

CX_Compiler
软件测试说明书

作者:周齐
版本:v1.0
日期:2015/01/09

1. 概述

测试描述	
测试环境	Windows 7 (32 bit) Visual Studio 2010 Bison 2.4.1 Flex 2.5.4 MinGW 5.1.6

2. 测试用例描述

Test No.1	prime.txt	100以内的素数	
测试目标			
测试用例的源程序	<pre>{ int a; a = 2; while(a < 101){ int i; int f; f = 0; i = 2; while(i * i <= a){ int b; b = a / i; if((b * i) == a){ f = 1; }; i = i + 1; }; if(f == 0) write a;; a = a + 1; }; }</pre>		
测试步骤	命令行输入： flex cxLex.l bison -d cxYacc.y gcc lex.yy.c cxyacc.tab.c -o cx.exe cx.exe prime.txt		

中间代码及 测试结果	0	jmp	0	1	28	opr	0	8	2
	1	ini	0	4	29	jpc	0	35	3
	2	lit	0	2	30	ini	0	3	5
	3	sto	0	3	31	lit	0	1	7
	4	lod	0	3	32	sto	2	4	11
	5	lit	0	101	33	opr	0	22	13
	6	opr	0	10	34	jmp	0	35	17
	7	jpc	0	54	35	lod	1	3	19
	8	ini	0	5	36	lit	0	1	23
	9	lit	0	0	37	opr	0	2	29
	10	sto	0	4	38	sto	1	3	31
	11	lit	0	2	39	opr	0	22	37
	12	sto	0	3	40	jmp	0	13	41
	13	lod	0	3	41	lod	0	4	43
	14	lod	0	3	42	lit	0	0	47
	15	opr	0	4	43	opr	0	8	53
	16	lod	1	3	44	jpc	0	48	59
	17	opr	0	13	45	lod	1	3	61
	18	jpc	0	41	46	opr	0	14	67
	19	ini	0	4	47	jmp	0	48	71
	20	lod	2	3	48	lod	1	3	73
	21	lod	1	3	49	lit	0	1	79
	22	opr	0	5	50	opr	0	2	83
	23	sto	0	3	51	sto	1	3	89
	24	lod	0	3	52	opr	0	22	97
	25	lod	1	3	53	jmp	0	4	
	26	opr	0	4	54	opr	0	0	
	27	lod	2	3					

Test No.2	factor.txt	阶乘	
测试目标			
测试用例的源程序	<pre> { int n; while(1) { int ans; ans = 1; read n; n++; while(--n) { ans = ans * n; }; write ans; }; } </pre>		
测试步骤	命令行输入: flex cxLex.l bison -d cxYacc.y gcc lex.yy.c cxyacc.tab.c -o cx.exe cx.exe factor.txt 10 9 5		
中间代码及 测试结果	0 jmp 0 1 1 ini 0 4 2 lit 0 1 3 jpc 0 30 4 ini 0 4 5 lit 0 1 6 sto 0 3 7 opr 0 16 8 sto 1 3 9 lit 0 1 10 lod 1 3 11 opr 0 2 12 sto 1 3 13 lit 0 -1 14 lod 1 3 15 opr 0 2	16 sto 1 3 17 lod 1 3 18 jpc 0 26 19 ini 0 3 20 lod 1 3 21 lod 2 3 22 opr 0 4 23 sto 1 3 24 opr 0 22 25 jmp 0 13 26 lod 0 3 27 opr 0 14 28 opr 0 22 29 jmp 0 2 30 opr 0 0	?10 3628800 ?9 362880 ?5 120 ?

Test No.3	lcm.txt	最小公倍数	
测试目标			
测试用例的源程序	<pre> { int m; int n; int r; int t; int p; int q; read m; read n; if (m > n) t=n; else t=m;; p = m%t; q = n%t; while(p!=0 q!=0) { t=t-1; p=m%t; q=n%t; }; r=m*n; r=r/t; write r; } </pre>		
测试步骤	命令行输入： flex cxLex.l bison -d cxYacc.y gcc lex.yy.c cxyacc.tab.c -o cx.exe cx.exe GBS.txt 16 24		
中间代码及 测试结果	0 jmp 0 1 1 ini 0 9 2 opr 0 16 3 sto 0 3 4 opr 0 16 5 sto 0 4 6 lod 0 3 7 lod 0 4 8 opr 0 12 9 jpc 0 13 10 lod 0 4 11 sto 0 6 12 jmp 0 15 13 lod 0 3 14 sto 0 6 15 lod 0 3 16 lod 0 6 17 opr 0 21 18 sto 0 7 19 lod 0 4 20 lod 0 6 21 opr 0 21 22 sto 0 8 23 lod 0 7 24 lit 0 0 25 opr 0 9 26 lod 0 8 27 lit 0 0 28 opr 0 9	29 opr 0 18 30 jpc 0 46 31 ini 0 3 32 lod 1 6 33 lit 0 1 34 opr 0 3 35 sto 1 6 36 lod 1 3 37 lod 1 6 38 opr 0 21 39 sto 1 7 40 lod 1 4 41 lod 1 6 42 opr 0 21 43 sto 1 8 44 opr 0 22 45 jmp 0 23 46 lod 0 3 47 lod 0 4 48 opr 0 4 49 sto 0 5 50 lod 0 5 51 lod 0 6 52 opr 0 5 53 sto 0 5 54 lod 0 5 55 opr 0 14 56 opr 0 0	?16 ?24 48

Test No.4	test4.txt	call函数调用	
测试目标			
测试用例的源程序	<pre> /*110,1110,30*/ { int a1; int a2; function p1 { int b1; int b2; function p2 { int c1; int c2; c1=1000; c2=2000; write a1+b1+c1; }; b1=100; b2=200; write a1+b1; call p2; }; a1=10; a2=20; call p1; write a1+a2; } </pre>		
测试步骤	命令行输入： flex cxLex.l bison -d cxYacc.y gcc lex.yy.c cxyacc.tab.c -o cx.exe cx.exe test4.txt		
中间代码及 测试结果	0 jmp 0 24 1 ini 0 5 2 lit 0 1000 3 sto 0 3 4 lit 0 2000 5 sto 0 4 6 lod 2 3 7 lod 1 3 8 opr 0 2 9 lod 0 3 10 opr 0 2 11 opr 0 14 12 opr 0 0 13 ini 0 5 14 lit 0 100 15 sto 0 3 16 lit 0 200 17 sto 0 4 18 lod 1 3	19 lod 0 3 20 opr 0 2 21 opr 0 14 22 cal 0 1 23 opr 0 0 24 ini 0 5 25 lit 0 10 26 sto 0 3 27 lit 0 20 28 sto 0 4 29 cal 0 13 30 lod 0 3 31 lod 0 4 32 opr 0 2 33 opr 0 14 34 opr 0 0	110 1110 30