

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Качество и метрология программного обеспечения»
Тема: Расчёт метрических характеристик качества разработки
программ по метрикам Холстеда

Студентка гр. 8304

Сани З.Б.

Преподаватель

Кириячиков В. А.

Санкт-Петербург

2022

Цель работы.

Изучить метрические характеристики качества разработки программ на основе метрик Холстеда для программ на Pascal, C и ассемблере.

Ход выполнения.

Был выбран 17 вариант задания – вычисление функции ошибок распределения Гаусса (вар.1). Из исходного варианта программы был удалён ввод и вывод данных – с целью оптимизации для последующих работ. Исходный код полученной программы представлен в Приложении А. Для этой программы был произведён ручной расчёт операторов и операндов. Результат представлен в Таблице 1.

№	Оператор	Количество	№	Операнд	Количество
1	;	26	1	x	10
2	:=	16	2	er	4
3	<	2	3	ec	4
4	()	25	4	done	1
5	begin end	8	5	sqrtpi	4
6	if ... then	3	6	t2	2
7	+	23	7	t3	2
8	-	10	8	t4	2
9	*	31	9	t5	2
10	/	14	10	t6	2
11	exp()	2	11	t7	2
12	erf(x)	1	12	t8	2
13	erfc(x)	1	13	t9	2
			14	t10	2
			15	t11	2
			16	t12	2
			17	x2	16
			18	sum	6
			19	erf	1
			20	v	13
			21	erfc	1

			22	true	1
			23	1.7724538	2
			24	0.66666667	2
			25	0.07619048	1
			26	0.01693122	1
			27	3.078403E-3	1
			28	4.736005E-4	1
			29	6.314673E-5	1
			30	7.429027E-6	1
			31	7.820028E-7	1
			32	7.447646E-8	1
			33	6.476214E-9	1
			34	2.0	2
			35	1.0	5
			36	1	14
			37	8	1
			38	9	1
			39	10	1
			40	11	1
			41	12	1
			42	3	1
			43	4	1
			44	5	1
			45	6	1
			46	7	1
			47	2	1
			48	0.0	3
			49	1.5	1

Таблица 1 – Ручной расчёт операторов и операндов в программе на Pascal

Затем был произведён расчёт измеримых характеристик этой программы. Полученный результат представлен в Таблице 2.

Характеристика	Формула	Значение
Число уникальных операторов	η_1	13
Число уникальных операндов	η_2	49
Число всех операторов	N_1	162
Число всех операндов	N_2	128

Словарь программы	$\eta = \eta_1 + \eta_2$	62
Длина программы	$N = N_1 + N_2$	290

Таблица 2 – Ручное определение измеримых характеристик программы на Pascal

После чего были получены расчётные характеристики программы на Pascal. Результат представлен в Таблице 3.

Для расчётов значение коэффициента Страуда S принято 10; значение η_2^* принято 2, поскольку исследуемая процедура принимает x и возвращает значение функции ошибок.

Характеристика	Формула	Значение
Теоретическая оценка длины программы	$\check{N} = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2$	323,226
Реальный объём	$V = N \log_2 \eta$	1726,717
Потенциальный объём	$V^* = (2 + \eta_2^*) \log_2(2 + \eta_2^*)$	8
Уровень программы	$L = \frac{V^*}{V}$	0,0046
Оценка уровня программы	$\hat{L} = \frac{2}{\eta_1} \frac{\eta_2}{N_2}$	0,0589
Интеллектуальное содержание программы	$I = \frac{2}{\eta_1} \frac{\eta_2}{N_2} * (N_1 + N_2) \log_2(\eta_1 + \eta_2)$	101,694
Работа программиста	$E = \frac{V}{L}$	372693
Время программирования	$T = \frac{E}{S}$	37269
Оценка времени программирования	$T = \frac{\eta_1 N_2 N \log_2 \eta}{2S \eta_2}$	2931,89
Уровень языка	$\lambda = LV^*$	0,037
Ожидаемое число ошибок	$B = \frac{v}{1000}$	2

Таблица 3 – Ручное определение расчётных характеристик программы на Pascal

С помощью программы автоматизации расчёта метрик Холстеда были получены следующие результаты: подсчитаны операторы и операнды в программе на Pascal (см. Таблицу 4), определены измеримые характеристики программы на Pascal (см. Таблицу 5), определены расчётные характеристики программы на Pascal (см. Таблицу 6). Скриншоты результатов работы программы представлены в приложении Б.

№	Оператор	Количество	№	Операнд	Количество
1	()	37	1	x	15
2	*	31	2	er	5
3	+	23	3	ec	5
4	-	3	4	done	2
5	/	14	5	sqrtpi	4
6	;	38	6	t2	2
7	<	2	7	t3	2
8	=	30	8	t4	2
9	const	2	9	t5	2
10	erf	2	10	t6	2
11	erfc	2	11	t7	2
12	exp	2	12	t8	2
13	if	3	13	t9	2
14	program	1	14	t10	2
15	real	2	15	t11	2
			16	t12	2
			17	x2	18
			18	sum	8
			19	erf	1
			20	v	14
			21	erfc	1
			22	true	1
			23	1.7724538	2
			24	0.66666667	2
			25	0.07619048	1
			26	0.01693122	1
			27	3.078403E-3	1

			28	4.736005E-4	1
			29	6.314673E-5	1
			30	7.429027E-6	1
			31	7.820028E-7	1
			32	7.447646E-8	1
			33	6.476214E-9	1
			34	2.0	2
			35	1.0	5
			36	1	14
			37	8	1
			38	9	1
			39	10	1
			40	11	1
			41	12	1
			42	3	1
			43	4	1
			44	5	1
			45	6	1
			46	7	1
			47	2	1
			48	0.0	3
			49	1.5	1
			50	erfd4	1

Таблица 4 – Программный расчёт операторов и операндов в программе на Pascal

Характеристика	Формула	Значение
Число уникальных операторов	η_1	15
Число уникальных операндов	η_2	50
Число всех операторов	N_1	192
Число всех операндов	N_2	146
Словарь программы	$\eta = \eta_1 + \eta_2$	65
Длина программы	$N = N_1 + N_2$	338

Таблица 5 – Программное определение измеримых характеристик программы на Pascal

Характеристика	Формула	Значение
----------------	---------	----------

Теоретическая оценка длины программы	$\check{N} = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2$	340,796
Реальный объём	$V = N \log_2 \eta$	2035,56
Потенциальный объём	$V^* = (2 + \eta_2^*) \log_2(2 + \eta_2^*)$	8
Уровень программы	$L = \frac{V^*}{V}$	0,0039
Оценка уровня программы	$\hat{L} = \frac{2}{\eta_1} \frac{\eta_2}{N_2}$	0,0457
Интеллектуальное содержание программы	$I = \frac{2}{\eta_1} \frac{\eta_2}{N_2} * (N_1 + N_2) \log_2(\eta_1 + \eta_2)$	92,948
Работа программиста	$E = \frac{V}{L}$	517938
Время программирования	$T = \frac{E}{S}$	28774.3
Оценка времени программирования	$T = \frac{\eta_1 N_2 N \log_2 \eta}{2S \eta_2}$	2497,09
Уровень языка	$\lambda = LV^*$	0,0314
Ожидаемое число ошибок	$B = \frac{V}{1000}$	3

Таблица 6 – Программное определение расчётных характеристик программы на Pascal

На основе программы на языке Pascal была написана аналогичная программа на языке C. Эта программа представлена в Приложении В. Для этой программы был произведён ручной расчёт операторов и операндов. Результат представлен в Таблице 7.

№	Оператор	Количество	№	Операнд	Количество
1	;	32	1	x	10
2	=	29	2	er	4
3	<	2	3	ec	4

4	()	28	4	done	1
5	==	1	5	sqrtpi	4
6	If else	3	6	t2	2
7	+	23	7	t3	2
8	-	10	8	t4	2
9	*	31	9	t5	2
10	/	14	10	t6	2
11	exp()	2	11	t7	2
12	erf(x)	1	12	t8	2
13	erfc(x)	1	13	t9	2
14	return	3	14	t10	2
15	{}	9	15	t11	2
			16	t12	2
			17	x2	16
			18	sum	6
			19	erf	1
			20	v	13
			21	erfc	1
			22	true	1
			23	1.7724538	2
			24	0.66666667	2
			25	0.07619048	1
			26	0.01693122	1
			27	3.078403E-3	1
			28	4.736005E-4	1
			29	6.314673E-5	1
			30	7.429027E-6	1
			31	7.820028E-7	1
			32	7.447646E-8	1
			33	6.476214E-9	1
			34	2.0	2
			35	1.0	5
			36	1	14
			37	8	1
			38	9	1
			39	10	1
			40	11	1

			41	12	1
			42	3	1
			43	4	1
			44	5	1
			45	6	1
			46	7	1
			47	2	1
			48	0.0	3
			49	1.5	1
			50	0	1

Таблица 7 – Ручной расчёт операторов и операндов в программе на С

Затем был произведён расчёт измеримых характеристик этой программы. Полученный результат представлен в Таблице 8.

Характеристика	Формула	Значение
Число уникальных операторов	η_1	15
Число уникальных операндов	η_2	50
Число всех операторов	N_1	189
Число всех операндов	N_2	133
Словарь программы	$\eta = \eta_1 + \eta_2$	65
Длина программы	$N = N_1 + N_2$	322

Таблица 8 – Ручное определение измеримых характеристик программы на С

После чего были получены расчётные характеристики программы на С. Результат представлен в Таблице 9.

Для расчётов значение коэффициента Страуда S принято 10; значение η_2^* принято 2, поскольку исследуемая процедура принимает x и возвращает значение функции ошибок.

Характеристика	Формула	Значение
Теоретическая оценка длины Программы	$\check{N} = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2$	340,80
Реальный объём	$V = N \log_2 \eta$	1939,20
Потенциальный объём	$V^* = (2 + \eta_2^*) \log_2(2 + \eta_2^*)$	8

Уровень программы	$L = \frac{V^*}{V}$	0,0041
Оценка уровня программы	$\hat{L} = \frac{2}{\eta_1} \frac{\eta_2}{N_2}$	0,0501
Интеллектуальное содержание Программы	$I = \frac{2}{\eta_1} \frac{\eta_2}{N_2} * (N_1 + N_2) \log_2(\eta_1 + \eta_2)$	97,2031
Работа программиста	$E = \frac{V}{L}$	470063
Время программирования	$T = \frac{E}{S}$	47006
Оценка времени программирования	$T = \frac{\eta_1 N_2 N \log_2 \eta}{2S \eta_2}$	3868.71
Уровень языка	$\lambda = LV^*$	0,0330
Ожидаемое число ошибок	$B = \frac{V}{1000}$	2

Таблица 9 – Ручное определение расчётных характеристик программы на С

С помощью программы автоматизации расчёта метрик Холстеда были получены следующие результаты: подсчитаны операторы и операнды в программе на С (см. Таблицу 10), определены измеримые характеристики программы на С (см. Таблицу 11), определены расчётные характеристики программы на С (см. Таблицу 12). Скриншоты результатов работы программы представлены в приложении Г.

№	Оператор	Количество	№	Операнд	Количество
1	()	34	1	x	15
2	*	31	2	er	5
3	+	23	3	ec	5
4	,	7	4	done	2
5	-	2	5	sqrtpi	4

6	/	14	6	t2	2
7	;	36	7	t3	2
8	<	2	8	t4	2
9	=	29	9	t5	2
10	==	1	10	t6	2
11	_-	1	11	t7	2
12	erf	2	12	t8	2
13	erfc	2	13	t9	2
14	exp	2	14	t10	2
15	if	3	15	t11	2
16	main	1	16	t12	2
17	return	3	17	x2	18
			18	sum	8
			19	erf	1
			20	v	14
			21	erfc	1
			22	1.00	1
			23	1.7724538	2
			24	0.66666667	2
			25	0.07619048	1
			26	0.01693122	1
			27	3.078403E-3	1
			28	4.736005E-4	1
			29	6.314673E-5	1
			30	7.429027E-6	1
			31	7.820028E-7	1
			32	7.447646E-8	1
			33	6.476214E-9	1
			34	2.0	2
			35	1.0	5
			36	1	14
			37	8	1
			38	9	1
			39	10	1
			40	11	1
			41	12	1
			42	3	1
			43	4	1

			44	5	1
			45	6	1
			46	7	1
			47	2	1
			48	0.0	3
			49	1.5	1
			50	0	1

Таблица 10 – Программный расчёт операторов и операндов в программе на С

Характеристика	Формула	Значение
Число уникальных операторов	η_1	17
Число уникальных операндов	η_2	50
Число всех операторов	N_1	193
Число всех операндов	N_2	150
Словарь программы	$\eta = \eta_1 + \eta_2$	67
Длина программы	$N = N_1 + N_2$	343

Таблица 11 – Программное определение измеримых характеристик программы на С

Характеристика	Формула	Значение
Теоретическая оценка длины Программы	$\check{N} = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2$	351,68
Реальный объём	$V = N \log_2 \eta$	2080,67
Потенциальный объём	$V^* = (2 + \eta_2^*) \log_2(2 + \eta_2^*)$	8
Уровень программы	$L = \frac{V^*}{V}$	0,0038
Оценка уровня программы	$\hat{L} = \frac{2}{\eta_1} \frac{\eta_2}{N_2}$	0,039
Интеллектуальное содержание Программы	$I = \frac{2}{\eta_1} \frac{\eta_2}{N_2} * (N_1 + N_2) \log_2(\eta_1 + \eta_2)$	81,59
Работа программиста	$E = \frac{V}{L}$	541148

Время программирования	$T = \frac{E}{S}$	30063,8
Оценка времени программирования	$T = \frac{\eta_1 N_2 N \log_2 \eta}{2S \eta_2}$	3022,2
Уровень языка	$\lambda = LV^*$	0,0308
Ожидаемое число ошибок	$B = \frac{V}{1000}$	3

Таблица 12 – Программное определение расчётных характеристик программы на С

Был получен ассемблерный код этой программы в файле erfd.s. В полученном коде были удалены все комментарии и отладочные директивы. Итоговый ассемблерный код представлен в приложении Д. Для ассемблерной программы был произведён ручной расчёт операторов и операндов. Результат представлен в Таблице 13.

№	Оператор	Количество	№	Операнд	Количество
1	push	3	1	rbp	7
2	mov	10	2	rsp	5
3	movsd	111	3	QWORD PTR [rbp-136]	3
4	mulsd	28	4	xmm0	152
5	pop	1	5	QWORD PTR .LC0[rip]	4
6	ret	3	6	QWORD PTR [rbp-8]	2
7	sub	2	7	QWORD PTR .LC1[rip]	4
8	subsd	2	8	QWORD PTR [rbp-16]	4
9	addsd	26	9	QWORD PTR [rbp-24]	14
10	movapd	23	10	QWORD PTR .LC2[rip]	2
11	movq	14	11	QWORD PTR [rbp-32]	5
12	xorpd	1	12	QWORD PTR .LC3[rip]	2
13	call exp	2	13	QWORD PTR [rbp-40]	3
14	divsd	14	14	QWORD PTR .LC4[rip]	2
15	leave	2	15	QWORD PTR [rbp-48]	1
16	pxor	4	16	QWORD PTR .LC5[rip]	2
17	comisd	2	17	QWORD PTR [rbp-56]	4

18	jbe .L17	1	18	QWORD PTR .LC6[rip]	2
19	jmp .L8	3	19	QWORD PTR [rbp-64]	1
20	ucomisd	2	20	QWORD PTR .LC7[rip]	2
21	jne .L9	1	21	QWORD PTR [rbp-72]	1
22	jp .L9	1	22	QWORD PTR .LC8[rip]	2
23	jbe .L18	1	23	QWORD PTR [rbp-80]	1
24	call erf	1	24	QWORD PTR .LC9[rip]	2
25	call erfc	1	25	QWORD PTR [rbp-88]	1
			26	QWORD PTR .LC10[rip]	2
			27	QWORD PTR [rbp-96]	1
			28	QWORD PTR [rbp-104]	13
			29	xmm1	70
			30	QWORD PTR [rbp-112]	2
			31	QWORD PTR .LC11[rip]	1
			32	rax	18
			33	xmm2	42
			34	QWORD PTR .LC12[rip]	19
			35	QWORD PTR [rbp-120]	2
			36	QWORD PTR .LC13[rip]	1
			37	QWORD PTR .LC14[rip]	1
			38	xmm3	25
			39	QWORD PTR .LC15[rip]	1
			40	xmm4	20
			41	QWORD PTR .LC16[rip]	1
			42	xmm5	8
			43	QWORD PTR .LC17[rip]	1
			44	QWORD PTR .LC18[rip]	1
			45	QWORD PTR .LC19[rip]	1
			46	QWORD PTR .LC20[rip]	1

			47	QWORD PTR .LC21[rip]	1
			48	QWORD PTR .LC22[rip]	1
			49	QWORD PTR x[rip]	6
			50	DWORD PTR done[rip]	1
			51	QWORD PTR er[rip]	4
			52	QWORD PTR ec[rip]	4
			53	QWORD PTR .LC24[rip]	1
			54	eax	1
			55	0	1
			56	1	1
			57	64	1
			58	144	1

Таблица 13 – Ручной расчёт операторов и операндов в программе на ассемблере

Затем был произведён расчёт измеримых характеристик этой программы. Полученный результат представлен в Таблице 14.

Характеристика	Формула	Значение
Число уникальных операторов	η_1	25
Число уникальных операндов	η_2	58
Число всех операторов	N_1	259
Число всех операндов	N_2	482
Словарь программы	$\eta = \eta_1 + \eta_2$	83
Длина программы	$N = N_1 + N_2$	741

Таблица 14 – Ручное определение измеримых характеристик программы на ассемблере

После чего были получены расчётные характеристики программы на ассемблере. Результат представлен в Таблице 15.

Для расчётов значение коэффициента Страуда S принято 10; значение η_2^* принято 2, поскольку исследуемая процедура принимает x и возвращает значение функции ошибок.

Характеристика	Формула	Значение
Теоретическая оценка длины Программы	$\check{N} = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2$	455,86
Реальный объём	$V = N \log_2 \eta$	4723,90
Потенциальный объём	$V^* = (2 + \eta_2^*) \log_2(2 + \eta_2^*)$	8
Уровень программы	$L = \frac{V^*}{V}$	0,0017
Оценка уровня программы	$\hat{L} = \frac{2}{\eta_1} \frac{\eta_2}{N_2}$	0,0096
Интеллектуальное содержание Программы	$I = \frac{2}{\eta_1} \frac{\eta_2}{N_2} * (N_1 + N_2) \log_2(\eta_1 + \eta_2)$	45,47
Работа программиста	$E = \frac{V}{L}$	2789408
Время программирования	$T = \frac{E}{S}$	278940
Оценка времени программирования	$T = \frac{\eta_1 N_2 N \log_2 \eta}{2S \eta_2}$	49071,59
Уровень языка	$\lambda = LV^*$	0,0135
Ожидаемое число ошибок	$B = \frac{V}{1000}$	5

Таблица 15 – Ручное определение расчётных характеристик программы на ассемблере

Сводная таблица для всех трёх языков:

	Паскаль (руч.)	Паскаль (авт.)	Си (руч.)	Си (авт.)	Ассемблер
Теоретическая оценка длины программы	323,226	340,796	340,80	351,68	455,86
Реальный объём	1726,717	2035,56	1939,20	2080,67	4723,90
Потенциальный объём	8	8	8	8	8
Уровень программы	0,0046	0,0039	0,0041	0,0038	0,0017
Оценка уровня	0,0589	0,0457	0,0501	0,039	0,0096

программы					
Интеллектуальное содержание программы	101,694	92,948	97,2031	81,59	45,47
Работа программиста	372693	517938	470063	541148	2789408
Время программирования	37269	28774.3	47006	30063,8	278940
Оценка времени программирования	2931,89	2497,09	3868.71	3022,2	49071,59
Уровень языка	0,037	0,0314	0,0330	0,0308	0,0135
Ожидаемое число ошибок	2	3	2	3	5

Таблица 16 – Расчётные характеристики программ на паскале, си и ассемблере

Выводы.

В ходе выполнения лабораторной работы были изучены метрические характеристики качества разработки программ на основе метрик Холстеда. Для заданного варианта программы обработки данных, представленной на языке Паскаль, был разработан вычислительный алгоритм и также варианты программ его реализации на языках программирования Си и Ассемблер. Расчёт характеристик программ выполнен вручную и с помощью программы автоматизации расчета метрик Холстеда (для Си- и Паскаль-версий программ). По сводной таблице, содержащей метрики Холстеда для трёх программ, видно, что программа на ассемблере обладает наибольшей длиной, объёмом и временем программирования, а также потенциальным числом ошибок в ней. Кроме этого, ассемблер имеет наименьший уровень языка.

ПРИЛОЖЕНИЯ

Приложение А. Исходный код программы на Pascal.

```

program erfd4;

var    x,er,ec                : real;
       done                   : boolean;

function erf(x: real): real;
const sqrtpi                   = 1.7724538;
       t2                     = 0.66666667;
       t3                     = 0.66666667;
       t4                     = 0.07619048;
       t5                     = 0.01693122;
       t6                     = 3.078403E-3;
       t7                     = 4.736005E-4;
       t8                     = 6.314673E-5;
       t9                     = 7.429027E-6;
       t10                    = 7.820028E-7;
       t11                    = 7.447646E-8;
       t12                    = 6.476214E-9;

var    x2,sum                  : real;

begin
  x2:=x*x;
  sum:=t5+x2*(t6+x2*(t7+x2*(t8+x2*(t9+x2*(t10+x2*(t11+x2*t12))))));
  erf:=2.0*exp(-x2)/sqrtpi*(x*(1+x2*(t2+x2*(t3+x2*(t4+x2*sum)))));
end;

function erfc(x: real): real;
const sqrtpi                   = 1.7724538;

var    x2,v,sum                : real;

begin
  x2:=x*x;
  v:=1.0/(2.0*x2);
  sum:=v/(1+8*v/(1+9*v/(1+10*v/(1+11*v/(1+12*v)))));
  sum:=v/(1+3*v/(1+4*v/(1+5*v/(1+6*v/(1+7*sum)))));
  erfc:=1.0/(exp(x2)*x*sqrtpi*(1+v/(1+2*sum)));
end;

begin
  x := 1;
  if x<0.0 then done:=true
  else
    begin
      if x=0.0 then
        begin
          er:=0.0;
          ec:=1.0
        end
      else
        begin
          if x<1.5 then
            begin
              er:=erf(x);
              ec:=1.0-er
            end
          else
            begin
              er:=erf(x);
              ec:=1.0-er
            end
          end
        end
      end
    end
  end
end;

```

```
        end
    else
        begin
            ec:=erfc(x);
            er:=1.0-ec
        end { if }
    end;
end { if }
end.
```

Приложение Б. Результат работы программы автоматизации расчёта метрик Холстеда в программе на Pascal.

```
Statistics for module .\pascal.lxm
=====
The number of different operators      : 15
The number of different operands      : 50
The total number of operators         : 192
The total number of operands         : 146

Dictionary                           ( D) : 65
Length                               ( N) : 338
Length estimation                     ( ^N) : 340.796
Volume                               ( V) : 2035.56
Potential volume                      ( *V) : 8
Limit volume                         (**V) : 8
Programming level                     ( L) : 0.00393012
Programming level estimation          ( ^L) : 0.0456621
Intellect                            ( I) : 92.948
Time of programming                   ( T) : 28774.3
Time estimation                       ( ^T) : 2497.09
Programming language level           (lambda) : 0.031441
Work on programming                   ( E) : 517938
Error                                 ( B) : 2.1498
Error estimation                      ( ^B) : 0.67852
```

```
operators:
1 37 (
2 31 *
3 23 +
4 3 -
5 14 /
6 38 ;
7 2 <
8 30 =
9 2 const
10 2 erf
11 2 erfc
12 2 exp
13 3 if
14 1 program
15 2 real

operands:
1 3 0.0
2 1 0.01693122
3 1 0.07619048
4 2 0.66666667
5 14 1
6 5 1.0
7 1 1.5
8 2 1.7724538
9 1 10
10 1 11
```

```
11 1 12
12 1 2
13 2 2.0
14 1 3
15 1 3.078403E-3
16 1 4
17 1 4.736005E-4
18 1 5
19 1 6
20 1 6.314673E-5
21 1 6.476214E-9
22 1 7
23 1 7.429027E-6
24 1 7.447646E-8
25 1 7.820028E-7
26 1 8
27 1 9
28 2 done
29 5 ec
30 5 er
31 1 erf
32 1 erfc
33 1 erfd4
34 4 sqrtpi
35 8 sum
36 2 t10
37 2 t11
38 2 t12
39 2 t2
40 2 t3
41 2 t4
42 2 t5
43 2 t6
44 2 t7
45 2 t8
46 2 t9
47 1 true
48 14 v
49 15 x
50 18 x2
```

Приложение В. Исходный код программы на С.

```
#include <stdio.h>
#include <math.h>

double x,er,ec;
int done;

double erf(double x){
    const double sqrtpi = 1.7724538;
    const double t2      = 0.66666667;
    const double t3      = 0.66666667;
    const double t4      = 0.07619048;
    const double t5      = 0.01693122;
    const double t6      = 3.078403E-3;
    const double t7      = 4.736005E-4;
    const double t8      = 6.314673E-5;
    const double t9      = 7.429027E-6;
    const double t10     = 7.820028E-7;
    const double t11     = 7.447646E-8;
    const double t12     = 6.476214E-9;

    double x2,sum, erf;

    x2 = x*x;
    sum = t5+x2*(t6+x2*(t7+x2*(t8+x2*(t9+x2*(t10+x2*(t11+x2*t12))))));
    erf = 2.0*exp(-x2)/sqrtpi*(x*(1+x2*(t2+x2*(t3+x2*(t4+x2*sum)))));
    return erf;
}

double erfc(double x){
    const double sqrtpi = 1.7724538;

    double x2,v,sum, erfc;
    x2 = x*x;
    v = 1.0/(2.0*x2);
    sum = v/(1+8*v/(1+9*v/(1+10*v/(1+11*v/(1+12*v)))));
    sum = v/(1+3*v/(1+4*v/(1+5*v/(1+6*v/(1+7*sum)))));
    erfc = 1.0/(exp(x2)*x*sqrtpi*(1+v/(1+2*sum)));
    return erfc;
}

int main()
{
    x = 1.00;
    if (x < 0.0){
        done = 1;
    } else {
        if (x == 0.0){
            er = 0.0;
            ec = 1.0;
        }
        else{
            if (x < 1.5){
                er = erf(x);
                ec = 1.0 - er;
            } else {
                ec = erfc(x);
                er = 1.0 - ec;
            }
        }
    }
}
```

```
    return 0;  
}
```

Приложение Г. Результат работы программы автоматизации расчёта метрик Холстеда в программе на С.

```

Statistics for module .\c.1xm
=====
The number of different operators      : 17
The number of different operands      : 50
The total number of operators         : 193
The total number of operands         : 150

Dictionary                           ( D) : 67
Length                              ( N) : 343
Length estimation                     ( ^N) : 351.68
Volume                               ( V) : 2080.67
Potential volume                     ( *V) : 8
Limit volume                         (**V) : 8
Programming level                    ( L) : 0.00384492
Programming level estimation          ( ^L) : 0.0392157
Intellect                           ( I) : 81.5948
Time of programming                  ( T) : 30063.8
Time estimation                      ( ^T) : 3022.2
Programming language level           (lambda) : 0.0307593
Work on programming                  ( E) : 541148
Error                               ( B) : 2.21355
Error estimation                     ( ^B) : 0.693556

```

Operators:

1	34	()
2	31	*
3	23	+
4	7	,
5	2	-
6	14	/
7	36	;
8	2	<
9	29	=
10	1	==
11	1	_-
12	2	erf
13	2	erfc
14	2	exp
15	3	if
16	1	main
17	3	return

Operands:

1	1	0
2	3	0.0
3	1	0.01693122
4	1	0.07619048
5	2	0.66666667
6	14	1
7	5	1.0
8	1	1.00
9	1	1.5
10	2	1.7724538
11	1	10
12	1	11
13	1	12
14	1	2
15	2	2.0
16	1	3
17	1	3.078403E-3
18	1	4
19	1	4.736005E-4
20	1	5

21	1	6
22	1	6.314673E-5
23	1	6.476214E-9
24	1	7
25	1	7.429027E-6
26	1	7.447646E-8
27	1	7.820028E-7
28	1	8
29	1	9
30	2	done
31	5	ec
32	5	er
33	3	erf
34	3	erfc
35	4	sqrtpi
36	8	sum
37	2	t10
38	2	t11
39	2	t12
40	2	t2
41	2	t3
42	2	t4
43	2	t5
44	2	t6
45	2	t7
46	2	t8
47	2	t9
48	14	v
49	15	x
50	18	x2

Приложение Д. Ассемблерный код программы.

```
erf:
    push    rbp
    mov     rbp, rsp
    sub     rsp, 144
    movsd   QWORD PTR [rbp-136], xmm0
    movsd   xmm0, QWORD PTR .LC0[rip]
    movsd   QWORD PTR [rbp-8], xmm0
    movsd   xmm0, QWORD PTR .LC1[rip]
    movsd   QWORD PTR [rbp-16], xmm0
    movsd   xmm0, QWORD PTR .LC1[rip]
    movsd   QWORD PTR [rbp-24], xmm0
    movsd   xmm0, QWORD PTR .LC2[rip]
    movsd   QWORD PTR [rbp-32], xmm0
    movsd   xmm0, QWORD PTR .LC3[rip]
    movsd   QWORD PTR [rbp-40], xmm0
    movsd   xmm0, QWORD PTR .LC4[rip]
    movsd   QWORD PTR [rbp-48], xmm0
    movsd   xmm0, QWORD PTR .LC5[rip]
    movsd   QWORD PTR [rbp-56], xmm0
    movsd   xmm0, QWORD PTR .LC6[rip]
    movsd   QWORD PTR [rbp-64], xmm0
    movsd   xmm0, QWORD PTR .LC7[rip]
    movsd   QWORD PTR [rbp-72], xmm0
    movsd   xmm0, QWORD PTR .LC8[rip]
    movsd   QWORD PTR [rbp-80], xmm0
    movsd   xmm0, QWORD PTR .LC9[rip]
    movsd   QWORD PTR [rbp-88], xmm0
    movsd   xmm0, QWORD PTR .LC10[rip]
    movsd   QWORD PTR [rbp-96], xmm0
    movsd   xmm0, QWORD PTR [rbp-136]
    mulsd   xmm0, xmm0
    movsd   QWORD PTR [rbp-104], xmm0
    movsd   xmm1, QWORD PTR [rbp-104]
    movsd   xmm0, QWORD PTR .LC10[rip]
    mulsd   xmm1, xmm0
    movsd   xmm0, QWORD PTR .LC9[rip]
    addsd   xmm0, xmm1
    movapd  xmm1, xmm0
    mulsd   xmm1, QWORD PTR [rbp-104]
    movsd   xmm0, QWORD PTR .LC8[rip]
    addsd   xmm0, xmm1
    movapd  xmm1, xmm0
    mulsd   xmm1, QWORD PTR [rbp-104]
    movsd   xmm0, QWORD PTR .LC7[rip]
    addsd   xmm0, xmm1
    movapd  xmm1, xmm0
    mulsd   xmm1, QWORD PTR [rbp-104]
    movsd   xmm0, QWORD PTR .LC6[rip]
    addsd   xmm0, xmm1
    movapd  xmm1, xmm0
    mulsd   xmm1, QWORD PTR [rbp-104]
    movsd   xmm0, QWORD PTR .LC5[rip]
    addsd   xmm0, xmm1
    movapd  xmm1, xmm0
    mulsd   xmm1, QWORD PTR [rbp-104]
    movsd   xmm0, QWORD PTR .LC4[rip]
    addsd   xmm0, xmm1
    movapd  xmm1, xmm0
    mulsd   xmm1, QWORD PTR [rbp-104]
    movsd   xmm0, QWORD PTR .LC3[rip]
    addsd   xmm0, xmm1
    movsd   QWORD PTR [rbp-112], xmm0
```



```

movsd    xmm0, QWORD PTR [rbp-104]
movq     xmm1, QWORD PTR .LC11[rip]
xorpd    xmm0, xmm1
movq     rax, xmm0
movq     xmm0, rax
call     exp
addsd    xmm0, xmm0
movsd    xmm2, QWORD PTR .LC0[rip]
movapd   xmm1, xmm0
divsd    xmm1, xmm2
movsd    xmm0, QWORD PTR [rbp-104]
movapd   xmm2, xmm0
mulsd    xmm2, QWORD PTR [rbp-112]
movsd    xmm0, QWORD PTR .LC2[rip]
addsd    xmm0, xmm2
movapd   xmm2, xmm0
mulsd    xmm2, QWORD PTR [rbp-104]
movsd    xmm0, QWORD PTR .LC1[rip]
addsd    xmm0, xmm2
movapd   xmm2, xmm0
mulsd    xmm2, QWORD PTR [rbp-104]
movsd    xmm0, QWORD PTR .LC1[rip]
addsd    xmm0, xmm2
movapd   xmm2, xmm0
mulsd    xmm2, QWORD PTR [rbp-104]
movsd    xmm0, QWORD PTR .LC12[rip]
addsd    xmm0, xmm2
mulsd    xmm0, QWORD PTR [rbp-136]
mulsd    xmm0, xmm1
movsd    QWORD PTR [rbp-120], xmm0
movsd    xmm0, QWORD PTR [rbp-120]
movq     rax, xmm0
movq     xmm0, rax
leave
ret

erfc:
push     rbp
mov      rbp, rsp
sub      rsp, 64
movsd    QWORD PTR [rbp-56], xmm0
movsd    xmm0, QWORD PTR .LC0[rip]
movsd    QWORD PTR [rbp-8], xmm0
movsd    xmm0, QWORD PTR [rbp-56]
mulsd    xmm0, xmm0
movsd    QWORD PTR [rbp-16], xmm0
movsd    xmm0, QWORD PTR [rbp-16]
movapd   xmm1, xmm0
addsd    xmm1, xmm0
movsd    xmm0, QWORD PTR .LC12[rip]
divsd    xmm0, xmm1
movsd    QWORD PTR [rbp-24], xmm0
movsd    xmm1, QWORD PTR [rbp-24]
movsd    xmm0, QWORD PTR .LC13[rip]
mulsd    xmm0, xmm1
movsd    xmm2, QWORD PTR [rbp-24]
movsd    xmm1, QWORD PTR .LC14[rip]
mulsd    xmm1, xmm2
movsd    xmm3, QWORD PTR [rbp-24]
movsd    xmm2, QWORD PTR .LC15[rip]
mulsd    xmm2, xmm3
movsd    xmm4, QWORD PTR [rbp-24]
movsd    xmm3, QWORD PTR .LC16[rip]
mulsd    xmm3, xmm4
movsd    xmm5, QWORD PTR [rbp-24]

```

```

movsd    xmm4, QWORD PTR .LC17[rip]
mulsd    xmm5, xmm4
movsd    xmm4, QWORD PTR .LC12[rip]
addsd    xmm5, xmm4
movapd   xmm4, xmm3
divsd    xmm4, xmm5
movsd    xmm3, QWORD PTR .LC12[rip]
addsd    xmm4, xmm3
movapd   xmm3, xmm2
divsd    xmm3, xmm4
movsd    xmm2, QWORD PTR .LC12[rip]
addsd    xmm3, xmm2
movapd   xmm2, xmm1
divsd    xmm2, xmm3
movsd    xmm1, QWORD PTR .LC12[rip]
addsd    xmm2, xmm1
movapd   xmm1, xmm0
divsd    xmm1, xmm2
movsd    xmm0, QWORD PTR .LC12[rip]
addsd    xmm1, xmm0
movsd    xmm0, QWORD PTR [rbp-24]
divsd    xmm0, xmm1
movsd    QWORD PTR [rbp-32], xmm0
movsd    xmm1, QWORD PTR [rbp-24]
movsd    xmm0, QWORD PTR .LC18[rip]
mulsd    xmm0, xmm1
movsd    xmm2, QWORD PTR [rbp-24]
movsd    xmm1, QWORD PTR .LC19[rip]
mulsd    xmm1, xmm2
movsd    xmm3, QWORD PTR [rbp-24]
movsd    xmm2, QWORD PTR .LC20[rip]
mulsd    xmm2, xmm3
movsd    xmm4, QWORD PTR [rbp-24]
movsd    xmm3, QWORD PTR .LC21[rip]
mulsd    xmm3, xmm4
movsd    xmm5, QWORD PTR [rbp-32]
movsd    xmm4, QWORD PTR .LC22[rip]
mulsd    xmm5, xmm4
movsd    xmm4, QWORD PTR .LC12[rip]
addsd    xmm5, xmm4
movapd   xmm4, xmm3
divsd    xmm4, xmm5
movsd    xmm3, QWORD PTR .LC12[rip]
addsd    xmm4, xmm3
movapd   xmm3, xmm2
divsd    xmm3, xmm4
movsd    xmm2, QWORD PTR .LC12[rip]
addsd    xmm3, xmm2
movapd   xmm2, xmm1
divsd    xmm2, xmm3
movsd    xmm1, QWORD PTR .LC12[rip]
addsd    xmm2, xmm1
movapd   xmm1, xmm0
divsd    xmm1, xmm2
movsd    xmm0, QWORD PTR .LC12[rip]
addsd    xmm1, xmm0
movsd    xmm0, QWORD PTR [rbp-24]
divsd    xmm0, xmm1
movsd    QWORD PTR [rbp-32], xmm0
mov      rax, QWORD PTR [rbp-16]
movq     xmm0, rax
call     exp
movq     rax, xmm0
movq     xmm1, rax

```

```

    mulsd    xmm1, QWORD PTR [rbp-56]
    movsd    xmm0, QWORD PTR .LC0[rip]
    mulsd    xmm1, xmm0
    movsd    xmm0, QWORD PTR [rbp-32]
    movapd   xmm2, xmm0
    addsd    xmm2, xmm0
    movsd    xmm0, QWORD PTR .LC12[rip]
    movapd   xmm3, xmm2
    addsd    xmm3, xmm0
    movsd    xmm0, QWORD PTR [rbp-24]
    movapd   xmm2, xmm0
    divsd    xmm2, xmm3
    movsd    xmm0, QWORD PTR .LC12[rip]
    addsd    xmm0, xmm2
    mulsd    xmm1, xmm0
    movsd    xmm0, QWORD PTR .LC12[rip]
    divsd    xmm0, xmm1
    movsd    QWORD PTR [rbp-40], xmm0
    movsd    xmm0, QWORD PTR [rbp-40]
    movq     rax, xmm0
    movq     xmm0, rax
    leave
    ret

main:
    push     rbp
    mov      rbp, rsp
    movsd    xmm0, QWORD PTR .LC12[rip]
    movsd    QWORD PTR x[rip], xmm0
    movsd    xmm1, QWORD PTR x[rip]
    pxor     xmm0, xmm0
    comisd   xmm0, xmm1
    jbe      .L17
    mov      DWORD PTR done[rip], 1
    jmp      .L8

.L17:
    movsd    xmm0, QWORD PTR x[rip]
    pxor     xmm1, xmm1
    ucomisd   xmm0, xmm1
    jp       .L9
    pxor     xmm1, xmm1
    ucomisd   xmm0, xmm1
    jne      .L9
    pxor     xmm0, xmm0
    movsd    QWORD PTR er[rip], xmm0
    movsd    xmm0, QWORD PTR .LC12[rip]
    movsd    QWORD PTR ec[rip], xmm0
    jmp      .L8

.L9:
    movsd    xmm1, QWORD PTR x[rip]
    movsd    xmm0, QWORD PTR .LC24[rip]
    comisd   xmm0, xmm1
    jbe      .L18
    mov      rax, QWORD PTR x[rip]
    movq     xmm0, rax
    call     erf
    movq     rax, xmm0
    mov      QWORD PTR er[rip], rax
    movsd    xmm1, QWORD PTR er[rip]
    movsd    xmm0, QWORD PTR .LC12[rip]
    subsd    xmm0, xmm1
    movsd    QWORD PTR ec[rip], xmm0
    jmp      .L8

.L18:
    mov      rax, QWORD PTR x[rip]

```

```

        movq    xmm0, rax
        call    erfc
        movq    rax, xmm0
        mov     QWORD PTR ec[rip], rax
        movsd   xmm1, QWORD PTR ec[rip]
        movsd   xmm0, QWORD PTR .LC12[rip]
        subsd   xmm0, xmm1
        movsd   QWORD PTR er[rip], xmm0
.L8:
        mov     eax, 0
        pop     rbp
        ret

```

