**Курсовая работа.**

**Часть 1.**

**Студент: Сорокин Д.М.**

**Группа: 8О-204Б**

**Номер по списку: 15**

**Вариант: а15**

**Дата: 29.05.2017**

**Тема: Разработка синтаксически управляемого транслятора в язык С++.**

**Задание: Для заданного диалекта языка МИКРОЛИСП на базе класса tCG разработать синтаксически управляемый транслятор (генератор кода) в язык С++.**

**Работоспособность транслятора проверить на трех контрольных задачах из лабораторных работ №1, №2 и №3:**

1. **Проверка числа на простоту.**
2. **Решение уравнения методом половинного деления.**
3. **Размен денег.**

**Тексты контрольных задач адаптировать к заданному диалекту языка с использованием всех доступных грамматических форм (в порядке убывания приоритетов).**

**Для записи условий:**

1. **Операторы отношения и логические операторы;**
2. **Определенные в тексте программы процедур-предикаты.**

**Для записи промежуточных результатов вычислений:**

1. **Пределение локальной переменной;**
2. **Присваивание глобальной переменной;**
3. **Присваивание параметру процедуры.**

**Для записи именованных констант:**

1. **Определение глобальной переменной;**
2. **Определение процедуры без параметров.**

**Если диалект позволяет сохранить грамматическую форму, использованную в лабораторной работе, запрещается заменять ее другой формой языка МИКРОЛИСП.**

**Перечень документации:**

* **полный текст задания;**
* **описание грамматики ;**
* **описание особенностей гамматики по форме**

**GrammaFeatures.txt;**

* **протоколы трансляции контрольных задач**

**без трассировки;**

* **протоколы запуска контрольных задач на ЛИСПе и С++;**
* **распечатку файла code-gen.cpp;**
* **диаграммы автоматов из лабораторной работы №5 для токенов $float, $id, $idq, над каждой диаграммой проставить номер варианта шаблона токена и его краткое описание, все диаграммы должны быть построены в одном редакторе и должны иметь единый стиль изображения;**
* **выводы по проделанной работе - не менее одной страницы не разбавленного «водой» текста.**

**…………………………………………………………………………………………………………………………………….**

**Вариант a15:**

**Грамматика # $a15**

**$id $idq $int $float**

**$zero $bool $str (**

**) + - \***

**/ < = >**

**<= >= and not**

**or cond else if**

**define display newline set!**

**#**

**S -> PROG #1**

**CALC -> E1 #2 |**

**BOOL #3 |**

**STR #4 |**

**SET #5 |**

**DISP #6**

**DISP -> ( display E1 ) #7 |**

**( display BOOL ) #8 |**

**( display STR ) #9 |**

**( newline ) #10**

**PRED -> HPRED BOOL ) #11**

**HPRED -> PDPAR ) #12**

**PDPAR -> ( define ( $idq #13 |**

**PDPAR $id #14**

**CPROC -> HCPROC ) #15**

**HCPROC -> ( $id #16 |**

**HCPROC E #17**

**CPRED -> HCPRED ) #18**

**HCPRED -> ( $idq #19 |**

**HCPRED E #20**

**E -> $id #21 |**

**$zero #22 |**

**ADD #23 |**

**SUB #24 |**

**DIV #25 |**

**MUL #26 |**

**COND #27 |**

**CPROC #28**

**ADD -> HADD E1 ) #29**

**HADD -> ( + #30 |**

**HADD E1 #31**

**MUL -> HMUL E1 ) #32**

**HMUL -> ( \* #33 |**

**HMUL E1 #34**

**COND -> HCOND CLAUS ) #35**

**HCOND -> ( cond CLAUS #36**

**CLAUS -> HCLAUS E1 ) #37**

**HCLAUS -> ( BOOL #38**

**STR -> $str #39 |**

**SIF #40**

**SIF -> SIFTRUE STR ) #41**

**SIFTRUE -> HIF STR #42**

**HIF -> ( if BOOL #43**

**E1 -> E #44**

**SET -> HSET E1 ) #45**

**HSET -> ( set! $id #46**

**CONST -> $int #47 |**

**$float #48**

**PROC -> PRBODY ) #49**

**PRBODY -> HPROC E #50 |**

**HPROC CONST #51**

**HPROC -> PCPAR ) #52 |**

**HPROC SET #53 |**

**HPROC DISP #54**

**PCPAR -> ( define ( $id #55 |**

**PCPAR $id #56**

**DEF -> PRED #57 |**

**PROC #58**

**DEFS -> DEF #59 |**

**DEFS DEF #60**

**CALCS -> CALC #61 |**

**CALCS CALC #62**

**CALCS1 -> CALCS #63**

**PROG -> CALCS1 #64 |**

**DEFS #65 |**

**DEFS CALCS1 #66**

**SUB -> HSUB E1 ) #67**

**HSUB -> ( - E1 #68**

**DIV -> ( / E1 ) #69**

**BOOL -> $bool #70 |**

**CPRED #71 |**

**REL #72 |**

**OR #73 |**

**( not BOOL ) #74**

**HREL -> ( > E #75 |**

**( = E #76**

**OR -> HOR BOOL ) #77**

**HOR -> ( or #78 |**

**HOR BOOL #79**

**REL -> HREL E1 ) #80**

**Особенности грамматики:**

**1. Вычитание.**

**1.2\* Только два операнда.**

**(- x y)**

**2. Деление.**

**2.2\* Только один операнд.**

**(/ x)**

**3. Числовые литералы токена $zero.**

**3.1\* В общем контексте числового выражения.**

**0**

**4. Числовые литералы токенов $int и $float.**

**4.3\* Только в определении процедуры.**

**(define (one) 1)(+ (one) (one))**

**5. Оператор OR.**

**5.1\* Один и более операндов.**

**(or #t #f #f)**

**6. Оператор AND.**

**6.3\* Отсутствует.**

**7. Оператор = .**

**7.1\* Есть.**

**(= x y)**

**8. Оператор отношения, кроме оператора = .**

**8.3\* (> x y)**

**9. Оператор IF для чисел.**

**9.2\* Отсутствует.**

**10. Оператор IF для строк.**

**10.1\* Есть.**

**(display(if (p?)"Yes""No"))**

**11. Оператор COND.**

**11.8\* Без ветви ELSE, две простые клаузы.**

**(cond(#f e)(#t pi))**

**12. Определяемые переменные.**

**12.4 Отсутствуют.**

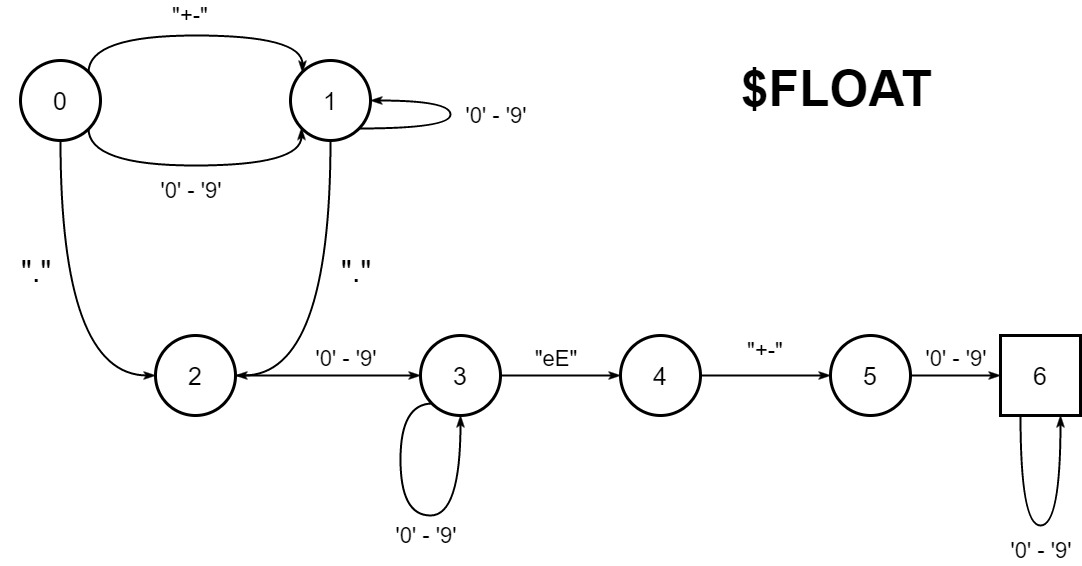
**(define(f a)(set! a pi)a)(f e)**

**…………………………………………………………………………………………………………………………………….**

**$float**

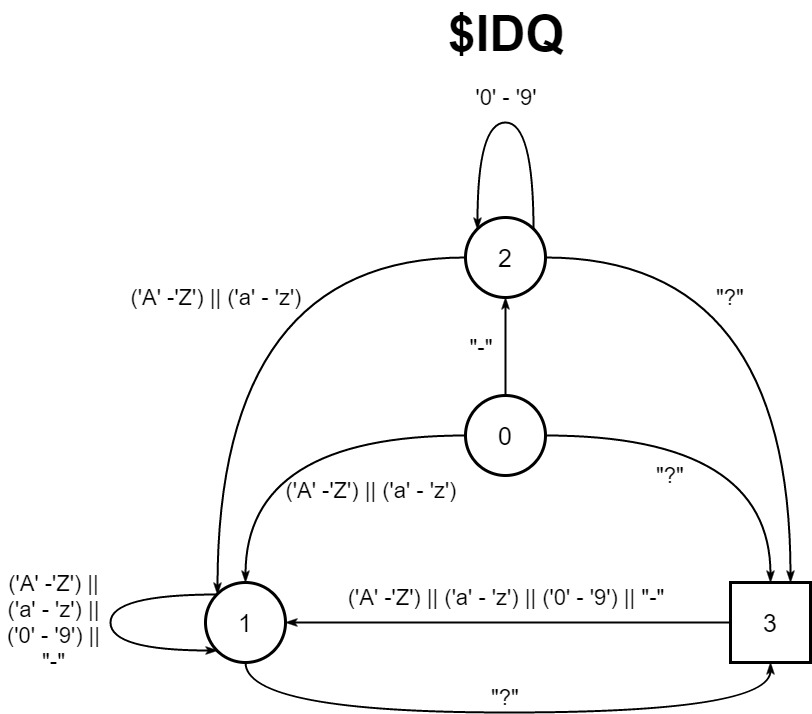
**[10]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **<+|->** | **<целая часть>** | **<.>** | **<дробная часть>** | **<e|E>** | **<+|->** | **<порядок>** |
| **\*** | **\*** | **+** | **+** | **+** | **+** | **+** |

****

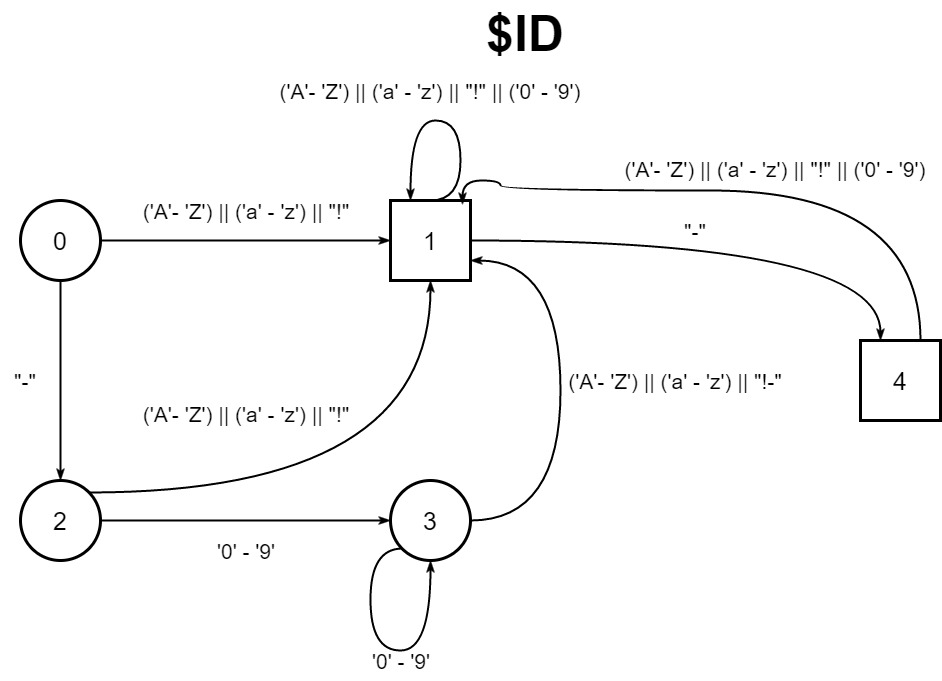
**$idq**

**[4] Знаки ‘?‘ не могут следовать друг за другом подряд**



**$id**

**[6] дополнительных ограничений нет**

****

**/\* $a15 \*/**

**#include "code-gen.h"**

**using namespace std;**

**int tCG::p01(){ // S -> PROG**

**string header ="/\* " + lex.Authentication()+" \*/\n";**

**header += "#include \"mlisp.h\"\n";**

**header += declarations;**

**header += "//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n";**

**S1->obj = header + S1->obj;**

**return 0;}**

**int tCG::p02(){ // CALC -> E1**

**S1->obj = " display(" + S1->obj + "); newline();\n";**

**return 0;**

**}**

**int tCG::p03(){ // CALC -> BOOL**

**S1->obj = " display(" + S1->obj + "); newline();\n";**

**return 0;}**

**int tCG::p04(){ // CALC -> STR**

**S1->obj = " display(" + S1->obj + "); newline();\n";**

**return 0;}**

**int tCG::p05(){ // CALC -> SET**

**return 0;}**

**int tCG::p06(){ // CALC -> DISP**

**return 0;}**

**int tCG::p07(){ // DISP -> ( display E1 )**

**S1->obj = " display(" + S3->obj + ")";**

**return 0;}**

**int tCG::p08(){ // DISP -> ( display BOOL )**

**S1->obj = " display(" + S3->obj + ")";**

**return 0;}**

**int tCG::p09(){ // DISP -> ( display STR )**

**S1->obj = " display(" + S3->obj + ")";**

**return 0;}**

**int tCG::p10(){ // DISP -> ( newline )**

**S1->obj += " newline()";**

**return 0;}**

**int tCG::p11(){ // PRED -> HPRED BOOL )**

**S1->obj += S2->obj + ";\n"+**

**"}}\n";**

**return 0;}**

**int tCG::p12(){ // HPRED -> PDPAR )**

**S1->obj += ")";**

**declarations += S1->obj + ";\n";**

**S1->obj += "\n{{\n return ";**

**return 0;}**

**int tCG::p13(){ // PDPAR -> ( define ( $idq**

**S1->obj = "bool " + decor(S4->name) + "(";**

**S1->count = 0;**

**return 0;}**

**int tCG::p14(){ // PDPAR -> PDPAR $id**

**if (S1->count > 0)**

**S1->obj += ", ";**

**S1->obj += "double " + decor(S2->name);**

**++S1->count;**

**return 0;}**

**int tCG::p15(){ // CPROC -> HCPROC )**

**S1->obj += ")";**

**return 0;}**

**int tCG::p16(){ // HCPROC -> ( $id**

**S1->obj = decor(S2->name) + "(";**

**S1->count = 0;**

**return 0;}**

**int tCG::p17(){ // HCPROC -> HCPROC E**

**if (S1->count > 0)**

**S1->obj += ", ";**

**S1->obj += S2->obj;**

**++S1->count;**

**return 0;}**

**int tCG::p18(){ // CPRED -> HCPRED )**

**S1->obj += ")";**

**return 0;}**

**int tCG::p19(){ // HCPRED -> ( $idq**

**S1->obj = decor(S2->name) + "(";**

**S1->count = 0;**

**return 0;}**

**int tCG::p20(){ // HCPRED -> HCPRED E**

**if (S1->count > 0)**

**S1->obj += ", ";**

**S1->obj += S2->obj;**

**++S1->count;**

**return 0;}**

**int tCG::p21(){ // E -> $id**

**S1->obj = decor(S1->name);**

**return 0;}**

**int tCG::p22(){ // E -> $zero**

**S1->obj = S1->name;**

**return 0;}**

**int tCG::p23(){ // E -> ADD**

**return 0;}**

**int tCG::p24(){ // E -> SUB**

**return 0;}**

**int tCG::p25(){ // E -> DIV**

**return 0;}**

**int tCG::p26(){ // E -> MUL**

**return 0;}**

**int tCG::p27(){ // E -> COND**

**return 0;}**

**int tCG::p28(){ // E -> CPROC**

**return 0;}**

**int tCG::p29(){ // ADD -> HADD E1 )**

**S1->obj += S2->obj + ")";**

**return 0;}**

**int tCG::p30(){ // HADD -> ( +**

**S1->obj = "(";**

**return 0;}**

**int tCG::p31(){ // HADD -> HADD E1**

**S1->obj += S2->obj + " + ";**

**return 0;}**

**int tCG::p32(){ // MUL -> HMUL E1 )**

**S1->obj += S2->obj + ")";**

**return 0;}**

**int tCG::p33(){ // HMUL -> ( \***

**S1->obj = "(";**

**return 0;}**

**int tCG::p34(){ // HMUL -> HMUL E1**

**S1->obj += S2->obj + " \* ";**

**return 0;}**

**int tCG::p35(){ // COND -> HCOND CLAUS )**

**S1->obj += S2->obj + "false";**

**return 0;}**

**int tCG::p36(){ // HCOND -> ( cond CLAUS**

**S1->obj += S3->obj;**

**return 0;}**

**int tCG::p37(){ // CLAUS -> HCLAUS E1 )**

**S1->obj += S2->obj + " : \n\t";**

**return 0;}**

**int tCG::p38(){ // HCLAUS -> ( BOOL**

**S1->obj = S2->obj + " ? ";**

**return 0;}**

**int tCG::p39(){ // STR -> $str**

**S1->obj = S1->name;**

**return 0;}**

**int tCG::p40(){ // STR -> SIF**

**return 0;}**

**int tCG::p41(){ // SIF -> SIFTRUE STR )**

**S1->obj += "\n\t\t:" + S2->obj;**

**return 0;}**

**int tCG::p42(){ //SIFTRUE -> HIF STR**

**S1->obj += S2->obj;**

**return 0;}**

**int tCG::p43(){ // HIF -> ( if BOOL**

**S1->obj = S3->obj + " ? ";**

**return 0;}**

**int tCG::p44(){ // E1 -> E**

**return 0;}**

**int tCG::p45(){ // SET -> HSET E1 )**

**S1->obj += S2->obj + ";\n";**

**return 0;}**

**int tCG::p46(){ // HSET -> ( set! $id**

**S1->obj = " " + decor(S3->name) + " = ";**

**return 0;}**

**int tCG::p47(){ // CONST -> $int**

**S1->obj = S1->name;**

**return 0;}**

**int tCG::p48(){ // CONST -> $float**

**S1->obj = S1->name;**

**return 0;}**

**int tCG::p49(){ // PROC -> PRBODY )**

**S1->obj += ";\n}}\n";**

**return 0;}**

**int tCG::p50(){ // PRBODY -> HPROC E**

**S1->obj += " return " + S2->obj;**

**return 0;}**

**int tCG::p51(){ // PRBODY -> HPROC CONST**

**S1->obj += " return " + S2->obj;**

**return 0;}**

**int tCG::p52(){ // HPROC -> PCPAR )**

**S1->obj += ")";**

**declarations += S1->obj + ";\n";**

**S1->obj += "\n{{\n";**

**S1->count = 0;**

**return 0;}**

**int tCG::p53(){ // HPROC -> HPROC SET**

**S1->obj += S2->obj;**

**return 0;}**

**int tCG::p54(){ // HPROC -> HPROC DISP**

**S1->obj += S2->obj + ";\n";**

**return 0;}**

**int tCG::p55(){ // PCPAR -> ( define ( $id**

**S1->obj = "double " + decor(S4->name) + "(";**

**S1->count = 0;**

**return 0;}**

**int tCG::p56(){ // PCPAR -> PCPAR $id**

**if (S1->count > 0)**

**S1->obj += ", ";**

**S1->obj += "double " + decor(S2->name);**

**++S1->count;**

**return 0;}**

**int tCG::p57(){ // DEF -> PRED**

**return 0;}**

**int tCG::p58(){ // DEF -> PROC**

**return 0;}**

**int tCG::p59(){ // DEFS -> DEF**

**return 0;}**

**int tCG::p60(){ // DEFS -> DEFS DEF**

**S1->obj += S2->obj;**

**return 0;}**

**int tCG::p61(){ // CALCS -> CALC**

**return 0;}**

**int tCG::p62(){ // CALCS -> CALCS CALC**

**S1->obj += S2->obj;**

**return 0;}**

**int tCG::p63(){ // CALCS1 -> CALCS**

**return 0;}**

**int tCG::p64(){ // PROG -> CALCS1**

**S1->obj = "int main()\n{\n" + S1->obj + " return 0;\n}\n";**

**return 0;}**

**int tCG::p65(){ // PROG -> DEFS**

**S1->obj += "int main()\n{\n display(\"No calculations!\\n\");\n return 0;\n}\n";**

**return 0;}**

**int tCG::p66(){ // PROG -> DEFS CALCS1**

**S1->obj += "int main()\n{\n" + S2->obj + " return 0;\n}\n";**

**return 0;}**

**int tCG::p67(){ // SUB -> HSUB E1 )**

**if (S1->count == 1)**

**S1->obj += "-" + S2->obj + ")";**

**else**

**S1->obj += S2->obj + ")";**

**return 0;}**

**int tCG::p68(){ // HSUB -> ( - E1**

**S1->obj += "(" + S3->obj + " - ";**

**return 0;}**

**int tCG::p69(){ // DIV -> ( / E1 )**

**S1->obj = "((double)";**

**S1->obj += "1 / " + S3->obj + ")";**

**return 0;}**

**int tCG::p70(){ // BOOL -> $bool**

**S1->obj = (S1->name == "#t" ? "true" : "false");**

**return 0;}**

**int tCG::p71(){ // BOOL -> CPRED**

**return 0;}**

**int tCG::p72(){ // BOOL -> REL**

**return 0;}**

**int tCG::p73(){ // BOOL -> OR**

**return 0;}**

**int tCG::p74(){ // BOOL -> ( not BOOL )**

**S1->obj += "!" + S3->obj;**

**return 0;}**

**int tCG::p75(){ // HREL -> ( > E**

**S1->obj = "(" + S3->obj + " > ";**

**return 0;}**

**int tCG::p76(){ // HREL -> ( = E**

**S1->obj = "(" + S3->obj + " == ";**

**return 0;}**

**int tCG::p77(){ // OR -> HOR BOOL )**

**S1->obj += S2->obj + ")";**

**return 0;}**

**int tCG::p78(){ // HOR -> ( or**

**S1->obj = "(";**

**return 0;}**

**int tCG::p79(){ // HOR -> HOR BOOL**

**S1->obj += S2->obj + " || ";**

**return 0;}**

**int tCG::p80(){ // REL -> HREL E1 )**

**S1->obj += S2->obj + ")";**

**return 0;}**

**//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**int tCG::p81(){return 0;} int tCG::p82(){return 0;}**

**int tCG::p83(){return 0;} int tCG::p84(){return 0;}**

**int tCG::p85(){return 0;} int tCG::p86(){return 0;}**

**int tCG::p87(){return 0;} int tCG::p88(){return 0;}**

**int tCG::p89(){return 0;} int tCG::p90(){return 0;}**

**int tCG::p91(){return 0;} int tCG::p92(){return 0;}**

**int tCG::p93(){return 0;} int tCG::p94(){return 0;}**

**int tCG::p95(){return 0;} int tCG::p96(){return 0;}**

**int tCG::p97(){return 0;} int tCG::p98(){return 0;}**

**int tCG::p99(){return 0;} int tCG::p100(){return 0;}**

**int tCG::p101(){return 0;} int tCG::p102(){return 0;}**

**int tCG::p103(){return 0;} int tCG::p104(){return 0;}**

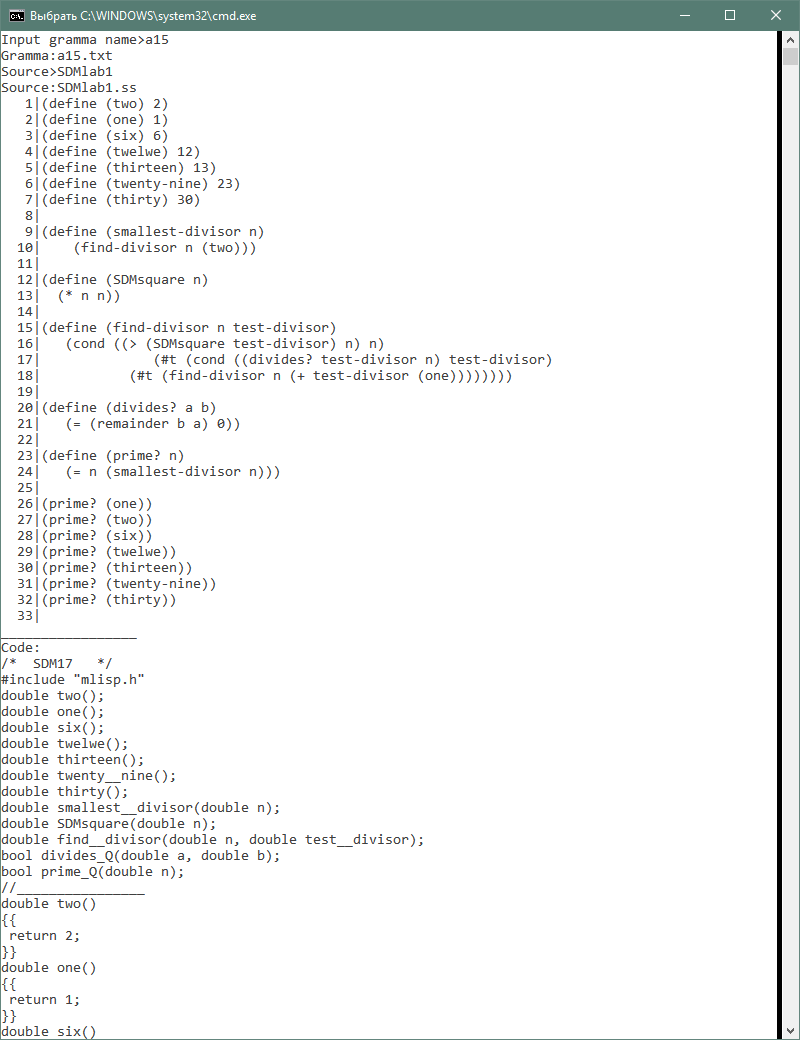
**int tCG::p105(){return 0;} int tCG::p106(){return 0;}**

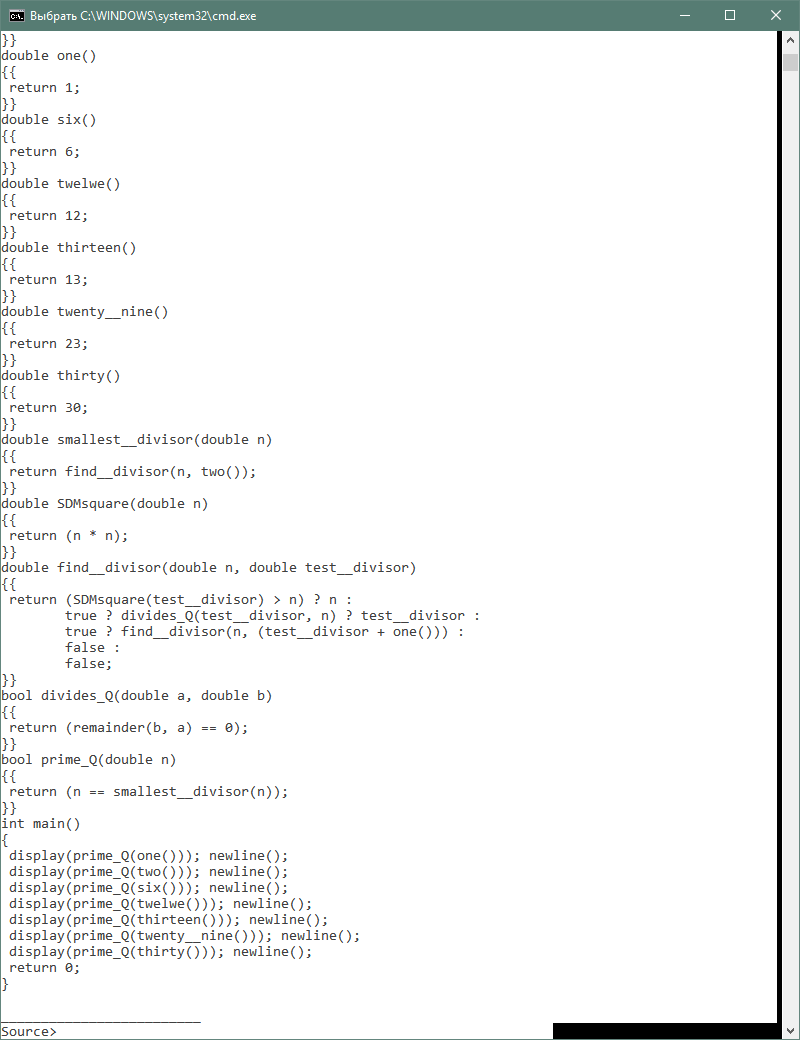
**int tCG::p107(){return 0;} int tCG::p108(){return 0;}**

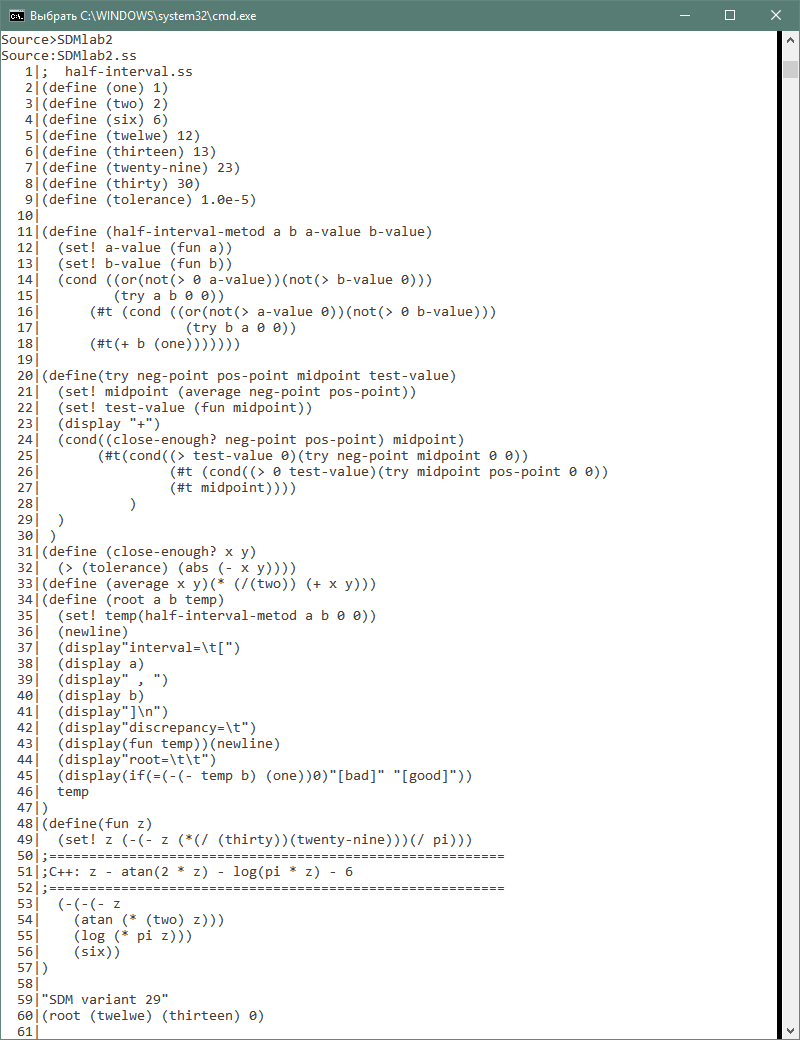
**int tCG::p109(){return 0;} int tCG::p110(){return 0;}**

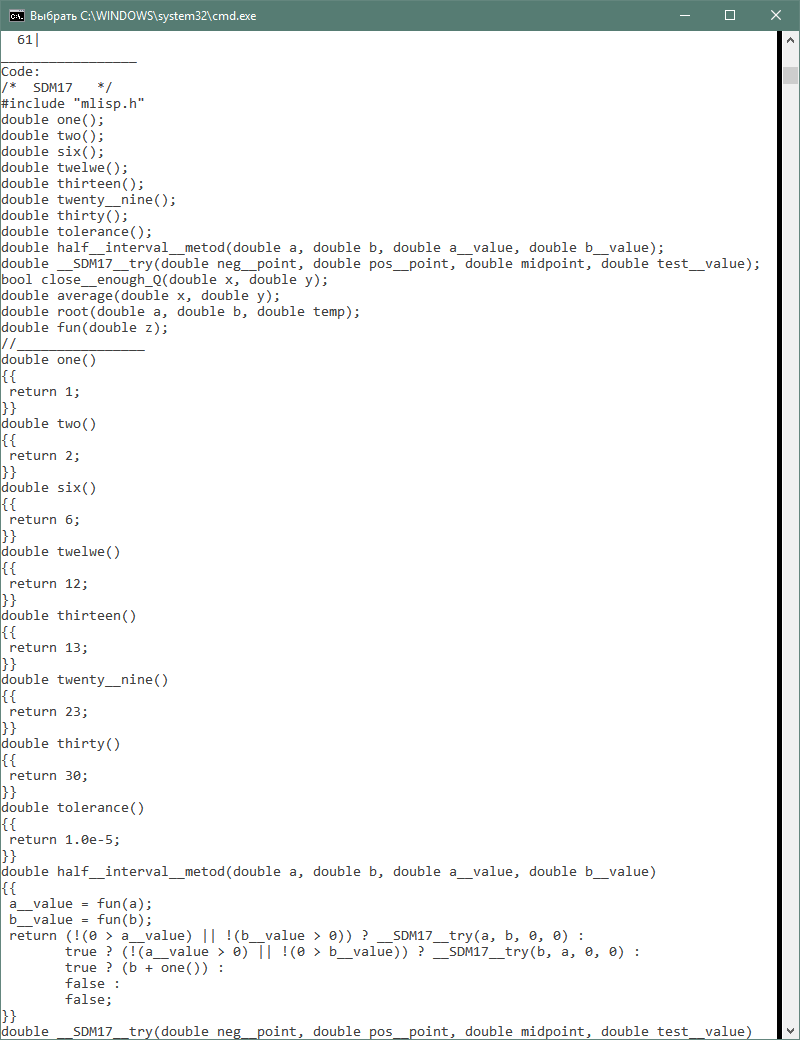
**…………………………………………………………………………………………………………………………………….**

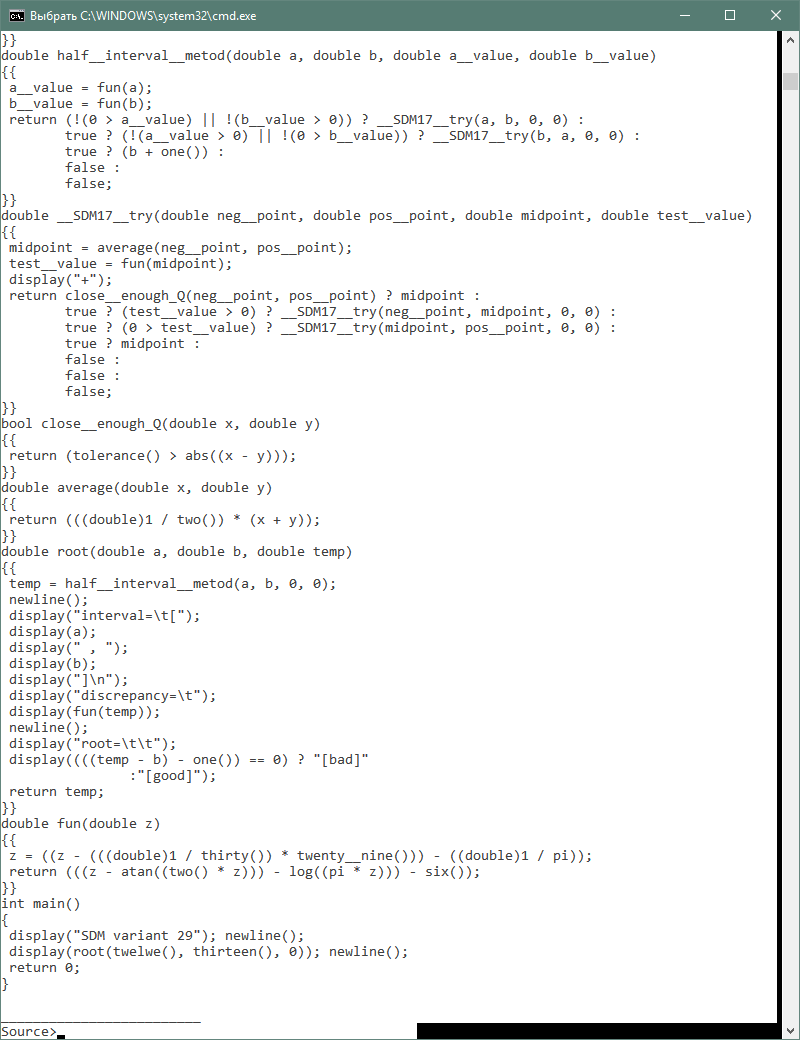
**Результат работы программы:**

1. **Трансляция контрольных задач:**
   * **SDMlab1.ss → SDMlab1.cpp (prime)**



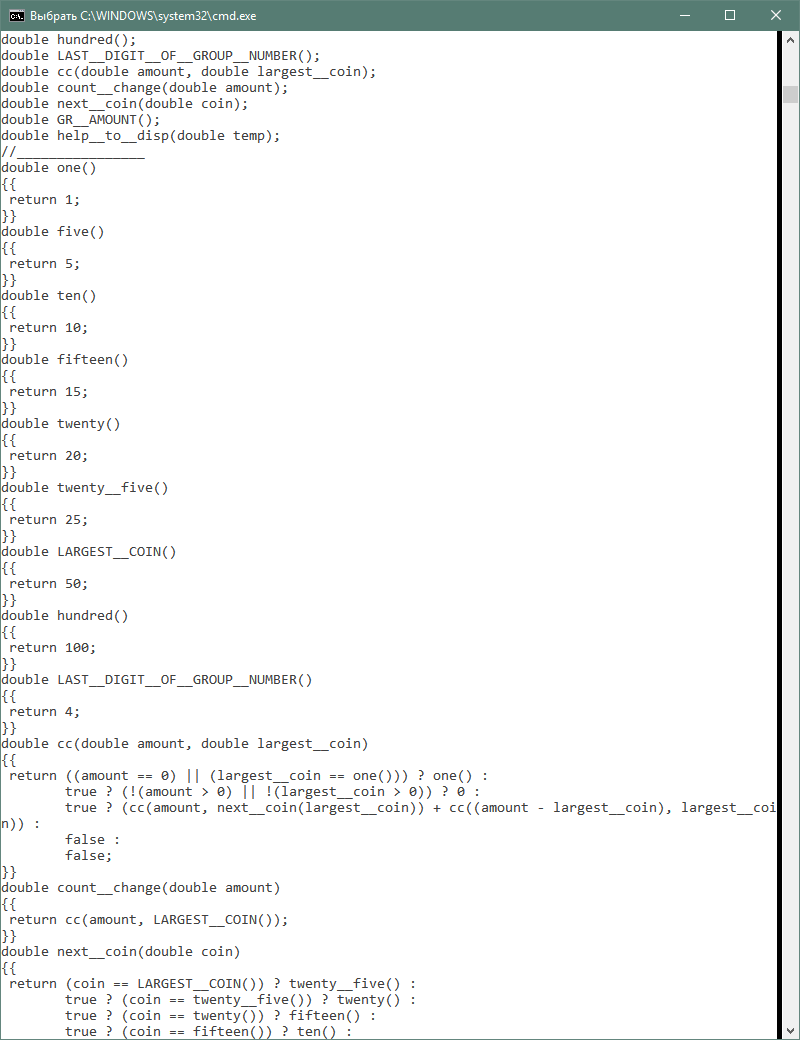
* + **SDMlab2.ss → SDMlab2.cpp (half-interval)**

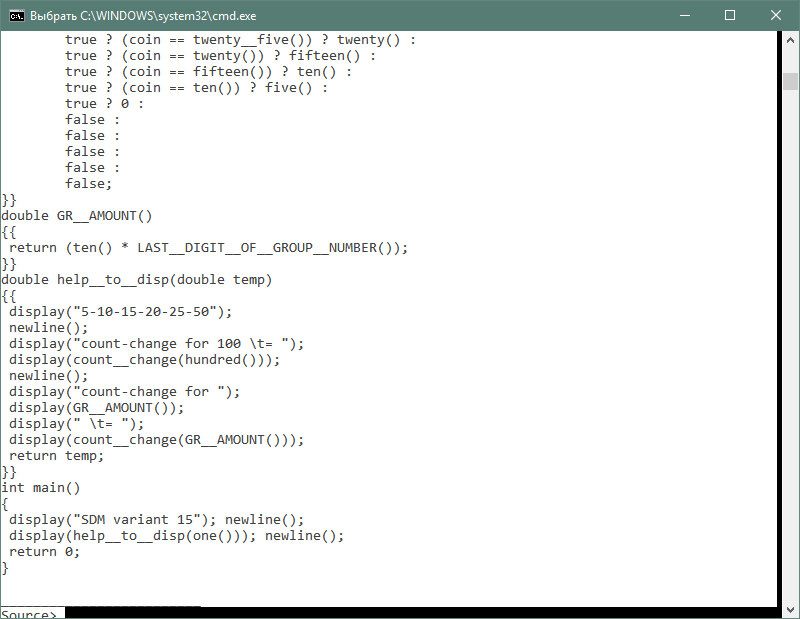




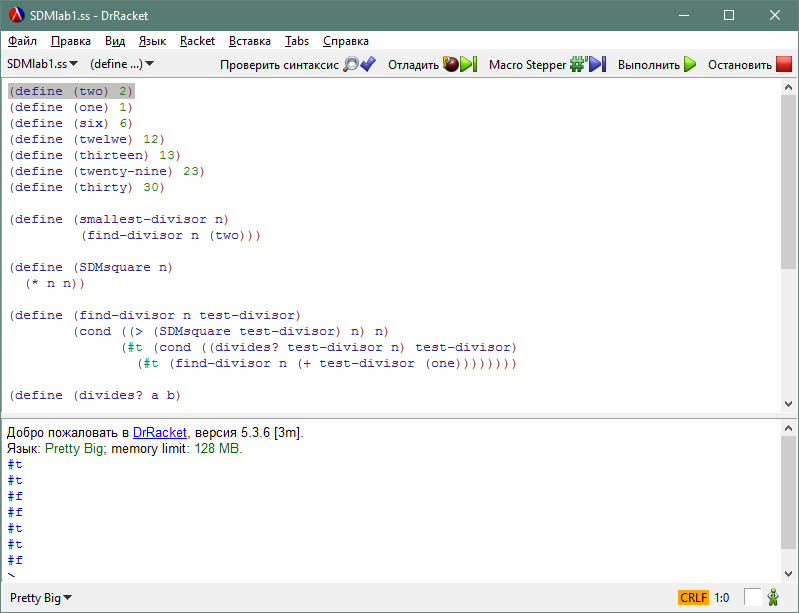
* + **SDMlab3.ss → SDMlab3.cpp (coin)**

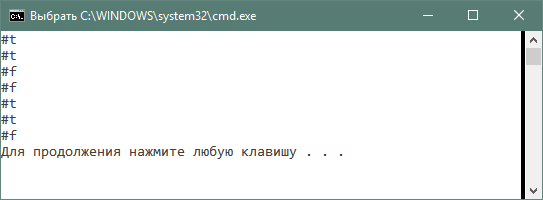




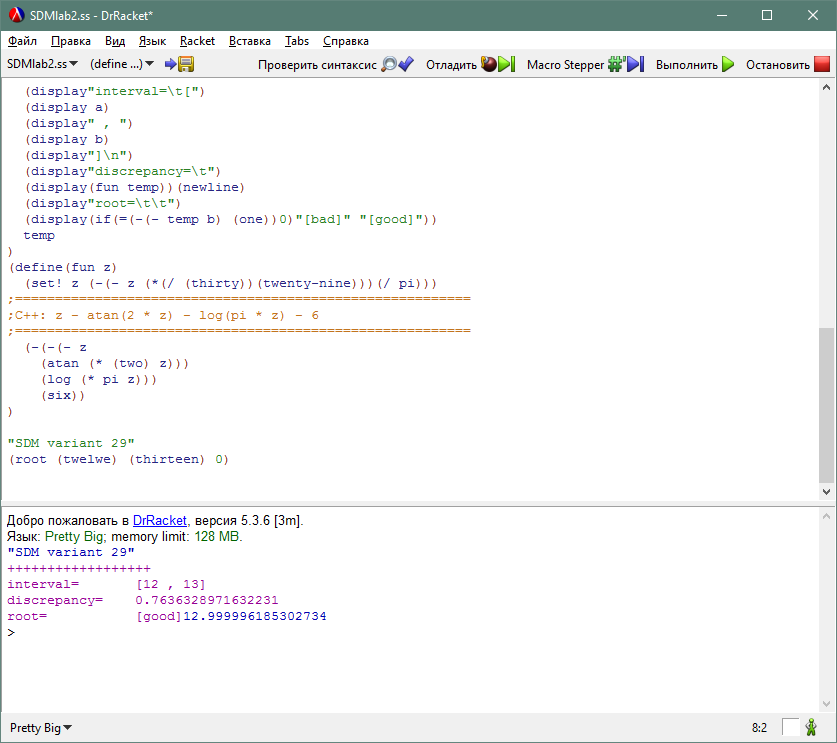


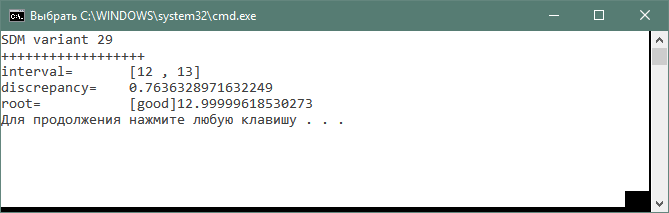
1. **Протоколы запуска контрольных задач:**
   * **SDMlab1.ss → SDMlab1.cpp (prime)**



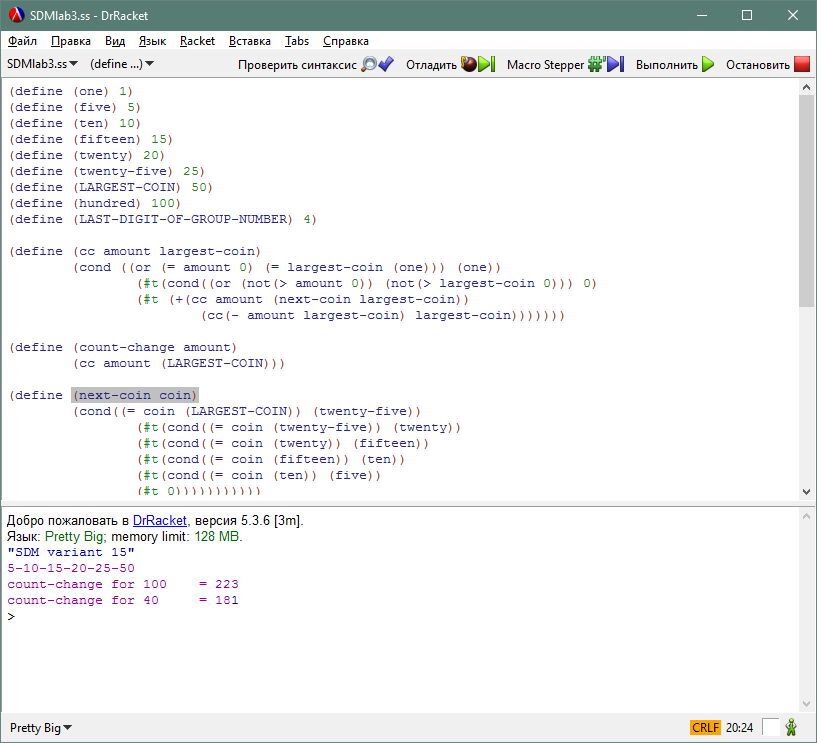


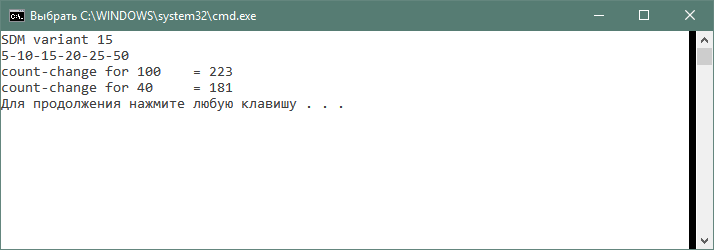
* + **SDMlab2.ss → SDMlab2.cpp (half-interval)**





* + **SDMlab3.ss → SDMlab3.cpp (coin)**





**………………………………………………………………………………………………………………………………………….**

**Выводы:**

**Данная курсовая работа удивила меня! Я вроде бы поступал на факультет прикладной математики и физики, а занимаюсь переводом языков! Это удивление приятное, ведь за этим переводом кроется логика и математика.**

**Идея курсовой заключается в разработке синтаксически управляемого транслятора из языка ЛИСП в язык C++. Звучит уже интересно! Но в то же время работа оказалась довольно непростой…**

**Для начала мне потребовалось установить особенности грамматики своего варианта с помощью программы parser.cpp. С этим я справился с легкостью! Однако дальше начались трудности. Необходимо было переписать код первых трех лабораторных под эти особенности.**

**В течение нескольких дней я пытался разобраться, как это сделать, но тут и там выскакивали нюансы. Например, долго не мог понять, как же мне быть с отсутствием в моем варианте определяемых переменных. Но с помощью преподавателя, моих товарищей из группы, а также с помощью законов де Моргана, я во всем разобрался и лабораторные были переписаны.**

**Однако и это не было самым сложным для меня. Далее нужно было описать продукции в файле code-gen.cpp. Было трудно, потому что не сразу прочувствовал, что от меня требуется и как это реализовывать. Но, когда осознал, меня уже было не остановить!**

**По завершению у меня получился рабочий транслятор. При запуске программ на разных языках, их выводы эквиваленты, что не может не радовать! Значит все верно.**

**Хочется отметить, что при написании работы мной были использованы почти все знания, полученные в ходе курса. Вообще весь курс был очень занимателен и интересен для меня. Задания были разнообразны и требовали навыков, полученных в предыдущих работах.**

**Эта работа является плодом накопленных за семестр знаний, на мой взгляд, получилось неплохо (работает же!). Я рад, что изучил данный курс.**