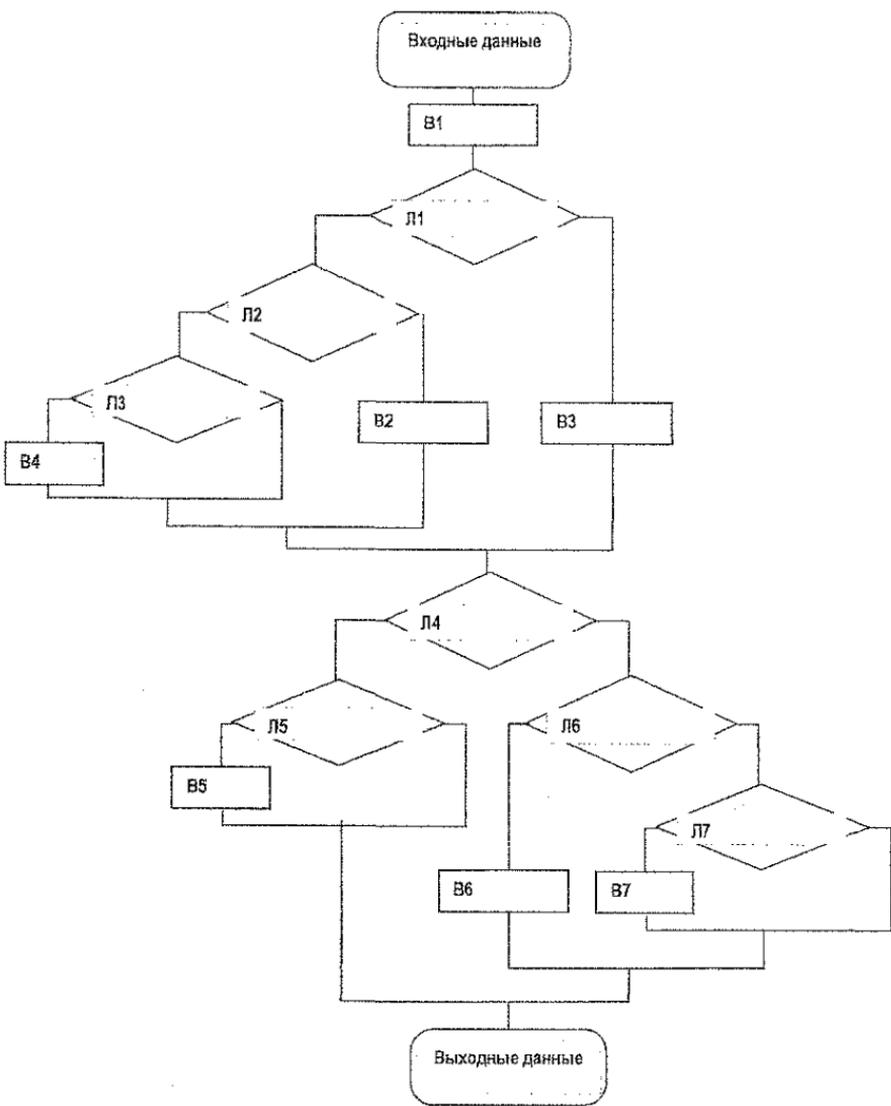


Рисунок 2.40 – Вариант D1

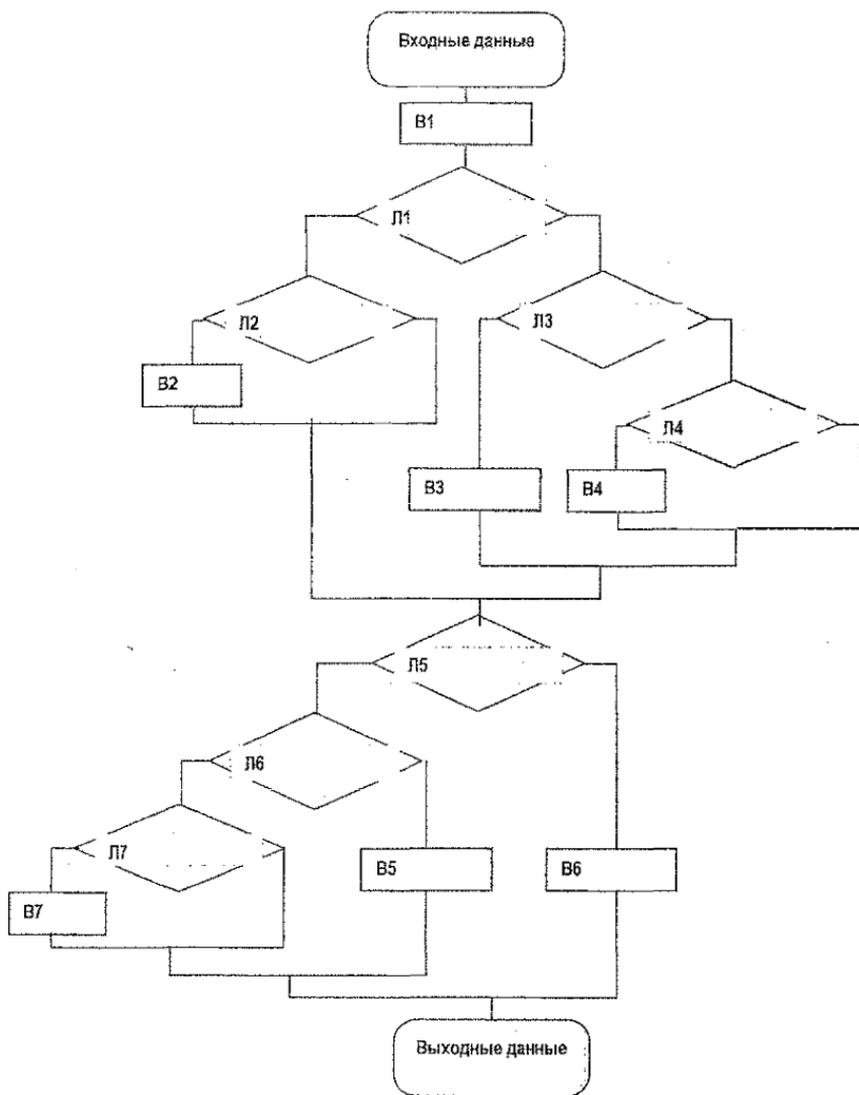


Рисунок 2.41 – Вариант D2

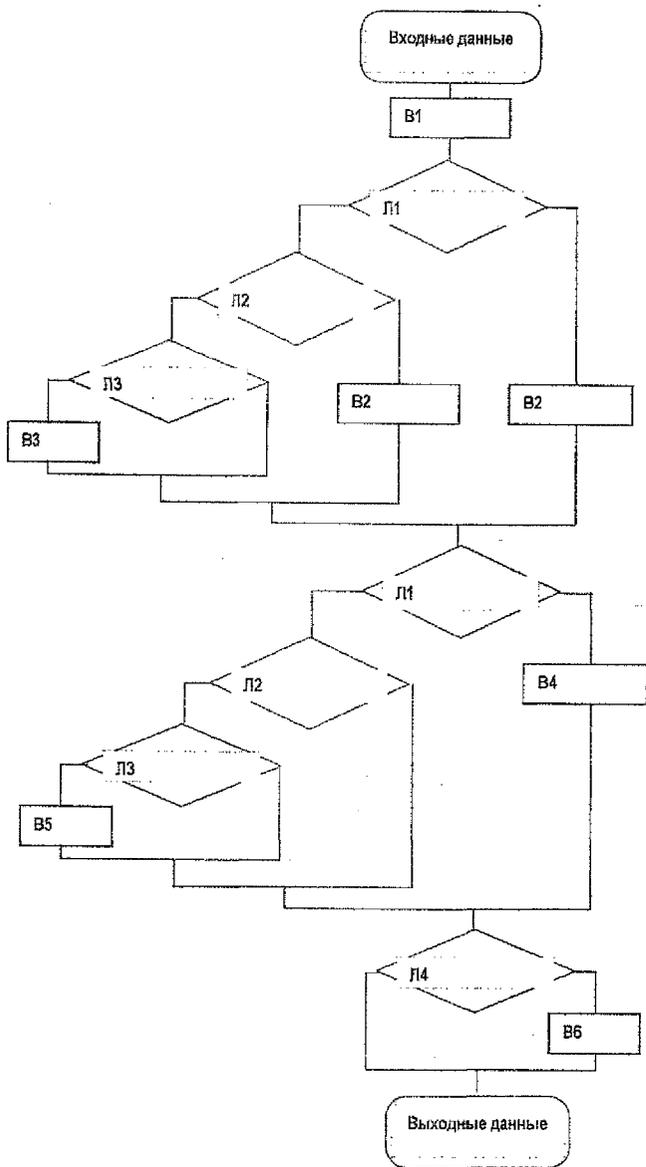


Рисунок 2.42 – Вариант Д3

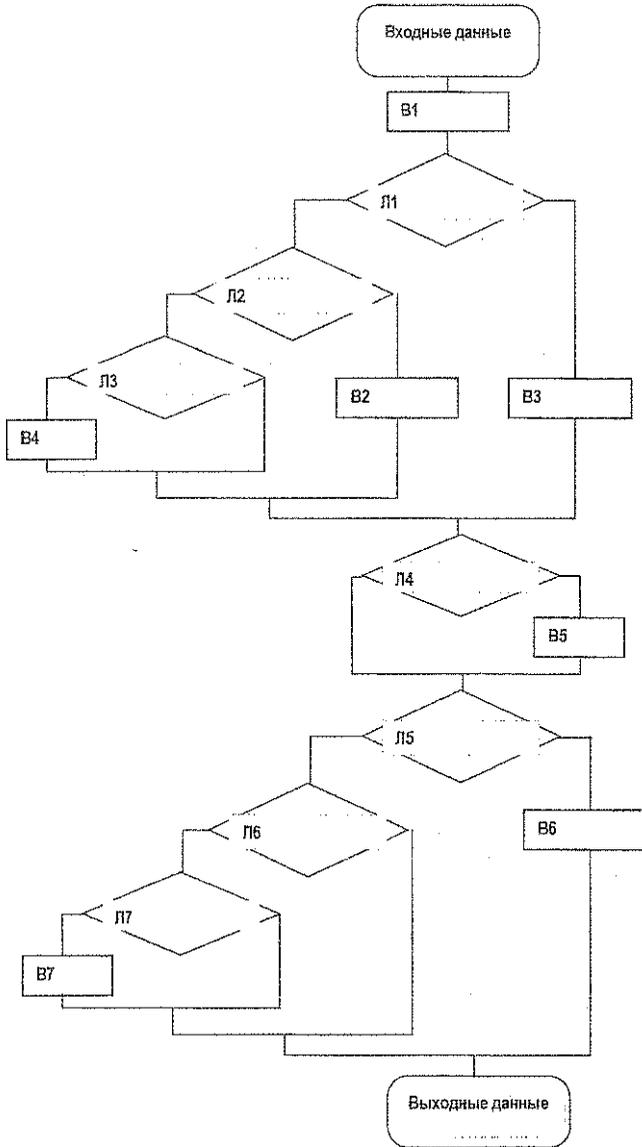


Рисунок 2.43 – Вариант D4

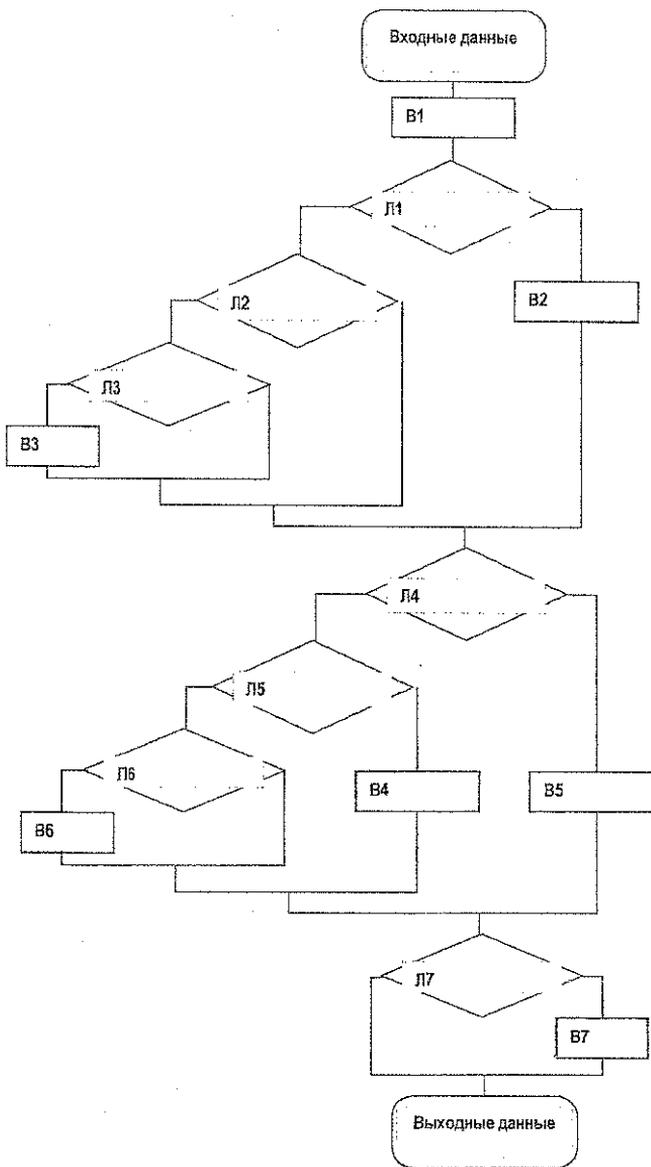


Рисунок 2.44 – Вариант D5

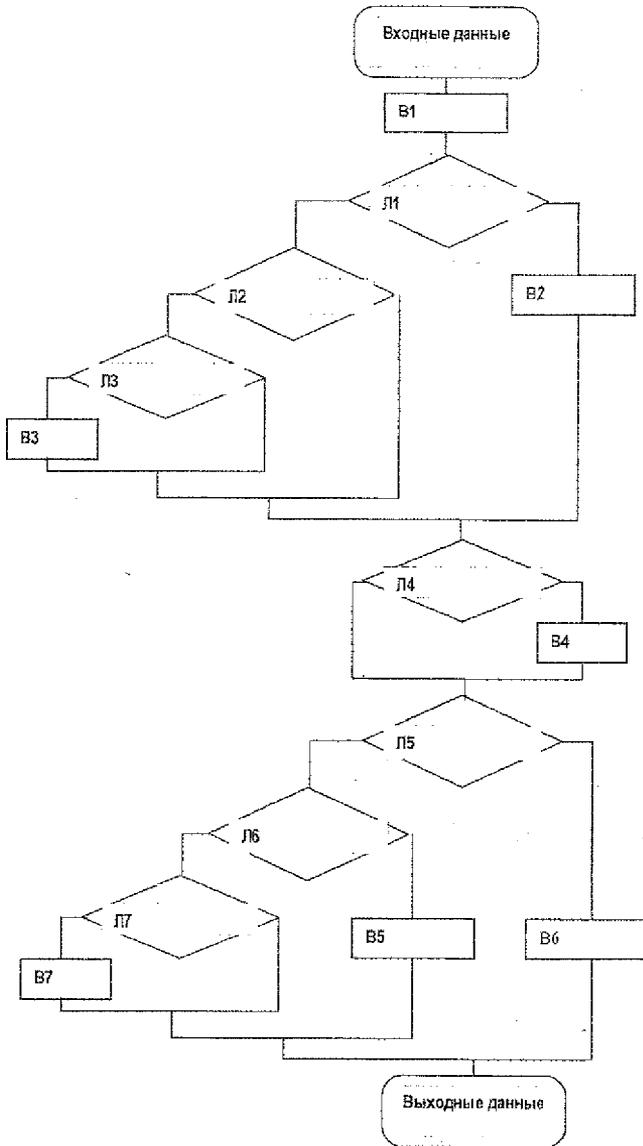


Рисунок 2.45 – Вариант D6

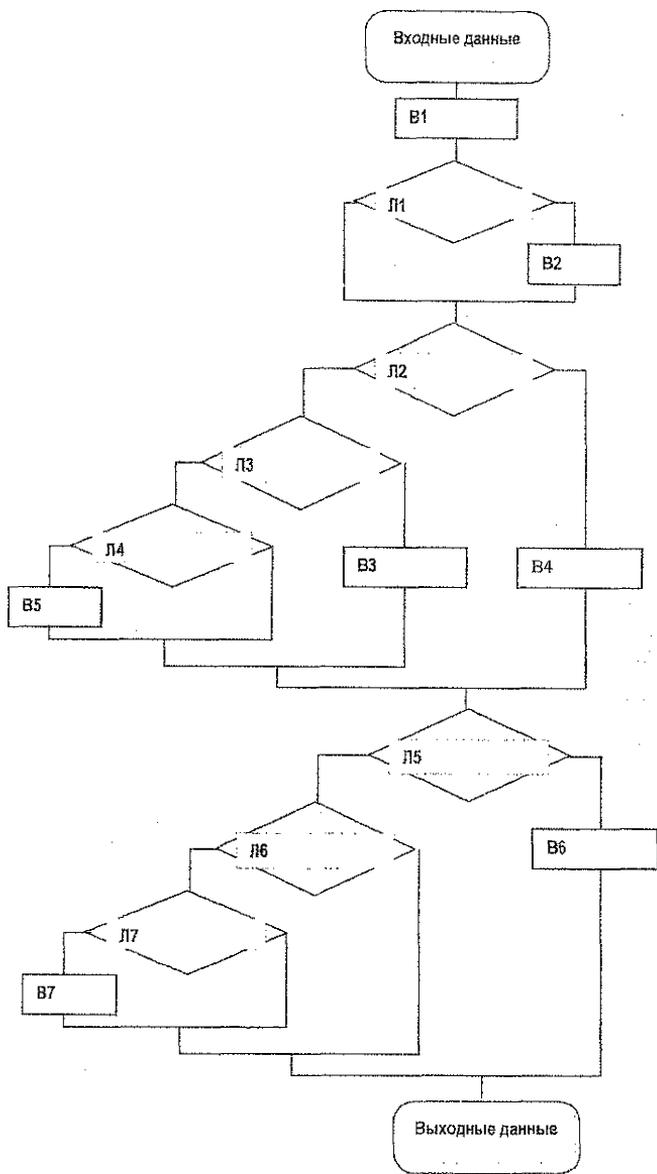


Рисунок 2.46 – Вариант D7

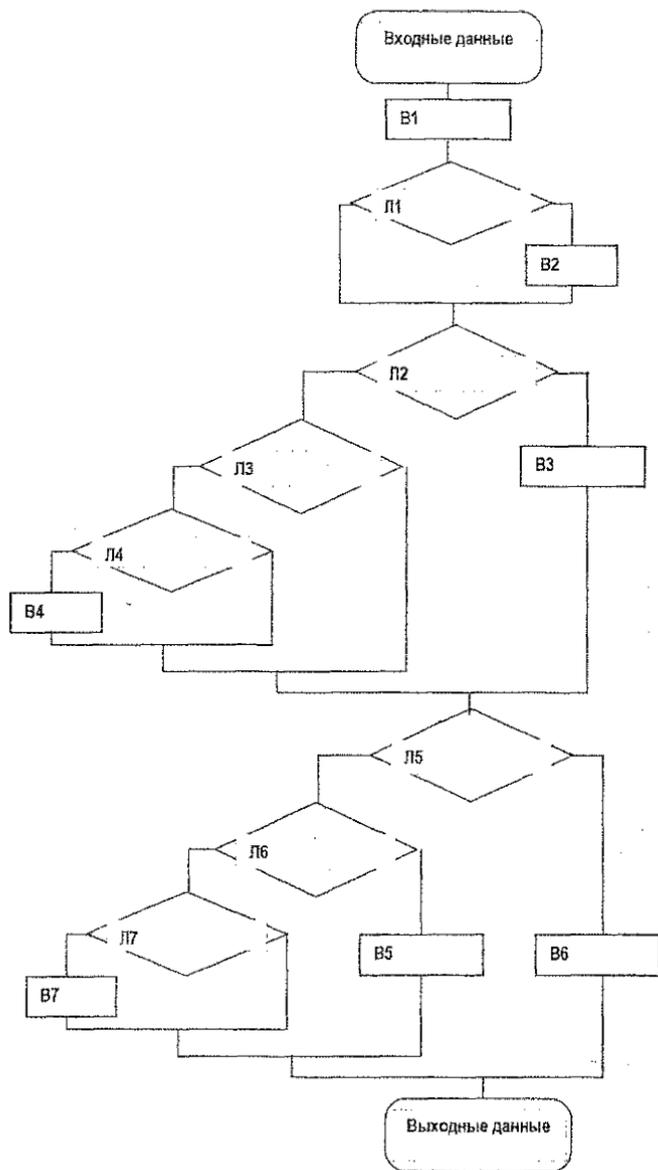


Рисунок 2.47 – Вариант D8

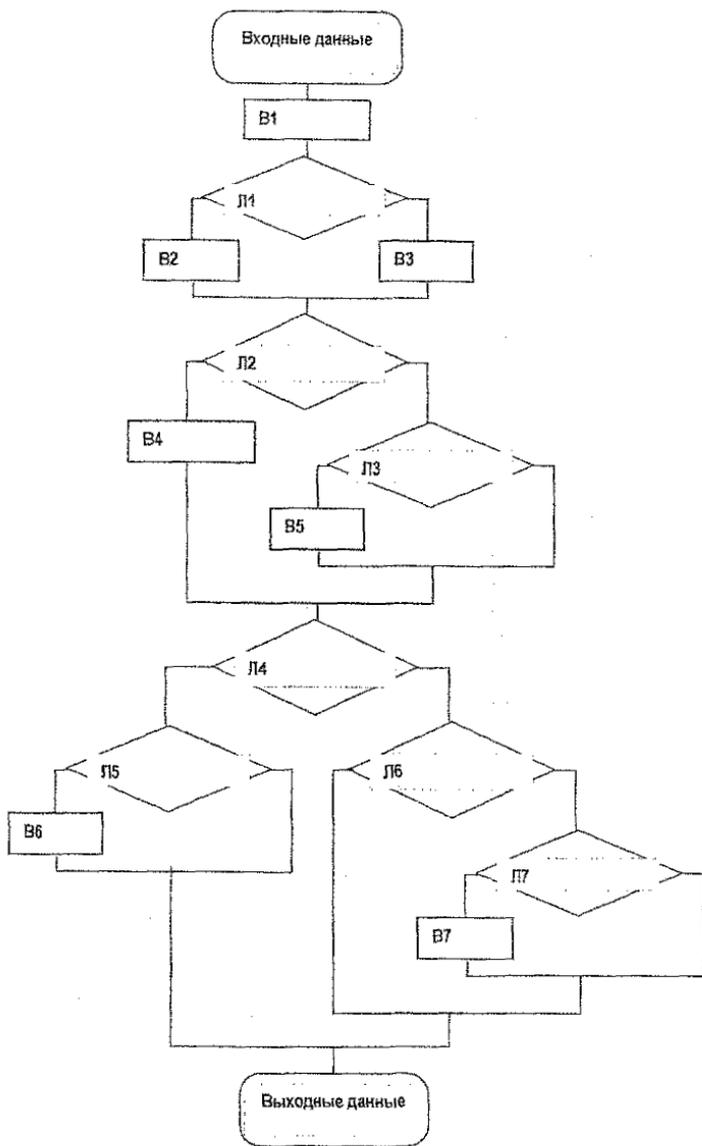


Рисунок 2.48 – Вариант D9

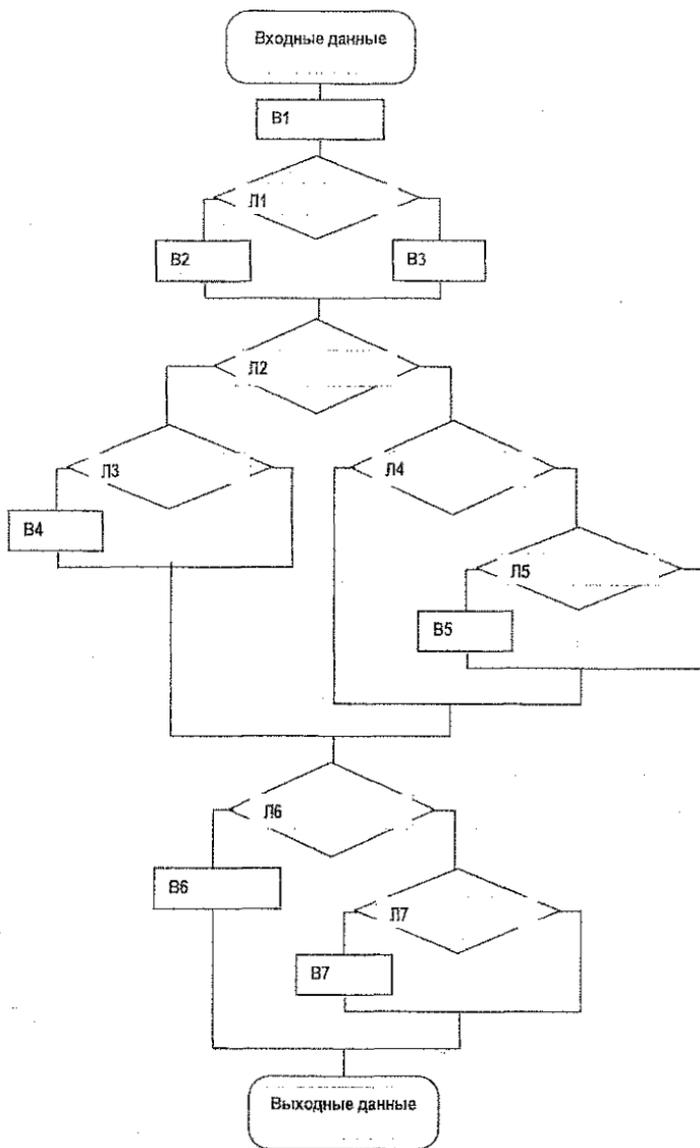


Рисунок 2.49 – Вариант D10

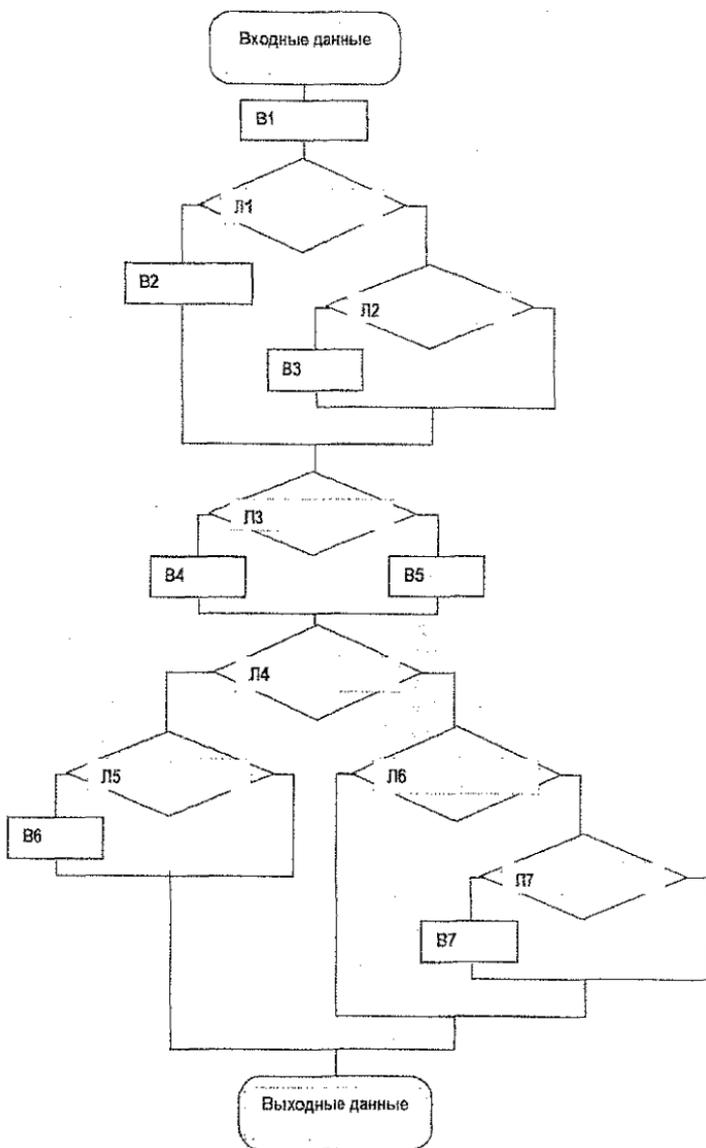


Рисунок 2.50 – Вариант D11

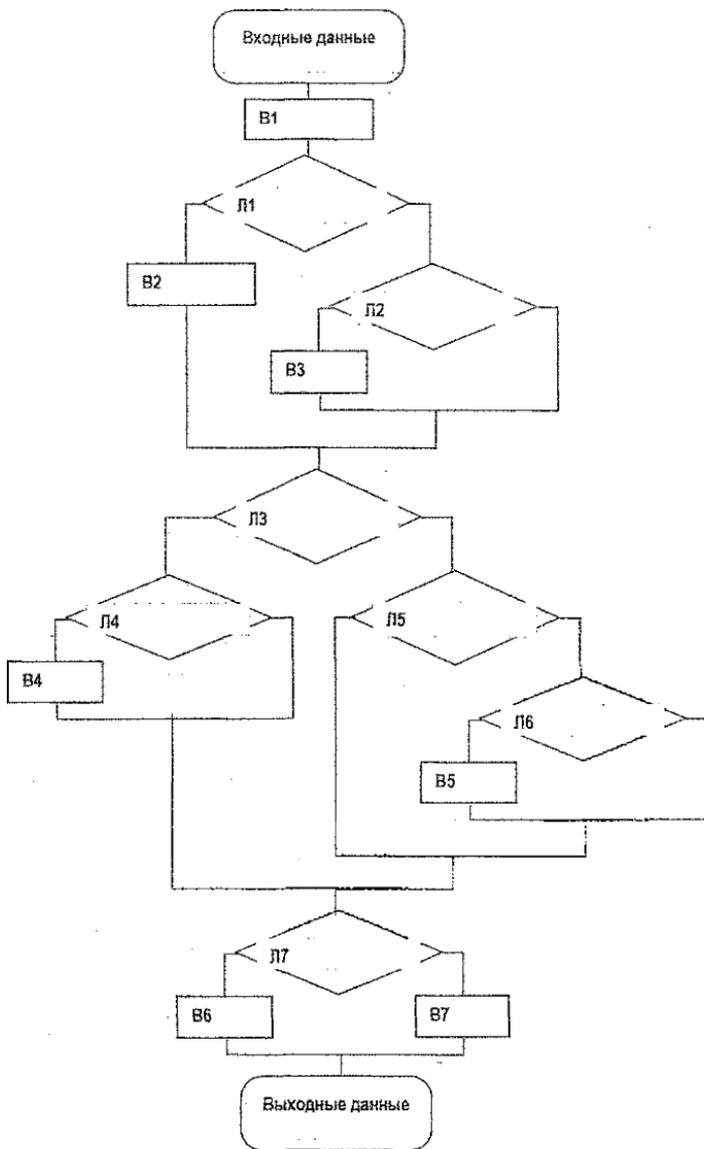


Рисунок 2.51 – Вариант D12

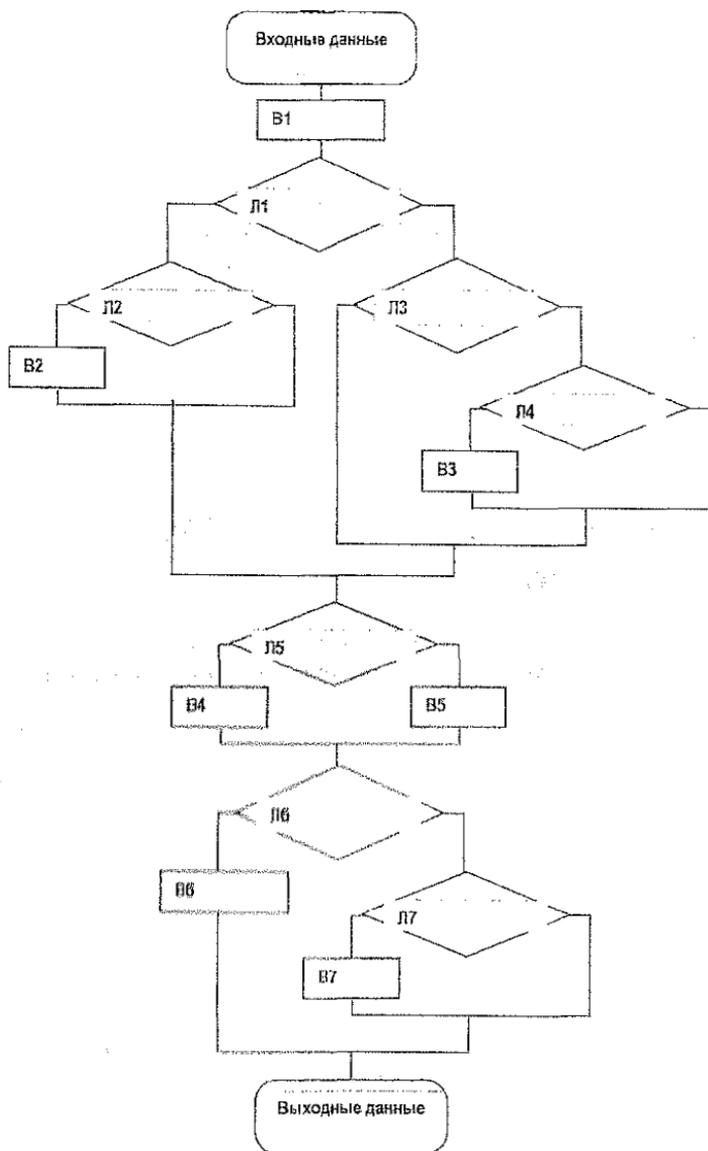


Рисунок 2.52 – Вариант D13

УЧЕБНОЕ ИЗДАНИЕ

Составители:
Хвещук Владимир Иванович
Муравьев Геннадий Леонидович

Лабораторный практикум по дисциплине
«ПРОЕКТИРОВАНИЕ СИСТЕМ ОБРАБОТКИ ДАННЫХ»
ТЕСТИРОВАНИЕ СТРУКТУРЫ МОДУЛЯ

для студентов специальности
«Автоматизированные системы обработки информации»

Ответственный за выпуск: **Хвещук В.И.**
Редактор: **Строкач Т.В.**
Компьютерная верстка: **Боровикова Е.А.**
Корректор: **Никитчик К.В.**

Подписано к печати 29.11.2009 г. Формат 60x84 1/16. Усл. печ. л. 4,0. Уч.-изд. л. 4,25.
Тираж 50 экз. Заказ № 1097. Отпечатано на ризографе учреждения образования
«Брестский государственный технический университет».
224017, г. Брест, ул. Московская, 267.