

«

»

“ ”

“ ”
_____ .

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Основы алгоритмизации и программирования

: 20.03.01

, :

: 1, : 2

		2
1	()	4
2		144
3	, .	89
4	, .	18
5	, .	18
6	, .	36
7	, .	0
8	, .	2
9	, .	15
10	, .	55
11	(, ,)	
12		

(): 20.03.01

246 21.03.2016 ., : 20.04.2016 .

: 1,

(): 20.03.01

, 17-04 20.06.2017

, 5 21.06.2017

:

,

:

,

:

. . . .

1.

1.1

Компетенция ФГОС: ОК.12 способность использования основных программных средств, умением пользоваться глобальными информационными ресурсами, владение современными средствами телекоммуникаций, способность использовать навыки работы с информацией из различных источников для решения профессиональных и социальных задач; в части следующих результатов обучения:	
5.	
7.	
8.	
Компетенция ФГОС: ПК.21 способность решать задачи профессиональной деятельности в составе научно-исследовательского коллектива; в части следующих результатов обучения:	
1.	,

2.

2.1

	(
, , ,)	

.12. 7	
1. использование навыков программирования на одном из языков программирования для решения задач	; ; ;
.12. 5	
2. знать правовые основы информационной безопасности и принципы защиты авторского права на программные продукты	;
3. уметь пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ	; ;
4. владеть навыками использования программных средств и работы в компьютерных сетях, уметь создавать базы данных и использовать ресурсы Интернет	; ;
5. уметь применять основные математические методы для решения задач экологии	;
.12. 8	
6. уметь использовать специализированные программные средства при решении профессиональных задач	; ; ;
7. уметь использовать языки и системы программирования для решения профессиональных задач	; ; ;
8. основные этапы решения задач на ПК	; ; ;

9.основные понятия, принципы построения и технологию работы с базами данных	;
10.состояние и тенденции развития информационных технологий и информатики в современном обществе	;
11.иметь опыт использования специализированных программных средств для решения профессиональных задач; быть способным осваивать новые программные продукты	;
.12. 7	
12.уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня, как средства программного моделирования изучаемых объектов и процессов	;
.21. 1	
13.иметь представление о методах решения систем линейных алгебраических уравнений	

3.

3.1

	,	.	
: 2			
:			
10.	0	4	1, 12, 2, 4, 6, 7, 8
:			
11.	0	4	2, 6
:			
13.	0	6	1, 2, 6, 8
:			
14.	0	4	12, 2, 4, 8

3.2

	,	.		
: 2				
:				
14.	0	4	10, 3, 6, 9	MS Excel
:				
27.	0	4	12, 4, 6	

:					
21.		0	8	1, 10, 12, 8	<p>1. FORTRAN. - -</p> <p>2.</p> <p>3.</p>
25.		0	8	10, 12, 13, 5, 6, 7	<p>1. FORTRAN. . .</p> <p>2.</p> <p>().</p> <p>3.</p>
:					
24.		0	6	10, 11, 12, 5, 6, 7	<p>1. FORTRAN. 'data'. .</p> <p>2.</p> <p>().</p> <p>3.</p>

22.	0	6	12, 6, 7	<p>1.</p> <p style="text-align: right;">FORTRAN.</p> <p>2.</p> <p>3.</p>
-----	---	---	----------	--------------------------------------------------------------------------------------------------

4.

: 2				
1		11, 5	10	6
<p>15-16 : «</p> <p>» []: - /</p> <p>; , [2016]. - :</p> <p>http://elibrary.nstu.ru/source?bib_id=vtls000232896. -</p>				
2		11, 2	20	4
<p>: , : 280200 -</p> <p>/ - ; [.]. - , 2007. - 46, [1]</p> <p>.: ., , .. - : http://elibrary.nstu.ru/source?bib_id=vtls000073310</p>				
3		1, 11, 2, 5, 8, 9	15	2
<p>: , 2015. - 81, [1] .: .. -</p> <p>: http://elibrary.nstu.ru/source?bib_id=vtls000221460</p>				
4		1, 10, 11, 12, 2, 3, 4, 5, 6, 7, 8, 9	10	3
<p>: «</p> <p>» []: - / ,</p> <p>; - , [2016]. - :</p> <p>http://elibrary.nstu.ru/source?bib_id=vtls000232896. - :</p> <p>280200 - / - ; [.]. - , 2007. - 46, [1] .: ., , .. - :</p> <p>http://elibrary.nstu.ru/source?bib_id=vtls000073310</p>				

5.

(. 5.1).

5.1

	-
	e-mail: golubeva@corp.nstu.ru

6.

(),

-
15-

ECTS.

. 6.1.

6.1

	.	
: 2		
Лекция:	5	10
Лабораторная:	20	40
" : 280200 - / : http://elibrary.nstu.ru/source?bib_id=vtls000073310"		
Практические занятия:	5	10
РГЗ:	9	20
" : : http://elibrary.nstu.ru/source?bib_id=vtls000232896. -		
Зачет:	10	20

6.2

6.2

		/		
.12	5.	+		+
	7.	+	+	+
	8.	+	+	+
.21	1.	+	+	

7.

1. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : учебное пособие для вузов / В. Г. Олифер, Н. А. Олифер. - СПб. [и др.], 2007. - 957 с. : ил.
2. Информатика. Базовый курс : [учебное пособие для втузов] / под ред. С. В. Симоновича. - СПб. [и др.], 2010. - 639 с. : ил.
3. Brainerd W. S. Guide to Fortran 2003 Programming [electronic resource] // by Walter S. Brainerd. - London :, 2009. : v.: digital // Springer eBooks. - Режим доступа: <http://dx.doi.org/10.1007/978-1-84882-543-7>

1. Гордеев А. В. Операционные системы : учебник для вузов / А. В. Гордеев. - СПб., 2007. - 415 с. : ил. - На тит. л.: Издательская программа "300 лучших учебников для высшей школы в честь 300-летия Санкт-Петербурга".
2. Бартенев О. В. Современный Фортран / О. В. Бартенев. - М., 2000. - 448 с.
3. Акулов О. А. Информатика: базовый курс : учебник [для студентов вузов, бакалавров, магистров, обучающихся по направлению "Информатика и вычислительная техника"] / О. А. Акулов, Н. В. Медведев. - Москва, 2008. - 574 с. : ил., табл.

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znaniy.com" : <http://znaniy.com/>
5. :

8.

8.1

1. Хиценко В. П. Основы программирования : учебное пособие / В. П. Хиценко ; Новосиб. гос. техн. ун-т. - Новосибирск, 2015. - 81, [1] с. : ил. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000221460
2. Информатика : лабораторные работы для ФЛА направления 280200 - Защита окружающей среды / Новосиб. гос. техн. ун-т ; [сост. Т. А. Коротаева]. - Новосибирск, 2007. - 46, [1] с. : ил., схемы, табл. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000073310
3. Турчинович А. О. Методические указания к выполнению расчетно-графического задания по дисциплине «Основы алгоритмизации и программирования» [Электронный ресурс] : учебно-методическое пособие / А. О. Турчинович, Д. А. Немущенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2016]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000232896. - Загл. с экрана.

8.2

- 1 MathCAD 14
- 2 Intel Visual Fortran Compiler
- 3 Office

9. -

1	(- , ,)	

1	(Internet)	

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра инженерных проблем экологии

“УТВЕРЖДАЮ”
ДЕКАН ФЛА
д.т.н., профессор С.Д. Саленко
“ ____ ” _____ ____ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования

Образовательная программа: 20.03.01 Техносферная безопасность,
профиль: Инженерная защита окружающей среды

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине «Основы алгоритмизации и программирования» приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (лабораторные работы РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.9 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	у2. уметь пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ	MS Excel. Создание и форматирование таблиц. Выполнение инженерных расчетов Инструментальные системы Основы программирования Пакеты прикладных программ по отраслям знаний Построение алгоритмов Программирование ветвящихся вычислительных процессов Программирование циклических вычислительных процессов Численные методы решения систем линейных алгебраических уравнений Этапы подготовки и решения задач на ЭВМ	Отчет по лабораторной работе №1, РГЗ, задача 1.	-
ОПК.9	у6. уметь использовать специализированные программные средства при решении профессиональных задач	MS Excel. Создание и форматирование таблиц. Выполнение инженерных расчетов Инструментальные системы Метод дихотомии Методы численного интегрирования Основы программирования Пакеты прикладных программ по отраслям знаний Построение алгоритмов Программирование ветвящихся вычислительных процессов Программирование циклических вычислительных процессов Сортировка и анализ списков. Фильтрация списков. Применение расширенного фильтра для сортировки Численные методы решения систем линейных алгебраических уравнений Этапы подготовки и решения задач на ЭВМ	Отчет по лабораторной работе №2-4, РГЗ задачи 2-3	Зачет, вопросы 1-9
ОПК.9	у8. уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования	Метод дихотомии Методы численного интегрирования Построение алгоритмов Программирование ветвящихся вычислительных процессов Численные методы решения систем линейных алгебраических уравнений	Отчет по лабораторной работе РГЗ, задачи 5	Зачет, вопросы 10-26

	изучаемых объектов и процессов			
ОПК.9	у10. быть способным осваивать новые программные продукты	Метод дихотомии Основы программирования Пакеты прикладных программ по отраслям знаний Этапы подготовки и решения задач на ЭВМ	РГЗ, задачи 1-5	Зачет, вопросы 26-28
ПК.21/НИ способность решать задачи профессиональной деятельности в составе научно-исследовательского коллектива	з1. иметь представление об общих принципах математического моделирования задач экологии, методах решения систем линейных алгебраических и дифференциальных уравнений	Численные методы решения систем линейных алгебраических уравнений	Отчет по лабораторной работе 4, РГЗ, задачи 1-3	-

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 2 семестре - в форме дифференцированного зачета, который направлен на оценку сформированности компетенций ОПК.9, ПК.21/НИ.

Зачет проводится в форме письменного тестирования, варианты теста составляются из вопросов, приведенных в паспорте зачета, позволяющих оценить показатели сформированности соответствующих компетенций

Кроме того, сформированность компетенции проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 2 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (РГЗ). Требования к выполнению РГЗ, состав и правила оценки сформулированы в паспорте РГЗ.

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.9, ПК.21/НИ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения

учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Правила аттестации студентов по учебной дисциплине

1. Итоговая оценка по дисциплине складывается по результатам работы в семестре (посещение лекционных, практических занятий, выполнение и защиты лабораторных работ) и сдачи РГР и зачета.

2. Оценка учебной деятельности студента в семестре.

2.1 Посещение лекционных, практических и лабораторных занятий студентом в семестре обязательно и оценивается в *1 балл* за каждое занятие.

20 баллов – 100 % посещение

10 баллов – 50% посещение

2.2 Работа студента на лабораторных работах в семестре оценивается *от 0 до 40 баллов*. Баллы начисляются за качественный подход к решению задач и оформлению отчета.

2.3 РГР студента оценивается *от 0 до 20 баллов*. Баллы начисляются за качественный подход к задаче, аккуратность и исполнительность, творческие решения при выполнении РГЗ, хорошее оформление работы.

2.4 Количество баллов, набранное студентом в течение семестра, рассчитывается как сумма баллов за все виды его учебной деятельности. Студенты, набравшие не менее *40 баллов* в течение семестра, допускаются до зачета.

3. На зачете студент может набрать *от 10 до 20 баллов*. В случае если студент набирает менее 5 баллов, выставляется оценка "неудовлетворительно" и студент направляется на пересдачу.

4. Количество баллов, набранное студентом по итогам изучения дисциплины, рассчитывается как сумма баллов за все виды его учебной деятельности и баллы, набранные на зачете. По результатам учебной деятельности в семестре и зачета в зачетную книжку и ведомость выставляется оценка по дисциплине:

- | | |
|-----------------------|-------------------|
| - "отлично" | - 87 -100 баллов; |
| - "хорошо" | - 73-86 баллов; |
| - "удовлетворительно" | - 50-72 баллов. |

Паспорт зачета

по дисциплине «Основы алгоритмизации и программирования», 2 семестр

1. Методика оценки

Зачет проводится в письменной форме, по тестам. Билет формируется по следующему правилу: список вопросов приведен ниже. В ходе зачета преподаватель вправе задавать студенту дополнительные вопросы из общего перечня.

Пример теста для зачета

Вопрос № 1. Характеристика языка ФОРТРАН

- Ответ 1: ФОРТРАН- язык программирования низкого уровня
- Ответ 2: ФОРТРАН- язык программирования высокого уровня
- Ответ3: ФОРТРАН- язык программирования для создания реляционных баз данных

Вопрос № 2. Символы языка ФОРТРАН

- Ответ 1: Алфавит языка ФОРТРАН состоит из букв латинского и греческого алфавита+ арабские цифры
- Ответ 2: Алфавит языка ФОРТРАН состоит из букв латинского алфавита+ арабские цифры+ специальные символы
- Ответ3: Алфавит языка ФОРТРАН состоит из букв русского алфавита+ арабские и римские цифры

Вопрос № 3. Что из ниже перечисленного не является типом ФОРТРАНа?

- Ответ 1: целый
- Ответ 2: комплексный
- Ответ3: иррациональный
- Ответ4: вещественный
- Ответ5: натуральный

Вопрос № 4. Как выполняется операция деления, когда делимое и делитель имеют различные типы (целый и вещественный)? Правильных ответов более, чем один.

- Ответ 1: $\text{целое} / \text{целое} = \text{целое}$
- Ответ 2: $\text{целое} / \text{вещественное} = \text{целое}$
- Ответ3: $\text{вещественное} / \text{вещественное} = \text{вещественное}$

Вопрос № 5. Определение массива:

- Ответ 1: Массив - это именованный набор конечного числа объектов одного типа
- Ответ 2: Массив - это именованный набор конечного числа объектов различных типов

Вопрос № 6. Что называется рангом массива?:

- Ответ 1: Число измерений массива
- Ответ 2: Общее число элементов массива

- Ответ 3: Максимальное число элементов массива по всем измерениям массива

Вопрос № 7. Что называется размером массива?

- Ответ 1: Число измерений массива
- Ответ 2: Общее число элементов массива
- Ответ 3: Максимальное число элементов массива по всем измерениям массива

Вопрос № 8. Выберите правильные варианты задания вещественного массива a, состоящего из 5 элементов:

- Ответ 1: `real a_CRLF_dimension a(5)`
- Ответ 2: `real, dimension (a) ::5`
- Ответ 3: `parameter(n=5)_CRLF_dimension a(n)`
- Ответ 4: `real, dimension (5) ::a`
- Ответ 5: `dimension a(1:5)`

Вопрос № 9. Каким образом располагаются в памяти элементы двумерного массива?

- Ответ 1: по столбцам
- Ответ 2: по строкам

Вопрос № 10. Определите, какие из следующих выражений верны, а какие нет

- Ответ 1: `IF(A>B)C=A-B`
`END IF`
- Ответ 2: `IF(J<D) STOP 'J<D'`
- Ответ 3: `WILE(X<Y)DO`
`X=X**2`
`END DO`

2. Критерии оценки

- Ответ на тест для зачета считается **неудовлетворительным**, если студент при ответе на вопросы не владеет основными понятиями и определениями, не способен определить структуру программы, при решении задач допускает принципиальные ошибки, правильных ответов менее 50%, оценка составляет *0-9 баллов*.
- Ответ на тест для зачета засчитывается на **пороговом** уровне, если студент при ответе на вопросы владеет основными понятиями, может разобраться в структуре программы, правильных ответов 50-72%, оценка составляет *10-14 баллов*.
- Ответ на тест для зачета билет засчитывается на **базовом** уровне, если студент при ответе на вопросы владеет основными понятиями, законами, может разобраться в структуре программы и обнаружить ошибку, не допускает ошибок при решении задачи, правильных ответов 73-86%, оценка составляет *15-17 баллов*.
- Ответ на тест для зачета билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы способен осуществить сравнительный анализ подходов и методов, способен составить алгоритм и код программы, способен обосновать выбор метода решения задачи, правильных ответов 87-100%, оценка составляет *18-20 баллов*.

3. Шкала оценки

Зачет считается сданным, если сумма баллов по всем вопросам оставляет не менее 10 баллов (из 20 возможных).

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. **Вопросы к зачету по дисциплине «Основы алгоритмизации и программирования»**

1. Характеристика языка ФОРТРАН
2. Символы языка ФОРТРАН
3. Структура программы на Фортране.
4. Свободный и фиксированный форматы записи программы.
5. Консоль-проект.
6. Структура головной программы.
7. Имена.
8. Скаляры и массивы.
9. Стандартные числовые типы данных.
10. Константы и переменные. Правила умолчания о типах данных.
11. Изменение правил умолчания.
12. Арифметические операции.
13. Арифметические выражения.
14. Целочисленное деление.
15. Ранг и типы арифметических операндов.
16. Операция отношения.
17. Логические операции.
18. Приоритет выполнения операций.
19. Операторы ввода-вывода.
20. Блок операторов и конструкций
21. Ветвление.
22. Цикл. Цикл "с параметром".
23. Цикл. Циклы "пока" и "до".
24. Использование функций.
25. Использование подпрограмм.
26. Метод половинного деления.
27. Этапы проектирования программ.
28. Правила записи исходного кода

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра инженерных проблем экологии

**Паспорт
расчетно-графического задания**

по дисциплине «Основы алгоритмизации и программирования», 2 семестр

1. Методика оценки

В рамках расчетно-графического задания по дисциплине студенты должны составить алгоритмы и написать программы для решения задач, приведенных ниже.

При выполнении расчетно-графического задания студенты должны провести анализ задачи и определить метод решения, составить подпрограммы.

Обязательные структурные части РГЗ:

1. Титульный лист
2. Содержание
3. Решение задач
 - 3.1 Условие задачи
 - 3.2 Метод решения
 - 3.3 Решение (код программы)
 - 3.4 Ответ
4. Вывод

Оцениваемые позиции: оформление работы, структурирование работы, решение каждой задачи.

5. Критерии оценки

- Работа считается **не выполненной**, если выполнены не все части РГЗ, отсутствует анализ задач и программы оценка составляет 0-8 баллов.
- Работа считается выполненной **на пороговом** уровне, если части РГЗ выполнены формально: присутствует задание и решение, но студент не отвечает на 50% вопросов, оценка составляет 9-12 баллов.
- Работа считается выполненной **на базовом** уровне, если все задания РГР выполнены верно, в работе присутствует описание решения, студент ответил на 75 % вопросов, оценка составляет 13-17 баллов.
- Работа считается выполненной **на продвинутом** уровне, если все задания РГР выполнены верно, в работе присутствует описание решения, работа структурирована студент ответил на все вопросы по работе, оценка составляет 18-20 баллов.

6. Шкала оценки

В общей оценке по дисциплине баллы за РГЗ учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

Примерный перечень задач РГЗ

Задание №1. ТАБУЛИРОВАНИЕ ФУНКЦИЙ

Цель работы: При табулировании и построении графиков функций получить начальные навыки программирования и работы на компьютере.

Задание: Получить таблицы значений шести функций в приложениях: Microsoft Excel, Mathcad и Fortran. По полученным данным построить графики функций. По таблицам

значений, полученных в Fortran, построить графики функций в Excel.

Выражение — это формула для получения значения. Выражения образуются из операндов и знаков операций, объединяемых по правилам синтаксиса алгоритмического языка. В качестве операндов выражения используются буквальные и именованные константы, переменные, указатели функций. В зависимости от типа возвращаемого результата выражения подразделяются на арифметические, логические, символьные и производного типа. В первой лабораторной работе рассмотрим правила составления арифметических выражений.

При записи арифметических выражений в Фортране используются следующие арифметические операции и функции (в порядке возрастания приоритета):

– , + — вычитание и сложение;

*, / — умножение и деление;

** — возведение в степень;

COS(x) — cos(x) SIN(x) — sin(x)

TAN(x) — tg(x) ATAN(x) — arctg(x)

ALOG(x) — ln(x) ALOG10(x) — lg(x)

EXP(x) — exp ABS(x) — |x|

SQRT(x) — кореньх INT(x) — целая часть x.

ASIN(x) — arcsin(x) ACOS(x) — arcos(x)

SEC(x) — sec(x)

Порядок выполнения операций одного и того же приоритета — слева направо (кроме возведения в степень). Для изменения вышеупомянутого порядка вычислений используются круглые скобки.

Варианты заданий приведены в таблице 1.

Таблица 1. Варианты заданий

№ п/п	1	2	3	4	5	6
1	$y = 2 \sin(x/3)$	$y = x+1 - x-1 $	$y = 2 + \sqrt{4-x^2-6x}$	$y = -2 + \frac{2}{3}\sqrt{x^2-9}$	$x = \arcsin(t)$ $y = \arctg(t)$	$\rho = \varphi$
2	$y = 2 \sin(3x + \pi)$	$y = 2 x - x + 1$	$y = -1 + \sqrt{2-x^2-5x}$	$y = \frac{x+1}{x+2}$	$x = \cos(t)$ $y = \sin^2(t)$	$\rho = \frac{\pi}{\varphi}$
3	$y = \tg(2x+1)$	$y = x - x $	$y = 2 - \sqrt{4-x^2-6x}$	$y = -3 \cdot \sqrt{(x+1)^2+1}$	$x = \cos(2t)$ $y = \cos^2(t)$	$\rho = e^{\pi/\varphi}$
4	$y = 2 \cos(3x+1)$	$y = \ln(\sqrt{x-4})$	$y = \frac{1}{2} - \sqrt{x^2-2y-\frac{1}{4}}$	$y = 1 + \frac{7}{3}\sqrt{3+x^2+4x}$	$x = 1$ $y = \sin(t)$	$\rho = 2\varphi$
5	$y = -\sin(\frac{x}{2}-1)$	$y = \sqrt{x^3}$	$y = -1 - \sqrt{2-x^2-5x}$	$y = 3 - \frac{4}{3}\sqrt{x^2+9}$	$x = \sqrt{\cos(t)}$ $y = \sqrt{\sin(t)}$	$\rho = \sec^2\left(\frac{\varphi}{2}\right)$
6	$y = 2 \tg(2x) $	$y = \ln(x^2)$	$y = 1 + \frac{3}{4}\sqrt{5-x^2-2x}$	$y = 1 - \frac{4}{3}\sqrt{1+x^2+4x}$	$x = \sin(t)$ $y = \ln(\sin(t))$	$\rho^2 \cos(2\varphi) = 1$
7	$y = \frac{1}{2}\sin 2x $	$y = 2 \cdot \ln(x)$	$y = 2 - \frac{5}{2}\sqrt{8-x^2+x}$	$y = 1 + \frac{2}{3}\sqrt{x^2-x+5}$	$x = \arcsin(t)$ $y = t^2$	$\rho = \frac{2}{1-\cos(\varphi)}$
8	$y = \tg\left \frac{x}{2}\right $	$y = x -1$	$y = 1 - \frac{2}{3}\sqrt{5-x^2-x}$	$y = 2 + \frac{3}{4}\sqrt{x^2-x-6}$	$x = 2^t$ $y = 2^{2t}$	$\rho = \tg\left(\frac{\varphi}{2}\right)$

№ п/п	1	2	3	4	5	6
9	$y = 2 \sin(\frac{\pi}{3} - x)$	$y = x + x $	$y = 1 + \frac{1}{7} \sqrt{8 - x^2} - x$	$y = 2 + \frac{2}{5} \sqrt{x^2 + 25}$	$x = 3^t$ $y = 9^t - 3^t$	$\rho = \operatorname{cosec}^2(\frac{\varphi}{2})$
10	$y = \cos(3x) $	$y = 2 x - 3 + 5x$	$y = 3 + \frac{3}{5} \sqrt{9 - x^2} + x$	$y = 2 - \frac{3}{4} \sqrt{x^2 - x - 3}$	$x = (t + 1/t)/2$ $y = (t - 1/t)/2$	$\rho = \varphi^2$
11	$y = \operatorname{tg}(\frac{x}{2} - 1)$	$y = -\frac{3}{2}x + 1$	$y = 1 + \frac{1}{2} \sqrt{5 - 3x^2} - x$	$y = 7 - \frac{3}{2} \sqrt{x^2 - 6x + 8}$	$x = 5^{t+1}$ $y = 5^{2t+3}$	$\rho = \sin(\frac{\varphi}{2})$
12	$y = \frac{3}{2} \cos(\frac{x}{2} + 1)$	$y = 2 x + 3$	$y = -1 + \frac{1}{3} \sqrt{x^2 + 3x}$	$y = 1 + \frac{3}{4} \sqrt{x^2 + 4x + 9}$	$x = 2^{\sin(t)}$ $y = \sin(t)$	$\rho^2 \cos(2\varphi) = 9$
13	$y = \frac{2}{3} \sin(3x + 1)$	$y = \frac{1}{7}(x + 1) - 2$	$y = 2 + \frac{2}{3} \sqrt{4 - x^2} + x$	$y = 9 - 2 \sqrt{x^2 + 4x + 8}$	$x = \sqrt{t}$ $y = 1/t$	$\rho = \frac{1}{2 + \cos(\varphi)}$
14	$y = \operatorname{tg} 2x $	$y = \sqrt{x^2 - 4x + 4}$	$y = 3 - \frac{3}{5} \sqrt{9 + 5x^2} + x$	$y = 1 - \frac{3}{4} \sqrt{x^2 + 4x + 1}$	$x = \cos(t)$ $y = 1/\sin(t)$	$\rho = \frac{2}{\cos(\varphi)}$
15	$y = \operatorname{tg}(1 - x)$	$y = \sqrt{x^2 - 6x + 9}$	$y = 1 - \frac{1}{2} \sqrt{6 - x^2} - 2x$	$y = 5 - \frac{4}{3} \sqrt{x^2 + 4x - 2}$	$x = \sec(t)$ $y = \cos(t)$	$\rho = \frac{2}{1 + \cos(\varphi)}$
16	$y = -\frac{2}{7} \sin(x + 1)$	$y = \frac{1}{5}(x - 1) + 1$	$y = 1 - \frac{1}{3} \sqrt{-5x^2 + 3x}$	$y = 2 + \frac{4}{3} \sqrt{x^2 + 2x - 1}$	$x = \cos(t)$ $y = \cos(2t)$	$\rho = \frac{8}{3 - \cos(\varphi)}$
17	$y = -3 \sin(5x)$	$y = \frac{1}{5}x - 1$	$y = 2 - \frac{2}{3} \sqrt{9 - x^2} + x$	$y = \frac{1}{x} - \frac{1}{2}$	$x = \operatorname{tg}(t)$ $y = \frac{2}{1 + \cos(t)}$	$\rho = 1 + \cos(\varphi)$
18	$y = \operatorname{ctg}(3x) $	$y = \sqrt{(x - 1)^2 + 3}$	$y = 1 + \frac{7}{3} \sqrt{x^2 - 4x + 5}$	$y = 2 - \frac{4}{3} \sqrt{x^2 + 2x - 8}$	$x = \sin(\frac{\pi}{2} + t)$ $y = \sin(\pi + t)$	$\rho^2 = 36 \cos(\varphi)$
19	$y = 1 - \cos(3x)$	$y = 1 + \sqrt{x^2 - x + 1}$	$y = 2 - \frac{3}{7} \sqrt{x^2 + 2x + 4}$	$y = \frac{2x}{3 - x}$	$x = \cos(t)$ $y = \pi \cdot \sin(t)$	$\rho = 1 - \cos(\varphi)$
20	$y = \frac{1}{3} \cos(2x) $	$y = 1 + \sqrt{x^2 - x + 4}$	$y = 1 + \frac{4}{3} \sqrt{-6x - x^2}$	$x = 1 + \frac{3}{2} \sqrt{y^2 + 4y + 8}$	$x = 1 + \operatorname{tg}(t)$ $y = \operatorname{ctg}^2(t)$	$\rho = \cos(2\varphi)$
21	$y = \operatorname{ctg} 3x - 3 $	$y = x - x + 2 $	$y = 1 - \frac{7}{3} \sqrt{x^2 - 4x + 5}$	$y = \frac{-x + 4}{x - 1}$	$x = \operatorname{cosec}(t)$ $y = \operatorname{tg}^2(t)$	$\rho = \sin^3 \varphi$
22	$y = \operatorname{tg}(x) + 2$	$y = 3 - \sqrt{x^2 - x + 1}$	$y = 7 + \frac{2}{5} \sqrt{6 - 6x - x^2}$	$y = 1 - \frac{3}{2} \sqrt{x^2 + 4x + 8}$	$x = \operatorname{cosec}(t)$ $y = \sin(t)$	$\rho = \sin^2(\varphi)$
23	$y = \sin(x) - 1 $	$y = \frac{3}{2}x + \frac{1}{2}$	$y = 2 + \frac{3}{7} \sqrt{x^2 + 2x + 8}$	$y = \frac{x - 1}{2x - 5}$	$x = \sec(t)$ $y = \ln(\cos(t))$	$\rho = 4 \sin(\varphi)$
24	$y = \frac{1}{3} \sin \frac{ x }{2}$	$y = \ln x $	$y = 2 - \sqrt{-5 - 6x - x^2}$	$y = 2 + \frac{2}{3} \sqrt{x^2 - x + 9}$	$x = \cos(t)$ $y = \lg(\sec(t))$	$\rho = \cos(\varphi) + \varphi$

Рассмотрим пример.

Дана функция $y = 0.5 \sin(3x + 2)$. Областью определения является вся числовая ось от $-\infty$ до $+\infty$. Но вследствие периодичности функции синус достаточно построить график в интервале $-\frac{2}{3} \leq x \leq \frac{2(\pi - 1)}{3}$. Областью изменения функции является интервал $-0.5 \leq y \leq 0.5$.

Каждая пара значений x и y должна быть записана в таблицу результатов счета. Запишем алгоритм решения первой задачи.

1. Зададим тип и значение неизменяемому параметру π .
2. Опишем тип используемых в программе переменных.
3. Откроем файл для записи значений x и y .
4. Определим начальное a и конечное b значения переменной x .
5. Введем количество строк n для табулирования функции.
6. Вычислим шаг h цикла для изменения переменной x .
7. Организуем цикл с управляющей переменной i целого типа.

8. В цикле вычислим значения переменных x и y .
9. Напечатаем на дисплей значения x и y .
10. Повторим печать значений x и y в файл.
11. Закончим цикл.

В соответствии с алгоритмом составим Фортран-программу:

```
PROGRAM LAB1
REAL, PARAMETER :: PI=3.141593
REAL A, B, H, X, Y
INTEGER I, N
OPEN(1, FILE='LAB1.TXT')
A=-2./3.; B=2.*(PI-1.)/3.
PRINT *, ' N=?'; READ *, N
H=(B-A)/N
DO I=0,N
X=A+H*I
Y=0.5*SIN(3.*X+2.)
PRINT 2, X,Y; WRITE(1, 2) X,Y
END DO
2 FORMAT (' X=',F8.4,3X,'Y=',F8.4)
END PROGRAM LAB1
```

Прокомментируем работу программы:

- вначале после ключевого слова PROGRAM объявляется имя программы, которое обязательно должно начинаться с буквы латинского алфавита или символа \$ (в общем случае ключевое слово PROGRAM с именем программы может отсутствовать);
- первый оператор задает тип и значение именованной константе PI;
- затем идут два оператора, описывающие тип используемых в программе переменных;
- оператор OPEN назначает логический номер 1 файлу «LAB1.TXT». Указанный файл создается в папке проекта либо предварительно, либо в процессе работы программы в момент выполнения оператора OPEN;
- в следующей строке через точку с запятой записаны два оператора, которые вычисляют значения переменных A и B (обратите внимание на десятичные точки в неименованных константах вещественного типа);
- оператор PRINT выводит на дисплей в свободном формате символьную константу, которая заключена в апострофы. Эта последовательность символов является подсказкой для пользователя;
- оператор READ требует ввода с клавиатуры значения целой переменной N. После нажатия клавиши [Enter] переменной N будет присвоено набранное значение;
- следующий оператор вычисляет значение переменной H;
- оператор цикла DO организует повторение вычислений в теле цикла необходимое число раз при изменении управляющей переменной I от 0 до N с шагом 1 (шаг равен 1 по умолчанию, т.к. его значение не указано);
- операторы в следующих двух строках обеспечивают вычисление арифметических выражений, записанных справа от знака равенства. Результаты присваиваются переменным X и Y;
- оператор PRINT работает под управлением оператора FORMAT с меткой 2. Он выводит на экран сначала символьную константу X=, которая в программе заключена в апострофы, затем числовое значение переменной X по дескриптору F8.4. Далее через три позиции выводится символьная константа Y=, и значение Y также по дескриптору F8.4, где 8 – общее число позиций для выводимого числа, а 4 – количество позиций для цифр после десятичной точки. В общее число позиций включается знак отрицательного числа «-» и десятичная точка (ноль перед десятичной точкой, как правило, не печатается);

- оператор WRITE повторяет печать значений X и Y в файл «LAB1.TXT» с логическим номером 1 под управлением оператора FORMAT с меткой 2;
- предпоследний оператор END DO ограничивает тело цикла;
- последний оператор END PROGRAM LAB1 ограничивает программный модуль, заканчивая его (LAB1 или PROGRAM LAB1 могут быть опущены). Если имя программы не объявлено, то оно не может присутствовать и в операторе END.

Задание №2. ПРОГРАММИРОВАНИЕ ВЕТВЯЩИХСЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

Цель работы: Получить навыки программирования ветвящихся вычислительных процессов. Изучить условные операторы, переменные логического типа, логические отношения и операции.

Задание: Пусть на плоскости x, y задана область D . Требуется определить «попадает» ли точка с координатами (x, y) в область D или нет. Если попадает, то необходимо вычислить функцию f_1 и переменной M, значение которой является признаком «попадания» или «непопадания», присвоить значение $M=1$. В противном случае вычислить f_2 и положить $M=2$.

В каждом варианте задания область D можно разбить на две подобласти: D_1, D_2 . Подобласть D_1 ограничена прямыми линиями, уравнения которых нужно составить. При этом удобно использовать уравнение прямой в отрезках на осях: $\frac{x}{a} + \frac{y}{b} = 1$. Подобласть D_2 ограничена кривой второго порядка, уравнение которой указано в задании. Варианты заданий приведены в таблице 2.

Пояснение задачи

1. Для заданного варианта из таблицы нарисовать область D и выписать функции $f_1(x, y)$ и $f_2(x, y)$. Записать уравнения, определяющие границы области, и подготовить исходные данные (координаты точек) так, чтобы часть точек находилась внутри области D , а часть — вне ее (см. рис. 1). При этом для каждой четверти координатной плоскости рекомендуется выбрать по две точки.

2. Создать на компьютере проект для решения поставленной задачи, составить и отладить программу.

3. Для выбранных точек получить и записать результаты счета.

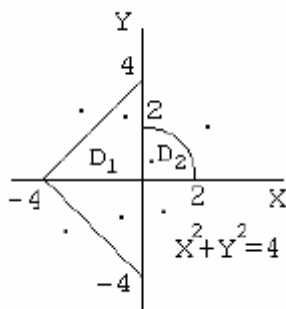


Рис. 1. Область на плоскости

Пусть область D имеет вид, показанный на рисунке 1, а функции f_1 и f_2 заданы как

$$f_1(x, y) = \ln(|x + y| + 1); \quad f_2(x, y) = e^{\frac{1}{|x| + |y|}}.$$

Прямой $x = 0$ разобьем область D на две подобласти: D_1 и D_2 . Уравнения прямых,

ограничивающих подобласть $D1$, будут: во второй четверти $y = x + 4$; в третьей $y = -x - 4$.

Уравнение окружности с центром в начале координат, ограничивающей подобласть $D2$ при $x > 0$ и $y > 0$, имеет вид $x^2 + y^2 = 4$.

Теперь можно записать, что некоторая точка (x, y) «попадает» в область D , если она «попадает» в область $D1$ или $D2$, то есть если $x \leq 0$ и $y \leq x + 4$ и $y \geq -x - 4$ или $x > 0$ и $y \geq 0$ и $x^2 + y^2 \leq 4$.

Точка (x, y) «не попадает» в область D , если $x \leq 0$ и $y > x + 4$ или $x \leq 0$ и $y < -x - 4$ или $x > 0$ и $y < 0$ или $x > 0$ и $x^2 + y^2 > 4$.

При выполнении задания можно проводить анализ как «попадания», так и «непопадания» точки в область D .

При составлении программы на Фортране для анализа «попадания» или «непопадания» точки в область D можно составить только одно логическое выражение с использованием следующих логических отношений и операций:

отношения:	операции
(в свободном и фиксированном форматах)	(в порядке убывания приоритета)
$< — .LT.$ — меньше,	
$<= — .LE.$ — меньше или равно, .	NOT. — отрицание,
$> — .GT.$ — больше, .	AND. — конъюнкция
(«и»),	
$>= — .GE.$ — больше или равно,	.OR. — дизъюнкция («или»),
$== — .EQ.$ — равно, .	EQV. —
эквивалентность,	
$/= — .NE.$ — не равно.	.NEQV. —
неэквивалентность.	

Результатом вычисления отношений является одно из двух логических значений .TRUE. — истина или .FALSE. — ложь. С помощью логических операций можно создавать сложные логические выражения, которые также могут принимать только два значения .TRUE. или .FALSE.

Например, точка (x, y) «попадает» в область D (см. рис. 1), если логическое выражение

$X \leq 0 \text{..AND.} Y \leq X + 4 \text{..AND.} Y \geq -X - 4 \text{..OR.} X > 0 \text{..AND.} Y \geq 0 \text{..AND.} X^2 + Y^2 \leq 4$ принимает значение .TRUE. — истина.

Ветвящиеся вычислительные процессы в Фортране можно запрограммировать с помощью следующих условных операторов IF.

Логический оператор IF имеет вид

IF(L) S

Здесь: L — логическое выражение, S — исполняемый оператор.

Если логическое выражение L истинно, то выполняется оператор S, если оно ложно, то выполнится следующий за IF оператор, а S не выполняется.

Блочный оператор IF имеет вид

IF(L) THEN

< блок операторов 1 >

ELSE

< блок операторов 2 >

END IF

Правило выполнения: если логическое выражение L истинно, то выполняется < блок операторов 1 >, а если оно ложно, то выполняется < блок операторов 2 >. При отсутствии альтернативы < блок операторов 2 > вместе с ELSE опускается.

Приведем пример Фортран-программы, в которой для анализа «попадания» или

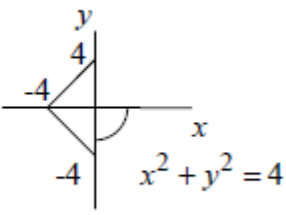
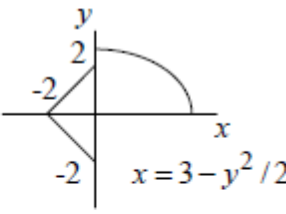
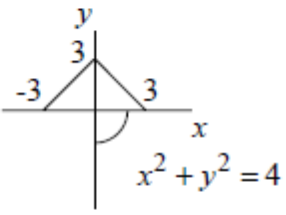
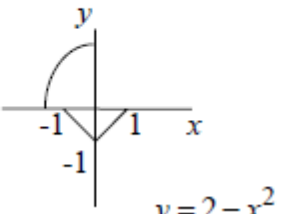
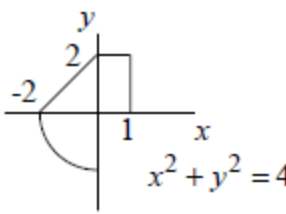
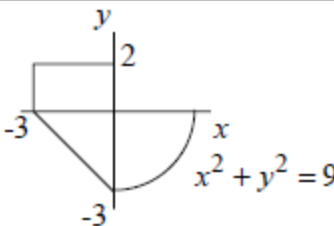
«непопадания» точки используется логическая переменная C. Этой переменной присваивается значение логического выражения. Если C=.TRUE., то в блочном операторе IF (C) THEN выполняется первый блок операторов, а если C=.FALSE., то второй блок (стоящий после ELSE).

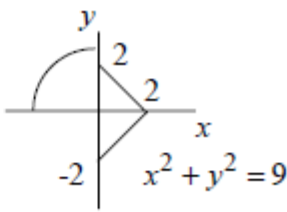
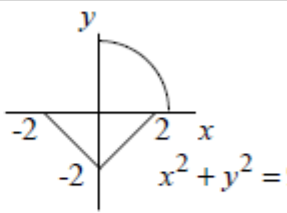
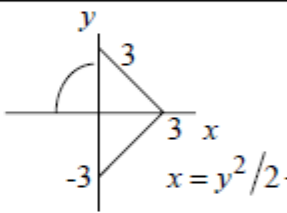
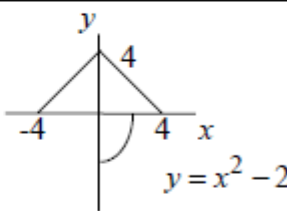
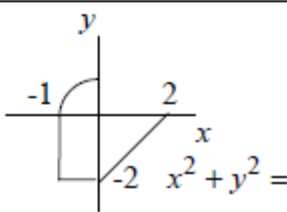
```
PROGRAM LAB2
LOGICAL C
REAL X, Y, Z
INTEGER I, M
DO I=1,8
    PRINT *, ' X=?, Y=? '; READ '(2F7.3)', X,Y
    C=X<=0..AND.Y<=X+4..AND.Y>=-X-4..OR.&
    &X>0..AND.Y>=0..AND.X**2+Y**2<=4.
    IF(C) THEN
        PRINT *, ' YES '
        Z=F1(X,Y); M=1
    ELSE
        PRINT *, ' NO '
        Z=F2(X,Y); M=2
    END IF
    PRINT 2, X,Y,Z,C,M
END DO
2 FORMAT (' X=',F7.3,3X,'Y=',F7.3,3X,'Z=',E11.4,3X,'C=', L2,3X,'M=',I2/)
END PROGRAM LAB2
FUNCTION F1(X,Y)
F1=ALOG(ABS(X+Y)+1.)
END FUNCTION F1
FUNCTION F2(X,Y)
F2=EXP(1./(ABS(X)+ABS(Y)))
END FUNCTION F2
```

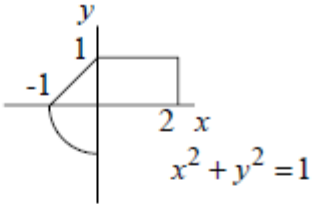
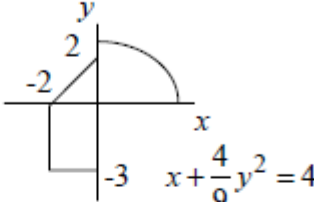
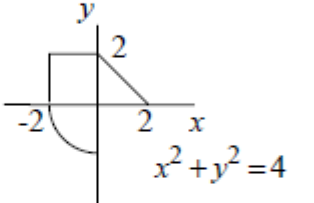
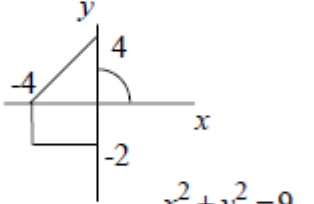
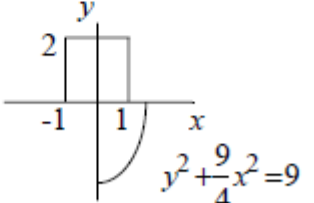
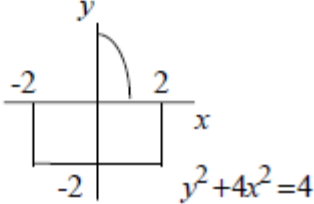
После окончания программы описаны внутренние процедуры-функции F1 и F2. Эта программа для своего завершения требует ввести координаты 8 точек.

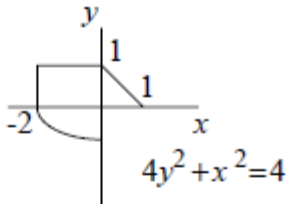
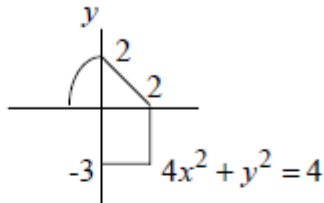
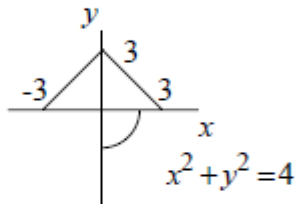
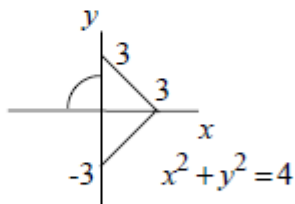
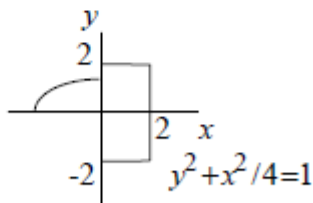
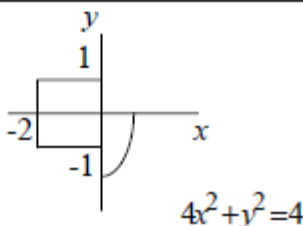
Варианты заданий приведены в таблице 2.

Таблица 2. Исходные данные

№ п/п	$f_1(x, y)$	$f_2(x, y)$	Область
1	$\operatorname{tg}(x) + \ln(y^2)$	$e^{\sin^2(x) + \cos^2(y)}$	
2	$\frac{\sin(x)}{1 - \cos(y)}$	$e^{\operatorname{arctg}(x^2 + y^2)}$	
3	$x^2 + \operatorname{tg}^2(y)$	$\ln(\sqrt{x^4 + 2y^2})$	
4	$\sqrt{ x \sin(y) }$	$\ln\left(\frac{1}{x^2} + \frac{1}{y^2}\right)$	
№ п/п	$f_1(x, y)$	$f_2(x, y)$	Область
5	$e^{2 \sin(x) + \cos(y)}$	$e^{\left(\frac{1}{x} + \frac{1}{y}\right)}$	
6	$x + \sin(y)$	$\operatorname{tg}(x^2 + y^2)$	

7	$2x + \operatorname{arctg}(y)$	$\operatorname{arctg}\left\{\frac{1}{x^2} + \frac{1}{y^2}\right\}$	
8	$\ln x^2 + 18y $	$\sqrt{\sin^2(x) + \cos^2(y)}$	
9	$x + \operatorname{tg}(y)$	$\ln x^3 + 3xy $	
10	$\operatorname{tg}\left(\frac{y}{\sin^2(x)}\right)$	$\sqrt{\frac{1}{x^2} + \frac{1}{y^2}}$	
11	$\frac{1}{\sin(x)} + \ln y $	$x^3 + x^2y + y^3$	

№ п/п	$f_1(x, y)$	$f_2(x, y)$	Область
13	$\sqrt{ x^2 + 10y }$	$\sqrt{x^2 + e^y}$	
14	$e^{\cos(y)+x}$	$\sqrt{ \sin(x) + \ln y }$	
15	$2\sin(x)\cos(y)$	$\sqrt{ x^2 + xy + y^2 }$	
16	$\frac{\operatorname{tg}(x)}{e^y}$	$\sqrt[3]{\sin^2(2x + 3y)}$	
17	$e^{5x} + 18\cos(y)$	$\frac{\sin(x^2 + y^2)}{x^2 + y^2}$	
18	$\ln\left(\frac{y}{\sin^2(x)}\right)$	$3\ln(x^2) + 5y$	

№ п/п	$f_1(x, y)$	$f_2(x, y)$	Область
19	$(\sin(x) + \operatorname{tg}(y))^3$	$\sqrt{e^{1/x^2+1/y^2}}$	
20	$\frac{\operatorname{tg}(y)}{1 - \cos^2(x)}$	$\frac{\sin(x)}{\sqrt{ xy }}$	
21	$\sin^2(x) + \ln y $	$\frac{x + \sin(y)}{\cos(xy)}$	
22	$\ln y - 3\sin(x) $	$\operatorname{arctg}(e^x + y^2)$	
23	$\sqrt{ y } - \cos^2(x)$	$\operatorname{arctg}(e^y + x^2)$	
24	$\frac{\sin(x)\cos(y)}{1 - \sin^2(x)}$	$\frac{\sin^2(x) + \cos^2(y)}{\sqrt{x^2 + y^2}}$	

Задание №3. ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

Цель работы: Получить навыки программирования циклических вычислительных процессов. Изучить циклы с шагом и циклы с условием

Задание.

1. Записать расчетные формулы для вычисления, слагаемого и конечной суммы ряда $S = \sum a_n(x)$. Варианты заданий приведены в таблице 3.

2. Составить алгоритм вычисления с заданной точностью суммы конечного числа членов ряда в указанном диапазоне изменения параметра x с заданным шагом. Для получения шага диапазон изменения x разделить на 10.

3. Составить программу решения задачи. В программе для каждого значения x предусмотреть вычисление как приближенной, так и точной суммы соответствующего ряда y по приведенной в таблице 3 формуле.

4. Отладить и протестировать программу. Результат получить в виде таблицы, содержащей для каждого x из указанного диапазона приближенное и точное значения суммы ряда, а также количество просуммированных членов.

Пояснение задачи

Работа содержит задачи, которые сводятся к нахождению конечной суммы ряда $S = \sum a_n(x)$ при различных значениях x . Каждое слагаемое a_n зависит от величины x и номера n , определяющего место этого слагаемого в сумме. Ряды подобраны таким образом, чтобы при бесконечно большом числе членов ряда их сумма в указанном диапазоне изменения величины x была конечным числом, зависящим только от x , т.е. ряды являются сходящимися.

Например, $S = \sum_{n=0}^{\infty} a_n(x) = \sum_{n=0}^{\infty} (-1)^n x^{2n+1} / (2n+1)! = \sin(x)$.

Первый член этого ряда при $n=0$ равен x , второй член при $n=1$ равен $(-x^3)/3!$, третий член равен $(-x^5)/5!$ и т.д. В сумме при бесконечно большом числе членов они дают функцию $\sin(x)$. Видно, что при ограниченных значениях x , сравнимых по модулю с единицей, каждый последующий член этого знакопеременного ряда существенно меньше предыдущего, поэтому ряд быстро сходится и для вычисления суммы ряда с необходимой точностью достаточно просуммировать относительно небольшое число членов ряда. Например, суммирование можно прекратить, когда текущий член ряда станет по абсолютной величине меньше заданного малого положительного числа ε , определяющего погрешность вычисления суммы ряда.

Формула общего члена суммы принадлежит к одному из следующих типов:

а) $x^n / n!$; $(-1)^n x^{2n-1} / (2n-1)!$; $x^{2n} / (2n)!$;

б) $\cos(nx) / n$; $\sin((2n-1)x) / (2n-1)$; $\cos(2nx) / (4n^2 - 1)$;

в) $x^{4n+1} / (4n+1)$; $(-1)^n \cos(nx) / n^2$; $(x/2)^2 (n^2 + 1) / n!$.

В случае (а) для вычисления члена суммы целесообразно использовать рекуррентные соотношения, т.е. выражать последующий член суммы через предыдущий, например, если

$$a_n = x^n / n!, \quad a_{n+1} = x^{n+1} / (n+1)! = (x^n / n!) (x / (n+1)) = a_n x / (n+1).$$

В случае (б) каждый член суммы вычисляется согласно общей формуле, которую и нужно запрограммировать.

В случае (в) член суммы целесообразно представить в виде двух сомножителей, один из которых вычисляется по рекуррентному соотношению, а другой – непосредственно, например, если

$$a_n = x^{4n+1} / (4n+1), \quad a_n = b_n c_n, \quad \text{где } b_n = 1 / (4n+1), \quad c_n = x^4 c_{n+1}$$

Пусть требуется составить программу для вычисления суммы ряда $S = -(2x)^2/2 + (2x)^4/24 + \dots + (-1)^n (2x)^{2n}/(2n)! + \dots$ и функции $y = 2(\cos^2(x) - 1)$ для $0.1 \leq x \leq 1.0$. Суммирование следует выполнять до тех пор, пока текущий член ряда по абсолютной величине не станет меньше заданного малого положительного числа ε .

Запишем ряд в виде

$$S = \sum a_n,$$

где $a_n = (-1)^n (2x)^{2n}/(2n)! = c_n b_n$, здесь $c_n = (-1)^n$; $b_n = (2x)^{2n}/(2n)!$.

Общий член ряда относится к типу (а), т.е. для вычисления члена суммы целесообразно использовать рекуррентные соотношения, выражая каждый последующий член суммы через предыдущий. Суммирование начинается с $n=1$, при этом начальные значения $c = -1$, $b = 1$.

Для каждого значения x , задаваемого во внешнем цикле, вычисление суммы ряда выполняется во внутреннем цикле, при прохождении которого номер члена ряда увеличивается на 1, а сумма изменяется по формуле $S = S + a$, где a вычисляется как произведение c и b . Начальное значение суммы в данном примере равно нулю $S=0$. Суммирование в глухом цикле DO необходимо продолжать до тех пор, пока величина a не станет по абсолютной величине меньше ε . Вычисление постоянной по модулю знакопеременной величины c_n можно организовать в цикле по формуле $c = -c$. Для вычисления величины $(2n)!$ в знаменателе необходимо во внутреннем цикле, где n каждый раз увеличивается на 1, число $2n$ умножать на число $(2n - 1)$, так как факториал – это, по определению, произведение последовательных натуральных чисел.

```
PROGRAM LAB3
REAL A, B, C, EPS, H, S, X, X0, XK, Y
INTEGER I, N
OPEN (1, FILE='LAB3.TXT')
X0=0.1; XK=1.; EPS=0.0001
H=(XK-X0)/10.
DO I=0,10
  X=X0+H*I
  N=1; C=-1.; B=1.; S=0.
  DO
    B=B*(2.*X)**2/(2.*N*(2.*N-1.)); A=C*B
    N=N+1; S=S+A; C=-C;
    IF (ABS(A) < EPS) EXIT
  END DO
  Y=2.*(COS(X)**2-1.)
  PRINT 2, X,S,Y,N; WRITE (1,2) X,S,Y,N
END DO
2 FORMAT (' X=',F6.3,3X,'S=',E10.3,3X,'Y=',E10.3,3X,'N=',I5)
END PROGRAM LAB3
```

Результаты счета выводятся не только на дисплей, но в файл LAB3.TXT. Вывод осуществляется под управлением оператора FORMAT с меткой 2, который формирует таблицу, содержащую для каждого x из указанного диапазона приближенное и точное значения суммы ряда, а также количество просуммированных членов.

Варианты заданий приведены в таблице 3.

Таблица 3. Варианты заданий.

№ п/п	Формула ряда	$x_0 \leq x \leq x_k$		Точное значение суммы ряда y
		x_0	x_k	
1	$S = 1 + x(\ln 3)/1 + x^2(\ln^2 3)/2! + \dots$ $+ x^n(\ln^n 3)/n! + \dots$	0.1	1.0	3^x
2	$S = \cos x + (\cos 2x)/2 + \dots$ $+ (\cos nx)/n + \dots$	$\pi/5$	$9\pi/5$	$-\ln(2 \sin(x/2))$
3	$S = x - x^3/3! + \dots$ $+ (-1)^n x^{2n+1}/(2n+1)! + \dots$	0.1	1.0	$\sin x$
4	$S = \sin x - (\sin 2x)/2 + \dots$ $+ (-1)^{n+1}(\sin nx)/n + \dots$	$\pi/5$	$4\pi/5$	$x/2$
5	$S = 1 + x/1 + x^2/2! + \dots + x^n/n! + \dots$	1.0	2.0	e^x
6	$S = 1 + x \cos(\pi/4) + \dots$ $+ x^n \cos(n\pi/4)/n! + \dots$	0.1	1.0	$\cos(x \sin(\pi/4))^*$ $e^{x \cos(\pi/4)}$
7	$S = 1 - x^2/2! + \dots + (-1)^n x^{2n}/(2n)! + \dots$	0.1	1.0	$\cos x$
8	$S = x \sin(\pi/4) + \dots$ $+ x^n \sin(n\pi/4) + \dots$	0.1	0.8	$\frac{x \sin(\pi/4)}{1 - 2x \cos(\pi/4) + x^2}$
9	$S = x + x^5/5 + \dots$ $+ x^{4n+1}/(4n+1) + \dots$	0.1	0.8	$\frac{1}{4} \ln \frac{1+x}{1-x} + \frac{1}{2} \operatorname{arctg} x$
10	$S = 1 + \cos x + \dots + \cos(nx)/n! + \dots$	0.1	1.0	$\cos(\sin x) e^{\cos x}$
11	$S = 1 + 3x^2 + \dots + (2n+1)x^{2n}/n! + \dots$	0.1	1.0	$(1+2x^2)e^{x^2}$
12	$S = x \cos(\pi/3) + x^2 \cos(2\pi/3)/2 + \dots$ $+ x^n \cos(n\pi/3)/n + \dots$	0.1	0.8	$\frac{-\ln(x^2 - 2x \cos(\pi/3) + 1)}{2}$

№ п/п	Формула ряда	$x_0 \leq x \leq x_k$	Точное значение суммы ряда y
13	$S = \frac{x-1}{x+1} + \frac{1}{3} \left(\frac{x-1}{x+1} \right)^3 + \dots$ $+ \frac{1}{2n+1} \left(\frac{x-1}{x+1} \right)^{2n+1} + \dots$	0.2 1.0	$\frac{\ln x}{2}$
14	$S = -\cos x + \frac{\cos 2x}{4} + \dots$ $+ (-1)^n \frac{\cos nx}{n^2} + \dots$	$\pi/5$ π	$\frac{1}{4} (x^2 - \pi^2/3)$
15	$S = x^3/3 - x^5/15 + \dots$ $+ (-1)^{n+1} x^{2n+1} / (4n^2 - 1) + \dots$	0.1 1.0	$\frac{1+x^2}{2} \operatorname{arctg} x - \frac{x}{2}$
16	$S = \sin x + \frac{1}{3} \sin 3x + \dots$ $+ \frac{1}{2n-1} \sin((2n-1)x) + \dots$	$\pi/10$ $9\pi/10$	$\pi/4$
17	$S = 1 + x^2/2! + \dots + x^{2n}/(2n)! + \dots$	0.1 1.0	$\frac{1}{2} (e^x + e^{-x})$
18	$S = \frac{1}{3} \cos 2x + \frac{1}{15} \cos 4x + \dots$ $+ \frac{1}{4n^2 - 1} \cos(2nx) + \dots$	0.1 0.8	$\frac{1}{2} - \frac{\pi}{4} \sin x $
19	$S = 1 + 2x + \dots + (2x)^n / n! + \dots$	0.1 1.0	e^{2x}
20	$S = 1 + 2 \frac{x}{2} + \dots + \frac{n^2 + 1}{n!} \left(\frac{x}{2} \right)^n + \dots$	0.1 1.0	$\left(1 + x/2 + x^2/4 \right) e^{x/2}$
21	$S = x - x^3/3 + \dots + (-1)^n \frac{x^{2n+1}}{2n+1} + \dots$	0.1 0.5	$\operatorname{arctg} x$

№ п/п	Формула ряда	$x_0 \leq x \leq x_k$		Точное значение суммы ряда y
22	$S = 1 - \frac{3}{2}x^2 + \dots$ $+(-1)^n \frac{2n^2 + 1}{(2n)!} x^{2n} + \dots$	0.1	1.0	$\left(1 - \frac{x^2}{2}\right) \cos x - \frac{x}{2} \sin x$
23	$S = -\frac{(2x)^2}{2} + \frac{(2x)^4}{24} + \dots$ $+(-1)^n \frac{(2x)^{2n}}{(2n)!} + \dots$	0.1	1.0	$2(\cos^2(x) - 1)$
24	$S = -(1+x)^2 + \frac{(1+x)^4}{2} + \dots$ $+(-1)^n \frac{(1+x)^{2n}}{n} + \dots$	-2.0	-1.0	$\ln \frac{1}{2 + 2x + x^2}$
25	$S = x + \frac{x^3}{3!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots$	0.1	1.0	$\frac{1}{2}(e^x - e^{-x})$
26	$S = x/3! + 4x^2/5! + \dots$ $+n^2 x^n / (2n+1)! + \dots$	0.2	0.8	$\frac{1}{4} \left(\frac{x+1}{\sqrt{x}} \operatorname{sh}(\sqrt{x}) - \operatorname{ch}(\sqrt{x}) \right)$
27	$S = x \cos(\pi/4) + x^2 \cos(2\pi/4) + \dots$ $+x^n \cos(n\pi/4) + \dots$	0.1	0.8	$\frac{x \cos(\pi/4) - x^2}{1 - 2x \cos(\pi/4) + x^2}$
28	$S = 3x + 8x^2 + \dots + n(n+2)x^n + \dots$	0.1	0.8	$x(3-x)/(1-x)^3$
29	$S = \cos x + \frac{\cos(3x)}{9} + \dots$ $+\frac{\cos((2n-1)x)}{(2n-1)^2} + \dots$	$\pi/5$	π	$\pi^2/8 - (\pi/4) x $
30	$S = x^2/2 - x^4/12 + \dots$ $+(-1)^{n+1} \frac{x^{2n}}{2n(2n-1)} + \dots$	0.1	0.8	$x \operatorname{arctg} x - \ln \sqrt{1+x^2}$

Задание №4. ИСПОЛЬЗОВАНИЕ МАССИВОВ

Цель работы. Изучить особенности описания и использования массивов в программе.

Задание. Разработать алгоритм решения выбранной по номеру варианта задачи, составить, отладить и просчитать программу с использованием массивов.

Пояснение задачи

Наряду с простыми переменными в алгоритмических языках используются переменные с индексами, называемые массивами. Массив – это упорядоченная последовательность величин, обозначаемая одним именем. Величины, из которых состоит массив, называются элементами массива. Элемент массива обозначается с помощью имени массива и индексов, следующих в круглых скобках за именем массива. Индексы отделяются друг от друга запятыми и указывают положение элемента в массиве. Индексов может быть 1, 2 и т.д. вплоть до 7. Соответственно массивы бывают одномерные, двумерные и т.д.

Все массивы, используемые в программе, должны быть описаны, так как для хранения элементов массива в определенной последовательности отводится необходимый объем памяти. При описании указываются имя массива и затем в круглых скобках через запятую граничные пары, т.е. разделенные двоеточием минимальные и максимальные значения соответствующего индекса. При отсутствии двоеточия минимальное значение соответствующего индекса в Фортране по умолчанию равно единице. Для описания массивов в Фортране используются операторы описания типа переменных (REAL, INTEGER, COMPLEX, LOGICAL, CHARACTER, DOUBLE PRECISION), а также атрибут и оператор DIMENSION.

Рассмотрим несколько примеров описания массивов. Значения граничных пар можно задавать с использованием именованных констант и константных выражений.

```
INTEGER, PARAMETER :: N=4
```

```
REAL, DIMENSION :: A(N,N), B(N), K(N+1,N+1), C(0:N)
```

В этом примере оператор REAL с атрибутом DIMENSION описывает двумерный вещественный массив A, содержащий 16 элементов; одномерный вещественный массив B, содержащий 4 элемента; двумерный вещественный массив K, содержащий 25 элементов, и одномерный вещественный массив C, содержащий 5 элементов.

Объекты с общими атрибутами можно группировать в единый список, например, следующий оператор описывает два двумерных вещественных массива D и E, каждый из которых имеет 3 строки и 4 столбца.

```
REAL, DIMENSION (3,4) :: D, E
```

Операторы описания типа могут использоваться для описания массивов и без атрибутов, но с указанием граничных пар, например,

```
INTEGER MIN(2,2), MAX(10)
```

```
REAL Q(5), SUMMA(3,3)
```

Здесь: MIN и MAX – целые массивы, содержащие 4 и 10 элементов соответственно; Q и SUMMA – вещественные массивы, содержащие 5 и 9 элементов.

Оператор DIMENSION также может использоваться для описания массивов, например, для описания одномерного вещественного массива F из 10 элементов

```
REAL F
```

```
DIMENSION F(-2:7)
```

Использование оператора DIMENSION без операторов описания типа, когда тип массива определяется по умолчанию возможно, но нежелательно. Операторы, описывающие массивы, необходимо располагать в начале программы перед первым исполняемым оператором.

В Фортране список операторов ввода READ и вывода PRINT или WRITE может содержать как имена простых переменных и элементов массива, так и имена массивов. В последнем случае вводу-выводу подлежат все элементы массива в «естественной последовательности» (т.е. сначала растет первый индекс, затем второй и т.д.). Так, если в

программе описан массив $A(2,2)$ и имеется оператор ввода `READ *, A`, то численные значения необходимо задавать в следующей последовательности: $A(1,1)$, $A(2,1)$, $A(1,2)$, $A(2,2)$, т.е. любая матрица вводится по столбцам. Для изменения порядка ввода-вывода или если требуется передать только часть элементов массива можно использовать неявную форму оператора цикла. Например, ввод матрицы $A(2,2)$ по строкам осуществляет оператор `READ *, ((A(I,J), J=1,2), I=1,2)`.

Не следует забывать, что в Фортране функции «что вводить (выводить)» и «как вводить (выводить)» распределены между операторами ввода-вывода и оператором `FORMAT`.

Если в программе предусмотрен ввод исходных данных с клавиатуры, то в процессе отладки приходится задавать одни и те же численные значения много раз. Поэтому, когда исходных данных достаточно много, удобнее записать их перед счетом программы в файл данных, а в программе предусмотреть оператор, считывающий данные из файла данных. Кроме того, результаты также имеет смысл выводить в файл данных, чтобы иметь возможность получить распечатку результатов счета. Обращение к файлу данных производится по его логическому номеру N , указанному в операторах `READ (N,...)` или `WRITE (N,...)`. С помощью оператора `OPEN` можно как назначать логические номера уже имеющимся файлам, так и создавать новые файлы с указанием соответствующего логического номера, например, `OPEN (1, FILE='LAB4.TXT')`. После выполнения этого оператора файлу `LAB4.TXT` будет соответствовать логический номер 1. Файл `LAB4.TXT` создается в папке проекта либо предварительно до запуска программы на счет, либо в процессе работы программы в момент выполнения оператора `OPEN`. Для завершения доступа к файлу используется оператор `CLOSE`.

Отличительной особенностью Фортрана является развитая система операций с массивами. Операции над массивами и сечениями массивов задают параллелизм действий над компонентами массивов (или массива). Такие средства, с одной стороны, позволяют пользователю лаконично и сжато описать алгоритм обработки массивов и, с другой стороны, дают возможность компилятору генерировать более эффективный код с учетом особенностей конкретного компьютера. Очевидно, что наиболее эффективно использование этих возможностей для вычислительных систем, имеющих аппаратные средства для векторной обработки.

В Фортран встроено большое число функций, позволяющих:

- выполнять вычисления в массивах, например, суммировать элементы массива или находить его максимальный элемент;
- преобразовывать массивы, например, можно получать из одномерного массива двумерный;
- получать справочные данные о массиве (размер, форма и значения границ каждого измерения).

1. Ввод одномерного массива $B(8)$ можно осуществить несколькими способами:

а) `READ (1,*) B`

Численные значения (8 значений) перед запуском программы на счет записываются через запятую или пробел в файл данных, соответствующий логическому номеру 1. Этот способ наиболее предпочтителен.

б) `PRINT 1`

`1 FORMAT (' INPUT VECTOR B(8)')`

`READ *, B`

После появления на экране монитора текста `INPUT VECTOR B(8)` численные значения элементов массива вводятся с клавиатуры (8 значений через запятую или пробел).

в) `PRINT 1`

`FORMAT (' INPUT VECTOR B')`

`DO I=1,8`

```

PRINT 2, I
READ *, B(I)
END DO
FORMAT (' B(', I2, ')=' )

```

Массив вводится с клавиатуры поэлементно по одному значению в строку после появления на экране соответствующей подсказки.

2. Ввод двумерного массива A(4,4)

а) из файла данных построчно с использованием неявной формы цикла

```
READ (1,*) ((A(I,J), J=1,4), I=1,4)
```

Перед выполнением программы численные значения записываются в файл данных, соответствующий логическому номеру 1, в четыре строки по четыре элемента в строку через запятую или пробел.

б) из файла данных построчно с использованием сечения массива (простейшей формы индексного триплета)

```
READ (1,*) (A(I,:), I=1,4)
```

Файл данных такой же, как и в случае а).

в) с клавиатуры построчно через запятую или пробел

```
READ *, (A(I,:), I=1,4)
```

3. Вывод одномерного массива B(4) на экран.

```
PRINT 5, B
5 FORMAT (' VECTOR B'/4F10.4)
```

При выводе в файл данных оператор PRINT заменяется на оператор WRITE, например, при выводе двумерного массива A(4,4)

```
WRITE (1,10) (A(I,:), I=1,4)
10 FORMAT (' MATRICA A'/4(4F10.4/))
```

4. Суммирование элементов одномерного массива B(10) $S = \sum_{i=1}^{10} b_i$

а) с использованием встроенной функции SUM

```
S=SUM(B)
```

б) по следующему алгоритму: начальное значение суммы принимается равным нулю $S = 0$, затем в цикле по i от 1 до 10 накапливается сумма $S = S + b_i$

```
S=0.
```

```
DO I=1,10
```

```
  S=S+B(I)
```

```
END DO
```

5. Произведение элементов одномерного массива B(10) $P = \prod_{i=1}^{10} b_i$

а) целесообразно использовать встроенную функцию PRODUCT

```
P=PRODUCT(B)
```

б) можно составить алгоритм, аналогичный алгоритму сложения элементов массива, в котором начальное значение произведения необходимо положить равным единице, а операцию сложения заменить произведением

```
P=1.
```

```
DO I=1,10
```

```
  P=P*B(I)
```

```
END DO
```

6. Пусть требуется определить максимальный элемент массива В из N элементов и его номер K.

а) Определение максимального (или минимального) элемента одномерного массива $S = \max(b_1, b_2, \dots, b_n)$ и номера k этого элемента выполняется по следующему алгоритму: сначала за максимальный принимается первый элемент массива $S = b_1$ и, следовательно, номером k максимального элемента будет единица $k = 1$. Затем в цикле по i от 2 до N текущее значение максимального S сравнивается с b_i . Если условие $S < b_i$ не выполняется, то цикл повторяется. Если условие $S < b_i$ выполняется, то текущему значению максимального S присваивается значение b_i , а его номеру – значение i .

```
INTEGER, PARAMETER :: N=10
REAL B(N) /1.,2.,3.,4.,5.,6.,5.,4.,3.,2./
S=B(1); K=1
DO I=2,N
    IF (S.LT.B(I)) THEN
        S=B(I); K=I
    END IF
END DO
PRINT *, ' S=', S, ' K=', K
END
```

б) Определение максимального (или минимального) элемента одномерного массива $S = \max(b_1, b_2, \dots, b_n)$ и номера k этого элемента можно выполнить с использованием встроенных функций MAXVAL и MAXLOC.

```
INTEGER, PARAMETER :: N=10
REAL B(N) /1.,2.,3.,4.,5.,6.,5.,4.,3.,2./
S=MAXVAL(B); K=MAXLOC(B)
PRINT *, ' S=', S, ' K=', K
END
```

7. Пусть требуется расположить элементы одномерного массива В(10) в порядке невозрастания, т.е. так чтобы для всех элементов массива выполнялось условие: $b_i \geq b_{i+1}$ при $i = 1, 9$. Рассмотрим один из способов решения этой задачи.

Сначала из всех элементов b_i при $i = 1, 10$ найти максимальный (его величину S и номер k), затем переставить первый и максимальный элементы. Из оставшихся девяти элементов b_i при $i = 2, 10$ найти максимальный и поменять его местами со вторым и т.д. Перестановка двух элементов массива с номерами k и j выполняется следующим образом: элемент b_k равен S , поэтому сначала следует заменить b_k на b_j , а затем b_j присваивается значение S . Программа упорядочения элементов массива будет иметь следующий вид:

```
DO J=1,9
    S=B(J); K=J
    DO I=J+1,10
        IF (S.LT.B(I)) THEN
            S=B(I); K=I
        END IF
    END DO
    B(K)=B(J); B(J)=S
END DO
```

Внешний цикл по j изменяет номер элемента, с которого начинается поиск.

8. Пусть требуется составить одномерный массив C(4) из максимальных элементов строк матрицы A(4,4). Программа строится по схеме двойного цикла. Параметром внешнего цикла является номер строки i . Во внутреннем цикле по j определяется максимальный элемент строки:

```

DO I=1,4
  C(I)=A(I,1)
  DO J=2,4
    IF (C(I).LT.A(I,J)) C(I)=A(I,J)
  END DO
END DO

```

9. Пусть требуется умножить матрицу $A(4,4)$ на вектор $B(4)$. Результатом умножения матрицы $A(4,4)$ на вектор $B(4)$ является вектор $C(4)$, элементы которого определяются по формуле $C_i = \sum_{j=1}^4 a_{ij} b_j$.

а) используем встроенную функцию MATMUL, которая выполняет матричное умножение операндов

$C = \text{MATMUL}(A, B)$

б) алгоритм строится по схеме двойного цикла: внешний цикл по параметру i , во внутреннем цикле по параметру j производится суммирование:

```

DO I=1,4
  C(I)=0.
  DO J=1,4
    C(I)=C(I)+A(I,J)*B(J)
  END DO
END DO

```

10. Пусть требуется найти произведение D матрицы A на матрицу F , где $A(4,4)$, $F(4,4)$ и $D(4,4)$. Умножение матрицы на матрицу осуществляется по схеме тройного цикла в соответствии с формулой $d_{ij} = \sum_{k=1}^4 a_{ik} f_{kj}$, где $i=1,4$ и $j=1,4$.

а) используем схему тройного цикла с суммированием во внутреннем цикле

```

DO I=1,4
  DO J=1,4
    D(I,J)=0.
    DO K=1,4
      D(I,J)=D(I,J)+A(I,K)*F(K,J)
    END DO
  END DO
END DO

```

б) используем схему двойного цикла со встроенной функцией суммирования

```

DO I=1,4
  DO J=1,4
    D(I,J)=SUM(A(I,:)*F(:,J))
  END DO
END DO

```

11. Пусть требуется вычислить элементы матрицы $D(4,4)$ по формулам:

$$d_{ij} = \begin{cases} (b_i + b_j)^2, & b_i + b_j < -1 \\ 2(b_i + b_j), & -1 \leq b_i + b_j \leq 1 \\ \sqrt{(b_i + b_j)}, & (b_i + b_j) > 1 \end{cases}$$

Программа, использующая блочный оператор IF, имеет следующий вид

```

DO I=1,4

```

```

DO J=1,4
  IF ((B(I)+B(J)).LT.-1.) THEN
    D(I,J)=(B(I)+B(J))*2
  ELSE IF ((B(I)+B(J)).GT.1.) THEN
    D(I,J)=SQRT(B(I)+B(J))
  ELSE
    D(I,J)=2.*(B(I)+B(J))
  END IF
END DO
END DO

```

Варианты заданий

В вариантах заданий в качестве исходных данных используются следующие массивы

$$A = \begin{pmatrix} -1.06 & 3.29 & -2.93 & 0.46 \\ 1.85 & -4.05 & 1.96 & -2.17 \\ 0.36 & 2.85 & -3.02 & 1.57 \\ 2.87 & -1.83 & 2.89 & -0.93 \end{pmatrix}; B = \begin{pmatrix} 1.07 \\ -2.36 \\ 1.98 \\ 0.76 \end{pmatrix}.$$

1. Заданы массивы A(4,4), B(4). Вычислить элементы массивов C(4), D(4,4), F(4,4) в соответствии с формулами:

$$c_i = \begin{cases} tg(0.1i + 1), & i = 1, 2 \\ ctg(0.2i + 1), & i = 3, 4 \end{cases}, d_{ij} = c_i + b_j, f_{ij} = a_{ij} + d_{ij}.$$

Вывести на печать массивы C, D, F.

2. Заданы массивы A(4,4), B(4). Вычислить элементы массива по формуле:

$$c_i = \begin{cases} \sqrt{b_i}, & b_i > 0 \\ b_i^2, & b_i \leq 0 \end{cases}$$

Найти максимальный элемент массива P=C+B. Преобразовать матрицу A, разделив все ее элементы на найденный максимум. Печатать A, B, C, P.

3. Заданы массивы A(4,4), B(4). Из 1-й строки A вычесть 3-ю и в полученной матрице найти максимальный элемент. Из всех элементов массива B вычесть найденный максимум. Печатать A, B.

4. Задан массив B(4). Вычислить элементы массива C(4) по формулам

$$c_i = \begin{cases} b_i^2, & b_i < -1 \\ 1 + b_i, & -1 \leq b_i \leq 1 \\ \sqrt{b_i}, & b_i > 1 \end{cases}$$

Вычислить элементы матрицы D(4,4) по формуле $d_{ij} = b_i c_j$. Найти $S = \sum_{i=1}^4 b_i c_i$. Из всех элементов массива B вычесть S. Печатать B, C, D, S.

5. Задан массив A(4,4). Составить вектор C(4) из минимальных элементов в столбцах A. Из каждой строки A вычесть C. Печатать A, C.

6. Задан массив A(4,4). Составить вектор C(4) из максимальных модулей элементов в строках A. Элементы вектора C разделить на максимальный элемент C. Из диагональных элементов A вычесть C. Печатать A, C.

7. Заданы массивы A(4,4), B(4). Найти C=AB. Максимальный элемент C и его номер k. Из k-й строки матрицы A вычесть C. Печатать A, B, C, k.

8. Заданы массивы A(4,4), B(4). Найти максимальные элементы массивов A и B и поменять их местами. Печатать A, B.

9. Задан массив B(4). Вычислить элементы массива C(4) по формулам

$$c_i = \begin{cases} 1 + \sqrt{b_i}, & b_i > 0 \\ 2b_i^2, & b_i \leq 0 \end{cases}$$

Вычислить элементы матриц $D(4,4)$ $d_{ij} = b_i + c_j$ и $F(4,4)$ $f_{ij} = b_i c_j$. Найти $Q=DF$. Печатать C, D, F, Q .

10. Заданы массивы $A(4,4)$, $B(4)$. Составить вектор $C(4)$ из максимальных элементов в строках A . Вычислить $P=BC$. Из диагональных элементов A вычесть P . Печатать A, B, C, P .

11. Задан массив $A(4,4)$. Вычислить элементы массива по формулам

$$d_{ij} = \begin{cases} \sqrt{a_{ij}}, & a_{ij} > 0 \\ a_{ij}^2 - 1, & a_{ij} \leq 0 \end{cases}$$

Найти $F=AD$. Печатать A, D, F .

12. Заданы массивы $A(4,4)$, $B(4)$. Найти максимальный элемент B и его номер k , минимальный элемент B и его номер l . В матрице A элемент a_{kl} заменить на найденный максимум, a_{lk} – на минимум. В векторе B поменять местами максимальный и минимальный элементы. Вычислить $C=AB$. Печатать A, B, C .

13. Заданы массивы $A(4,4)$, $B(4)$. Упорядочить B в порядке не убывания. Из каждой строки A вычесть B . Печатать A, B .

14. Заданы массивы $A(4,4)$, $B(4)$. Вычислить $C=AB$. Упорядочить C в порядке не возрастания. Из столбцов A вычесть C . Печатать A, B, C .

15. Задан массив $B(4)$. Вычислить $C(4)$ по формулам:

$$c_1 = \max(b_i), c_2 = \min(b_i), c_3 = c_1 + c_2, c_4 = c_1 - c_2.$$

Вычислить элементы $D(4,4)$ по формулам

$$d_{ij} = \begin{cases} b_i - c_j, & b_i > 0 \\ b_i + c_j, & b_i \leq 0 \end{cases}. \text{ Печатать } B, C, D.$$

16. Заданы массивы $A(4,4)$, $B(4)$. Упорядочить строки A по не убыванию значений первых элементов строк. Вычислить элементы вектора $C(4)$ по формуле $c_i = a_{ii} - b_i$. Печатать A, B, C .

17. Задан массив $B(4)$. Вычислить элементы матрицы $D(4,4)$ по формулам

$$d_{ij} = \begin{cases} \sin(b_i + b_j), & 0 \leq |b_i + b_j| < \pi \\ \cos(b_i + b_j), & \pi \leq |b_i + b_j| < 2\pi \\ \sin(b_i - b_j), & 2\pi \leq |b_i + b_j| \end{cases}$$

Найти максимальные элементы D и B , поменять их местами. Печатать B, D .

18. Задан массив $B(4)$. Вычислить элементы матрицы $D(4,4)$ по формулам

$$d_{ij} = \begin{cases} 2(b_i + b_j)^2, & b_i + b_j < -2 \\ 4(b_i + b_j), & -2 \leq b_i + b_j \leq 2 \\ \sqrt{b_i + b_j}, & b_i + b_j > 2 \end{cases}$$

Из всех элементов D вычесть максимальный элемент B . Вычислить $C=DB$. Печатать B, C, D .

19. Задан массив $A(4,4)$. Вычислить элементы вектора $C(4)$ по формулам:

$$c_1 = \max(a_{ij}), c_2 = (\max(a_{ij}) + \min(a_{ij}))/2,$$

$$c_3 = (\max(a_{ij}) - \min(a_{ij}))/2, c_4 = c_1 - c_2, c_4 = \min(a_{ij}).$$

Вычислить $P=AC$. Из диагональных элементов A вычесть вектор P . Печатать A, C, P .

20. Заданы массивы A(4,4), B(4). В матрице A найти минимальный и максимальный элементы, поменять их местами. Вычислить C=AB. Печатать A, B, C.

21. Заданы массивы A(4,4), B(4). В массиве B найти максимальный и минимальный элементы, поменять их местами. Вычислить C=AB. Из диагональных элементов A вычесть вектор C. Печатать A, B, C.

22. Задан массив A(4,4). Из диагональных элементов A вычесть максимальный элемент A. Вычислить вектор C(4) как сумму 1-й и 3-й строк A. Вычислить P=AC. Печатать A, C, P.

23. Заданы массивы A(4,4), B(4). Вычислить C=AB. Упорядочить C в порядке не убывания. Из диагональных элементов A вычесть вектор C. Печатать A, B, C.

24. Заданы массивы A(4,4), B(4). Вычислить элементы массива C(4) как сумму элементов в соответствующих строках A. Из диагональных элементов A вычесть C. Вычислить P=AC. Печатать A, C, P.

25. Задан массив A(4,4). Вычислить элементы массива D(4,4) по формулам

$$d_{ij} = \begin{cases} a_{ij} / 2, & i < j \\ 2a_{ij}^2, & i \geq j \end{cases}.$$

Вычислить F=AD. Печатать A, D, F.

26. Задан массив B(4). Вычислить элементы массива C(4) по формуле $c_k = \sum_{i=1}^k b_i$.

Вычислить элементы массива D(4,4) по формуле $d_{ij} = b_i c_j$. Печатать B, C, D.

27. Задан массив B(4). Вычислить элементы массива C(4) по формуле $c_k = \prod_{i=1}^k b_i$.

Вычислить элементы массива D(4,4) по формуле $d_{ij} = (b_i + c_j)^2$. Печатать B, C, D.

28. Заданы массивы A(4,4), B(4). Вычислить C=AB. Из первой строки A вычесть B, из второй строки A вычесть вектор C. Печатать A, B, C.

29. Заданы массивы A(4,4), B(4). Найти максимальный и минимальный элементы B и поменять их местами. Вычислить C=AB. Печатать A, B, C.

30. Задан массив B(4). Вычислить элементы массива C(4) по формуле: $c_i = i - \sin(0.4i + 1)$. Найти и поменять местами максимальные элементы B и C.

$d_{ij} = \sqrt{|b_i + c_j|}$. Печатать B, C, D.

Задание №5. РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Задание: Используя методы половинного деления и Ньютона, найти корни нелинейных уравнений $f_1(x)=0$.

Решение нелинейных (трансцендентных или алгебраических) уравнений вида $f(x)=0$ заключается в отыскании одного или нескольких корней, т.е. таких значений аргумента x , для которых функция $f(x)$ обращается в нуль. В общем случае функции $f(x)$ не имеют аналитических формул для своих корней, поэтому приходится использовать приближенные методы.

Решение нелинейных уравнений обычно состоит из двух этапов:

1) Отделение, или локализация корней, т.е. отыскание таких отрезков $[a,b]$ (одного или нескольких), внутри которых имеется только один корень нелинейного уравнения.

2) Уточнение приближенного значения корня до некоторой заданной степени точности. Отделить корни в некоторых случаях можно графически. Так, если уравнение имеет вид $\cos(x)-x=0$, то переписав его в виде $\cos(x)=x$ и построив графики функций $y_1=\cos(x)$ и $y_2=x$, найдем приближенное значение корня как точку пересечения функций y_1 и y_2 .

Локализовать корень уравнения можно также программно, вычисляя функцию $f(x)$ для значений x , изменяющихся с некоторым заданным шагом h . При этом отыскиваются два

таких соседних значения x , для которых $f(x)$ имеет противоположные знаки. Алгоритм отделения корня удобно оформить в виде подпрограммы. В качестве параметров подпрограммы можно выбрать следующие: F,XL,XR,H,A,B,IER.

Входные параметры:

F - имя внешней функции $f(x)$;

XL,XR - соответственно левая и правая границы отрезка оси x , на котором отделяется корень уравнения;

H - шаг перебора аргумента функции.

Выходные параметры:

A,B - соответственно левая и правая границы отрезка $[a,b]$, содержащего первый, считая от точки XL вправо, корень нелинейного уравнения;

IER - код ошибки. IER=0, если на участке от XL до XR с шагом H найден отрезок $[A,B]$, содержащий корень уравнения; IER=1, если корень не локализован.

Алгоритм отделения корня нелинейного уравнения может быть следующим:

- 1) Зададим IER=1, т.е. предполагаем, что на отрезке от XL до XR корней нет. В дальнейшем проверим, так ли это.
- 2) Зададим начальное значение аргумента функции $X=XL$.
- 3) Вычислим значение функции в точке X: $Y=F(X)$.
- 4) Изменим значение аргумента функции на величину шага: $X=X+H$.
- 5) Проверим, лежит ли X внутри $[XL,XR]$: если $X>XR$, то возвращаемся в вызывающую программу. В этом случае отрезок $[A,B]$ не найден, IER=1.
- 6) Вычислим значение функции в новой точке X: $Z=F(X)$.
- 7) Проверим, изменила ли функция знак при последнем изменении аргумента. Если не изменила, то опять меняем X: если $Y*Z>0$ идти на 4.
- 8) В противном случае определим границы отрезка $[A,B]$ по формулам $A=X-H$, $B=X$, изменим код ошибки IER=0 и вернемся в вызывающую программу.

Для уточнения корня нелинейного уравнения до заданной точности можно воспользоваться методами половинного деления, Ньютона, простых итераций и другими.

МЕТОД ПОЛОВИННОГО ДЕЛЕНИЯ состоит в построении последовательности вложенных отрезков, на концах которых функция принимает значения разных знаков. Каждый последующий отрезок получают делением предыдущего пополам. Процесс построения последовательности отрезков позволяет найти корень уравнения $f(x)=0$ с любой заданной точностью.

В качестве параметров подпрограммы метода половинного деления можно выбрать следующие: F,A,B,EPS,C,N,IER.

Входные параметры:

F - имя внешней функции $f(x)$;

A,B - соответственно левая и правая граница отрезка, содержащего один корень уравнения $f(x)=0$;

EPS - точность вычисления корня.

Выходные параметры:

C - корень уравнения (если он найден);

N - количество итераций, которое потребовалось выполнить для вычисления корня с заданной точностью;

IER - код ошибки. IER=0, если корень найден; IER=1, если $A>B$ или на отрезке $[A,B]$ нет корня, т.е. если $F(A)*F(B)>0$.

Алгоритм метода половинного деления:

- 1) Зададим код ошибки IER=1.
- 2) Проверим, правильно ли заданы исходные данные: если $F(A)*F(B)>0$ или $A>B$, то возвращаемся в вызывающую программу.
- 3) Зададим текущие границы отрезка, содержащего корень $A1=A$, $B1=B$; зададим начальное

число итераций $N=0$ и $IER=0$; вычислим значение функции в точке $A1$: $Y=F(A1)$.

4) Вычислим середину отрезка $[A1,B1]$: $C=.5*(A1+B1)$; изменим счетчик итераций на единицу: $N=N+1$.

5) Если длина текущего отрезка $[A1,B1]$ меньше заданной точности, то возвращаемся в вызывающую программу.

6) Проверим, внутри какого из отрезков $[A1,C]$ или $[C,B1]$ лежит корень уравнения. Если корень лежит внутри $[A1,C]$, то изменяем правую текущую границу отрезка, содержащего корень, в противном случае - левую: если $F(A1)*F(C)<0$, то $B1=C$, иначе $A1=C$; итерационный процесс повторяется с пункта 4.

Метод половинного деления - наиболее универсальный метод отыскания корней нелинейных уравнений. К его недостаткам относится невысокая скорость сходимости.

МЕТОД НЬЮТОНА (метод касательных) является одним из наиболее эффективных методов нахождения корней нелинейных уравнений. Он состоит в построении итерационной последовательности $x_{i+1}=x_i-f(x_i)/f'(x_i)$, сходящейся к корню уравнения $f(x)=0$. Геометрическая интерпретация метода следующая: если через точку с координатами $(x_i, f(x_i))$ провести касательную к функции $y=f(x)$, то точка пересечения этой касательной с осью абсцисс принимается за уточненное значение корня x_{i+1} . Метод Ньютона особенно эффективен, когда известно хорошее приближение корня и в окрестности корня функция имеет большую кривизну.

В качестве параметров подпрограммы метода Ньютона можно рекомендовать следующие:

F,PF,X0,EPS,NMAX,X,N,IER.

Входные параметры:

F - имя внешней функции $f(x)$;

PF - имя внешней функции $f'(x)$;

X0 - начальное приближение для корня уравнения;

EPS - точность вычисления корня;

NMAX - максимальное количество итерационных циклов.

Выходные параметры:

X - корень уравнения;

N - количество итераций, выполненных для вычисления корня с заданной точностью;

IER - код ошибки. Если корень найден, то $IER=0$; если корень не найден за NMAX итерационных циклов, то $IER=1$.

Используемый в подпрограмме метода Ньютона алгоритм:

1) Зададим: $IER=1$; начальное значение количества итераций $N=0$; текущее значение корня $X=X0$.

2) Выполним итерационный цикл: изменим содержимое счетчика итераций на единицу $N=N+1$; если количество итераций превысило NMAX, осуществим возврат в вызывающую программу; вычислим уточненное значение корня по формуле $Y=X-F(X)/PF(X)$; вычислим модуль разности между уточненным и текущим значениями корня $E=ABS(Y-X)$; в качестве нового текущего зададим уточненное значение корня $X=Y$; проверим, следует ли продолжать уточнение корня: если $E>EPS$, то итерационный цикл следует повторить с начала пункта 2.

3) Если точность вычисления корня достигнута, то следует изменить код ошибки $IER=0$ и осуществить возврат в вызывающую программу.

Один из недостатков метода Ньютона состоит в том, что пользуясь им, приходится дифференцировать функцию $f(x)$. Если сделать это затруднительно, то можно производную заменить конечной разностью.

При выполнении задания требуется:

1) Построить графики функций $y_1=f_1(x)$ и $y_2=f_2(x)$ в MathCad для того, чтобы определить количество корней и отрезки, где они расположены. Функции могут иметь один, несколько

или бесчисленное множество корней. Необходимо решить вопрос с преподавателем, какие корни подлежат определению.

2) Составить все необходимое для расчетов программное обеспечение: подпрограммы-функции для вычисления $f_1(x), f_1'(x), f_2(x), f_2'(x), p_1(x), p_2(x)$; подпрограммы для локализации корней, методов половинного деления, Ньютона, простых итераций; главную программу.

3) Провести счет. Результаты оформить в виде таблицы. Предусмотреть печать не только окончательных, но и промежуточных результатов (отрезков $[A, B]$, числа итераций, кодов ошибок)

Варианты задания приведены в таблице 4.

Таблица 4. Варианты заданий

N	f1(x)	f2(x)
1	$x + \sqrt{x} + x^{1/3} - 2.5$	$\operatorname{tg}(1.5773 \cdot x) - 2.3041 \cdot x$
2	$\cos(2/x) - 2 \cdot \sin(1/x) + 1/x$	$\ln(7.6221 \cdot x) - 8.591 \cdot x + 1.5$
3	$\cos(x) - \exp(-x^{2/2}) + x - 1$	$9.33 \cdot \sin(6.977 \cdot x) - 7.25 \cdot x$
4	$\sin(x^2) + \cos(x^2) - 10 \cdot x$	$0.9737 \cdot \exp(-.5067 \cdot x) - x$
5	$\operatorname{tg}(x/2) - \operatorname{ctg}(x/2) + x$	$\operatorname{tg}(2.2083 \cdot x) - 3.2258 \cdot x$
6	$x - \sin(x) - .25$	$\ln(6.0976 \cdot x) - 6.872 \cdot x + 2.01$
7	$\sqrt{x} - \cos(.387 \cdot x)$	$7.67 \cdot \sin(5.983 \cdot x) - 6.01 \cdot x$
8	$x^2 + 4 \cdot \sin(x)$	$0.9286 \cdot \exp(-.5185 \cdot x) - x$
9	$1.8 \cdot x^2 - \sin(10 \cdot x)$	$\operatorname{tg}(3.7855 \cdot x) - 5.5301 \cdot x$
10	$3 \cdot x - \cos(x) - 1$	$\ln(4.5732 \cdot x) - 3.963 \cdot x + 5.154$
11	$2 \cdot x - \ln(x) - 7$	$6.67 \cdot \sin(5.387 \cdot x) - 5.25 \cdot x$
12	$x^2 \cdot \cos(2 \cdot x) - 1$	$0.5458 \cdot \exp(-.5391 \cdot x) - x$
13	$\sin(x + \pi/3) - .5 \cdot x$	$\operatorname{tg}(9.1483 \cdot x) - 13.364 \cdot x$
14	$\cos(x + \pi/8) - x^3$	$\ln(3.9634 \cdot x) - 4.4868 \cdot x + 2.01$
15	$2 \cdot \operatorname{arctg}(x) - 3 \cdot x + 2$	$5.67 \cdot \sin(4.794 \cdot x) - 4.55 \cdot x$
16	$3 \cdot x - \cos(x) - 1$	$0.7593 \cdot \exp(-.5683 \cdot x) - x$
17	$(x-3) \cdot \cos(x) - 1$	$\operatorname{tg}(5.9937 \cdot x) - 8.7558 \cdot x$
18	$\sqrt{1-x} - \operatorname{tg}(x)$	$\ln(3.0488 \cdot x) - 3.436 \cdot x + 2.5$
19	$\operatorname{ctg}(1.05 \cdot x) - x^2$	$4.33 \cdot \sin(4.008 \cdot x) - 3.55 \cdot x$
20	$.6 \cdot 3 \cdot x - 2.3 \cdot x - 3$	$0.5909 \cdot \exp(-.6286 \cdot x) - x$

Рекомендуется в качестве отрезка $[A, B]$ для всех вариантов принять $[0; 1]$. Кроме № 10, интервал $[0; 1.5]$, №20, интервал $[1.5; 2.5]$, №23, интервал $[1.5; 2.5]$

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра инженерных проблем экологии



Расчетно-графическая работа
«название»
по дисциплине: «название»

Выполнил(а):
Студент(ка) гр. «название», «факультет»
«ФИО»
«__» _____ 20__ г.

(подпись)

Проверил:
«должность»
«ФИО»
«__» _____ 20__ г.

(подпись)

Новосибирск
20__