Parsing and Translating Binary Expressions

May 23, 2018

Masenya L 25480774
Mokgotla S 26428059
Ngoasheng D 26672995
Group Xliser
CISM 314
LAB 3

1 To create a compiler that pass and translate general expressions that involve Addition and Subtraction.

Firstly we downloaded three applications which are Sublime Text 3 where we are going to type the code using pascal programming language, DOSBOX to compile our pascal code and lastly cmdir to submit our work at GitHub.

Sublime Text is a proprietary cross-platform source code editor with python application programming interface (API). It natively supports many programming languages and markup languages. DOSBox is a full CPU emulator, capable of running DOS programs that require the CPU to be in real mode or protected mode. Since DOSBox can emulate its CPU by interpretation, the environment it emulates is completely independent of the host CPU.

Below are the procedures we took in Parsing and Translating Binary Expressions, from writting the code in sublime, saving the code in C drive under TPascal4 folder since we are using pascal environment. We then went to DOSBox to mount C drive so that we can load, compiler and run our code in DOSBox since is a command line simulator and C drive is not, so we want DOSBox to recognize C Drive so that it can compiler the code.

Finally we checked if the compiled code works by adding two numbers and subtracting two numbers the it gave us results that addition and subtraction is allowed, Again we multiplied the two numbers the it gave us runtime error, because we did not include the multiplication procedure.

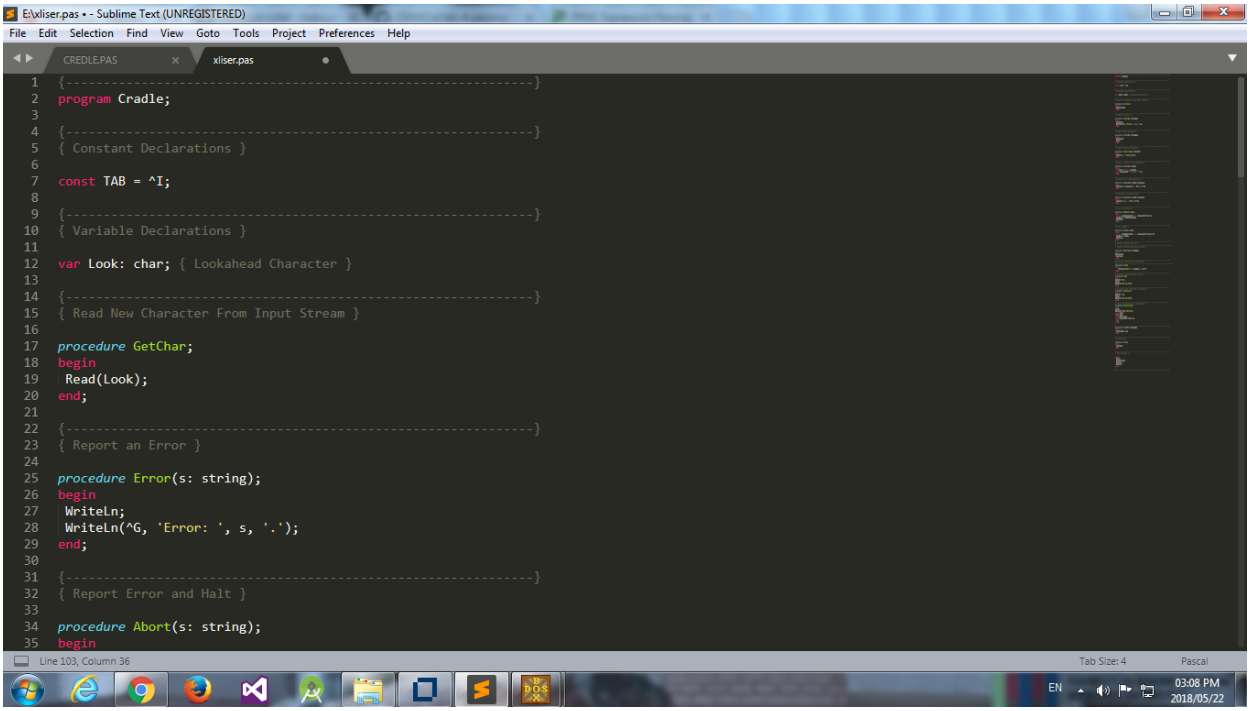
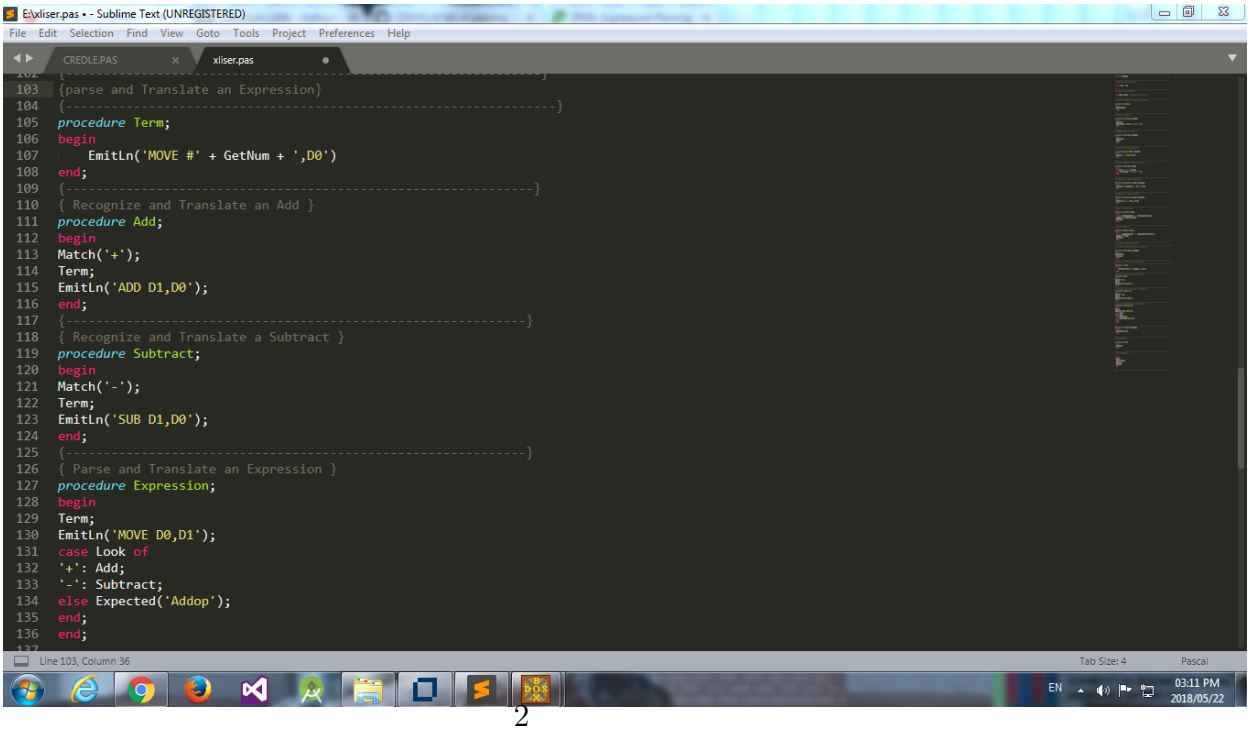

E:\xliser.pas • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
CREDLE.PAS
xliser.pas
{--}
program Cradle;
{--}
{ Constant Declarations }
const TAB = ^I;
{--}
{ Variable Declarations }
var Look: char; { Lookahead Character }
{--}
{ Read New Character From Input Stream }
procedure GetChar;
begin
Read(Look);
end;
{--}
{ Report an Error }
procedure Error(s: string);
begin
WriteLn;
WriteLn(^G, 'Error: ', s, '.');
end;
{--}
{ Report Error and Halt }
procedure Abort(s: string);
begin
Line 103, Column 36
Tab Size: 4
Pascal
EN
03:08 PM
2018/05/22

Figure 1:


E:\xliser.pas • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
CREDLE.PAS
xliser.pas
{parse and Translate an Expression}
{--}
procedure Term;
begin
EmitLn('MOVE #' + GetNum + ',D0')
end;
{--}
{ Recognize and Translate an Add }
procedure Add;
begin
Match('+');
Term;
EmitLn('ADD D1,D0');
end;
{--}
{ Recognize and Translate a Subtract }
procedure Subtract;
begin
Match('-');
Term;
EmitLn('SUB D1,D0');
end;
{--}
{ Parse and Translate an Expression }
procedure Expression;
begin
Term;
EmitLn('MOVE D0,D1');
case Look of
'+': Add;
'-': Subtract;
else Expected('Addop');
end;
end;
Line 103, Column 36
Tab Size: 4
Pascal
EN
03:11 PM
2018/05/22

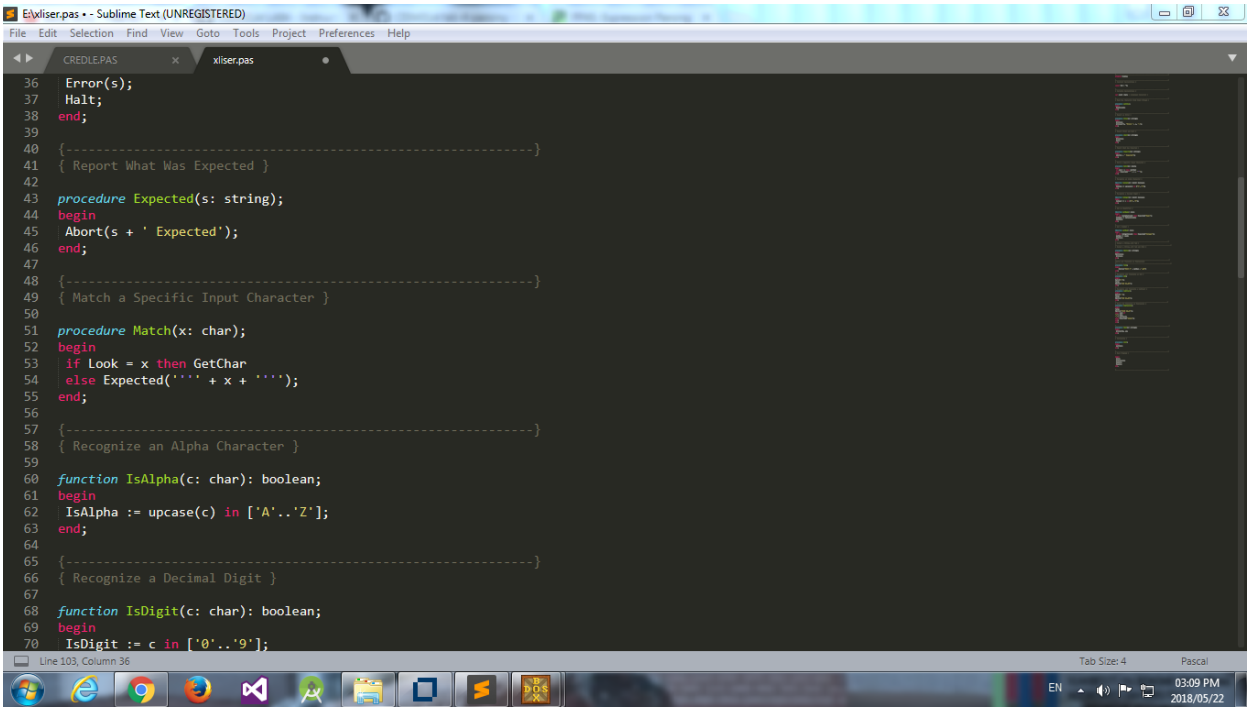
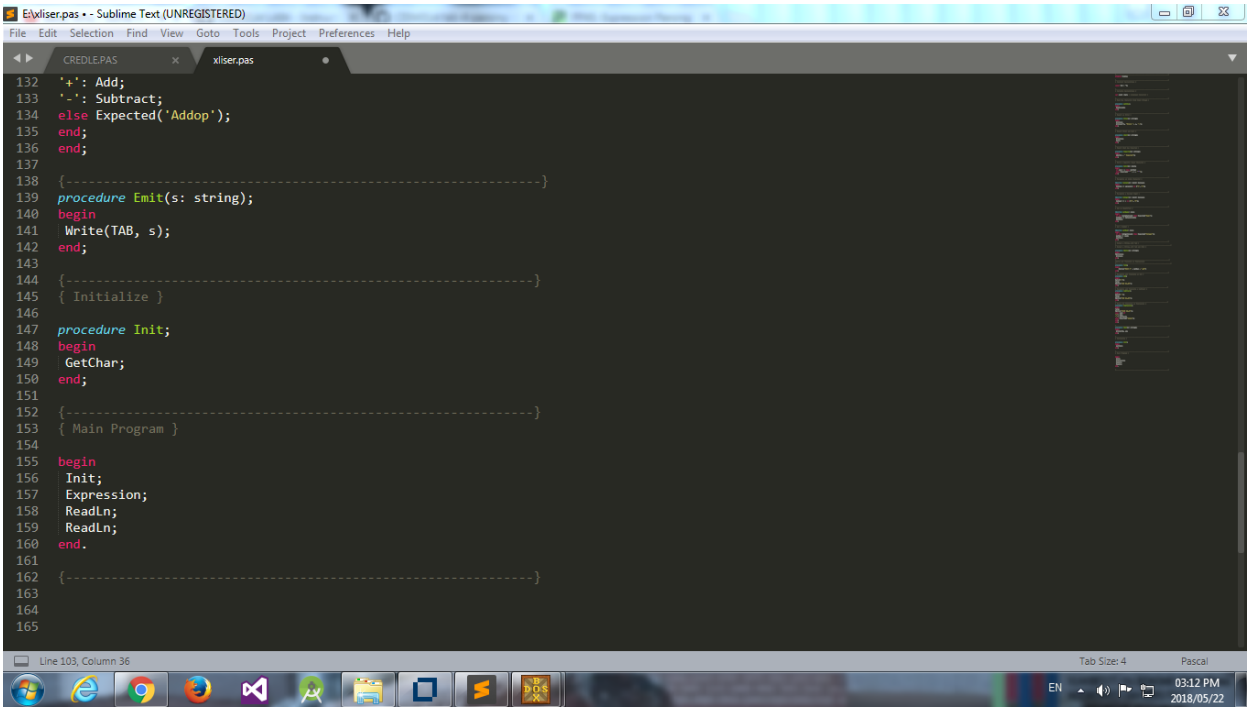
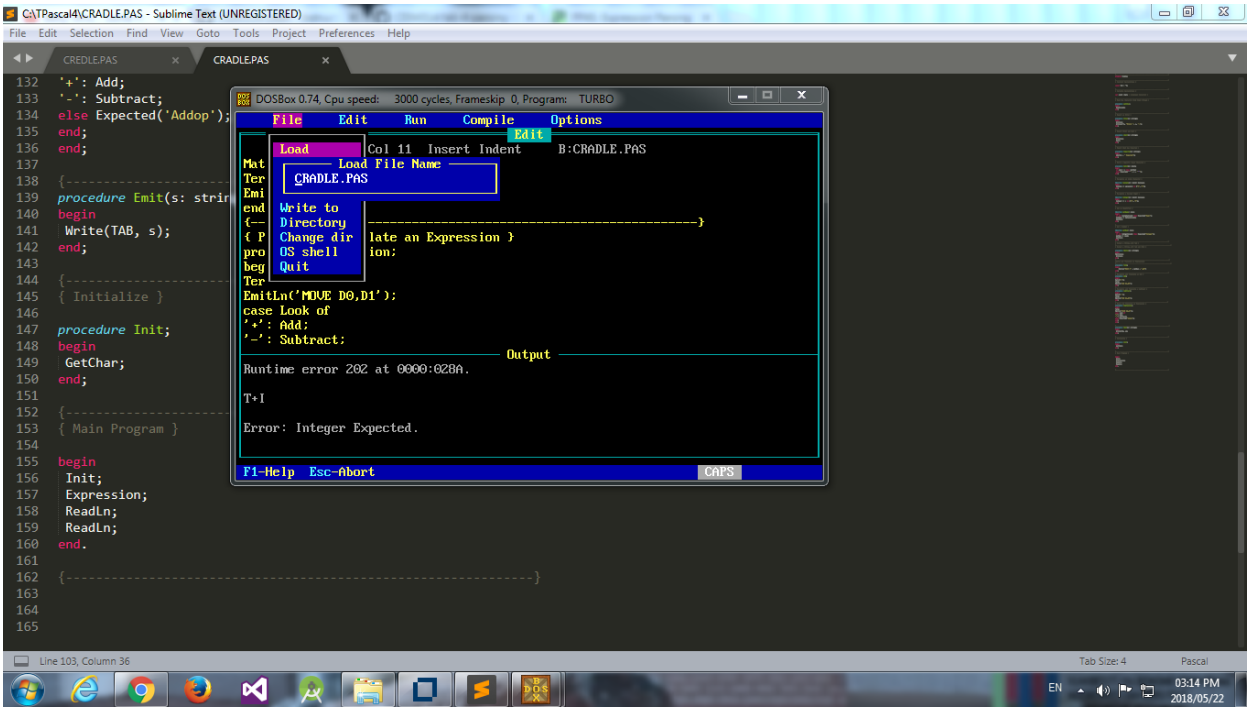
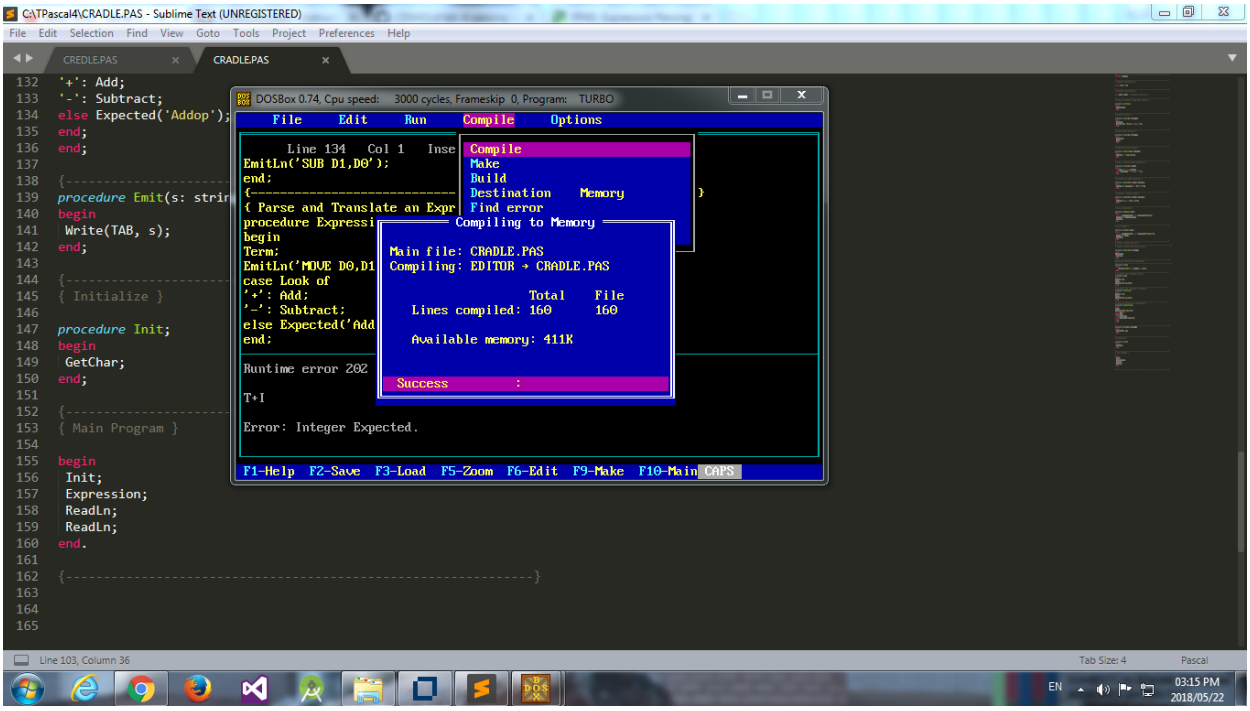
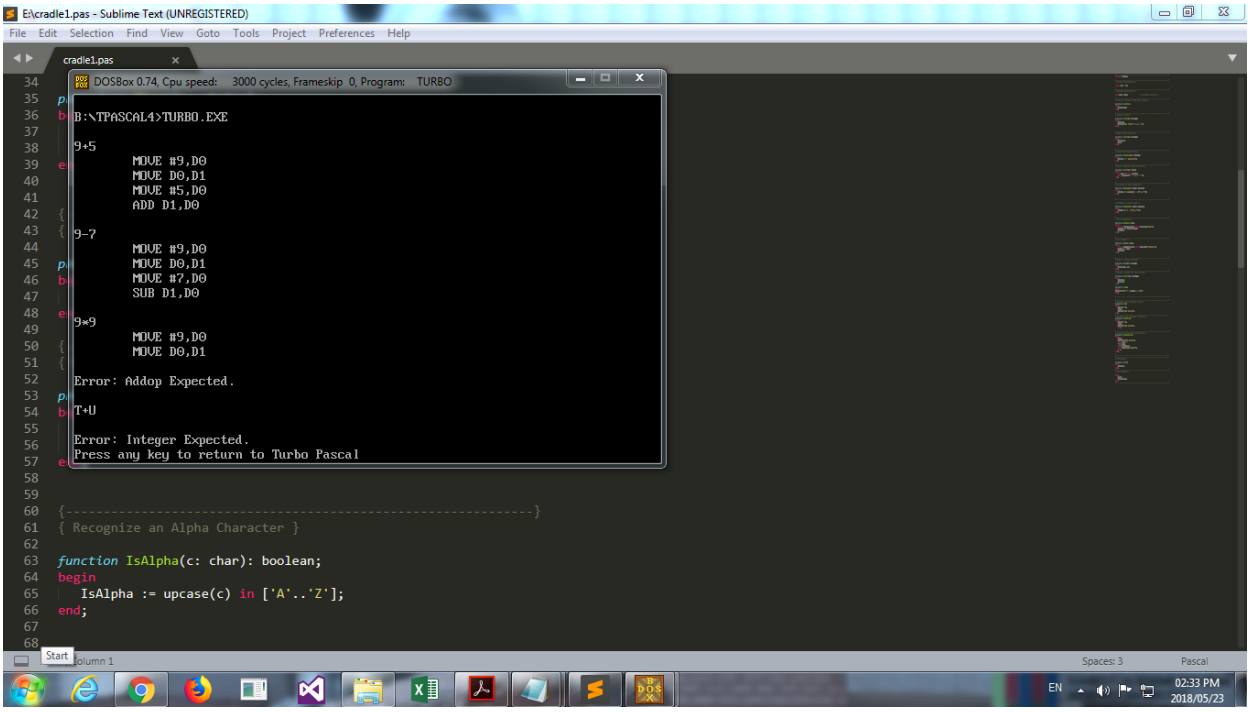

E:\xliser.pas • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
CREDLE.PAS
xliser.pas
Error(s);
Halt;
end;
{ Report What Was Expected }
procedure Expected(s: string);
begin
Abort(s + ' Expected');
end;
{ Match a Specific Input Character }
procedure Match(x: char);
begin
if Look = x then GetChar
else Expected('''' + x + '''');
end;
{ Recognize an Alpha Character }
function IsAlpha(c: char): boolean;
begin
IsAlpha := upcase(c) in ['A'..'Z'];
end;
{ Recognize a Decimal Digit }
function IsDigit(c: char): boolean;
begin
IsDigit := c in ['0'..'9'];
Line 103, Column 36
Tab Size: 4
Pascal
EN
03:09 PM
2018/05/22

Figure 2:


E:\xliser.pas • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
CREDLE.PAS
xliser.pas
'+': Add;
'-': Subtract;
else Expected('Addop');
end;
end;
{---}
procedure Emit(s: string);
begin
Write(TAB, s);
end;
{---}
{ Initialize }
procedure Init;
begin
GetChar;
end;
{---}
{ Main Program }
begin
Init;
Expression;
ReadLn;
ReadLn;
end.
{---}
Line 103, Column 36
Tab Size: 4
Pascal
EN
03:12 PM
2018/05/22


C:\TPascal4\CRADLE.PAS - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
CREDLE.PAS
CRADLE.PAS
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TURBO
File Edit Run Compile Options
Edit
Load
Col 11 Insert Indent B:CRADLE.PAS
Load File Name
CRADLE.PAS
Write to
Directory
Change dir
OS shell
Quit
late an Expression }
ion;
EmitLn('MOVE D0,D1');
case Look of
'+': Add;
'-': Subtract;
Output
Runtime error 202 at 0000:028A.
T+I
Error: Integer Expected.
F1-Help Esc-Abort
CAPS
Line 103, Column 36
Tab Size: 4
Pascal
EN
03:14 PM
2018/05/22


CATPascal4\CRADLE.PAS - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
CREDLE.PAS
CRADLE.PAS
'+': Add;
'-': Subtract;
else Expected('Addop');
end;
end;
procedure Emit(s: strir
begin
Write(TAB, s);
end;
{ Initialize }
procedure Init;
begin
GetChar;
end;
{ Main Program }
begin
Init;
Expression;
ReadLn;
ReadLn;
end.
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TURBO
File Edit Run Compile Options
Line 134 Col 1 Inse
EmitLn('SUB D1,D0');
end;
{ Parse and Translate an Expr
procedure Expressi
begin
Term;
EmitLn('MOVE D0,D1
case Look of
'+': Add;
'-': Subtract;
else Expected('Add
end;
Runtime error 202
T+I
Error: Integer Expected.
Compile
Make
Build
Destination Memory
Find error
Compiling to Memory
Main file: CRADLE.PAS
Compiling: EDITOR → CRADLE.PAS
Total File
Lines compiled: 160 160
Available memory: 411K
Success :
F1-Help F2-Save F3-Load F5-Zoom F6-Edit F9-Make F10-Main CAPS
Line 103, Column 36
Tab Size: 4
Pascal
EN
03:15 PM
2018/05/22


E:\cradle1.pas - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
cradle1.pas
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TURBO
B:\TPASCAL4>TURBO.EXE
9+5
MOVE #9,D0
MOVE D0,D1
MOVE #5,D0
ADD D1,D0
9-7
MOVE #9,D0
MOVE D0,D1
MOVE #7,D0
SUB D1,D0
9*9
MOVE #9,D0
MOVE D0,D1
Error: Addop Expected.
T+U
Error: Integer Expected.
Press any key to return to Turbo Pascal
{ Recognize an Alpha Character }
function IsAlpha(c: char): boolean;
begin
IsAlpha := upcase(c) in ['A'..'Z'];
end;
Start
Spaces: 3
Pascal
EN
02:33 PM
2018/05/23