

# DATA STRUCTURE, JAVA, C++, C, ASSEMBLY 8086 codes

Tuesday, April 14, 2015

## Intermediate code generator

```
%{
// yacc program
#include <stdio.h>
#include <stdlib.h>
}%

%token ID NUM IF THEN ELSE WHILE FOR LE GE EQ NE OR AND //value return by lex
program
%right "=" /* equal to right associative*/
/* All the other are left associative and down the line precedence increases */
%left OR AND
%left '>' '<' LE GE EQ NE //relational operators
%left '+' '-'
%left '*' '/'
%right UMINUS //unary minus
%left '!'

%%

/* This is the Grammar for the language */
/* S is start and it has three possibilities FOR, WHILE, IF ELSE */
/* E is variable */
S : FOR '(' E ';' '{lab1();} E {lab2();};' E {lab3();};' E ';' '{lab4();}
exit(0); //label generate for loop
| WHILE '{wlab1();} '(' E ')' '{wlab2();} E ';' '{wlab3();} //label generated
for while
| IF '(' E ')' '{ilab1();} THEN E ';' '{ilab2();} ELSE E ';' '{ilab3();}
//if else
;

/* E has following productions */
/* V is also a variable */
/* everytime number or identifier or arithmetic symbol encounters it pushes them
in to stack */
/* code_assign is function is call when assignment encounters*/
/* codegen is function is call when arithmetic operation encounters */
E : V '=' {push();} E {codegen_assign();}
| E '+' {push();} E {codegen();}
| E '-' {push();} E {codegen();}
| E '*' {push();} E {codegen();}
| E '/' {push();} E {codegen();}
| '(' E ')'
| '-' {push();} E {codegen_umin();} %prec UMINUS
| V
| NUM {push();}
;
V : ID {push();}
;

%%

#include "lex.yy.c"
#include <ctype.h>
char st[100][10];
int label[20];
int top=0;
char i_[2]="0"; /* for tracking count 0,1,2*/
char temp[2]="t"; /* t for three address code*/

int lno=0,ltop=0;
int lnum=0;
int start=1;

main()
{
```

### Label

8086 assembly

8086 assembly language program

Add all greater values to every node in a given BST

Addition and subtraction using Bitwise operators

Addition of two Algebraic functions in mathematics using Data structure C++ program

anagrams for given string

applet

Apriori in JAVA

assembly language graphics programming

assembly language programming

AVL tree in C++

bd tree

BFS

Bin sort algorithm and program

binary tree comparison

Binary Tree is BST or not c++

BINARY TREE Level Printing in c++

bitwise operator

block wise reverse of linked list

bouncing ball assembly language

bouncing ball assembly program in java

breadth first search graph

BST

bst or not

btree node deletion in C++

Bubble sort

c++

c++ code for finding anagrams

c++ program

calculate the value of infix expression using 1 stack in C++

calculate the value of infix expression using 2 stacks in C++

```

printf("Enter the expression/program:\n");
yyvsparse();
}

push()
{
    /* push it on the stack and increase the top count*/
    strcpy(st[++top],yytext);
}

codegen()
{
    strcpy(temp,"t");
    strcat(temp,i_);
    /* print the arithmetic expression statement*/
    printf("%s = %s %s %s\n",temp,st[top-2],st[top-1],st[top]);
    top-=2; /* decrement the top count by 2*/
    strcpy(st[top],temp);
    i_[0]++;
}

codegen_umin()
{
    strcpy(temp,"t");
    strcat(temp,i_);
    /* print the unary minus statement*/
    printf("%s = -%s\n",temp,st[top]);
    top--; /* decrement the top count by 1*/
    strcpy(st[top],temp);
    i_[0]++;
}

codegen_assign()
{
    /*print the assignment statement*/
    printf("%s = %s\n",st[top-2],st[top]);
    top-=2; /* decrement the top by 2*/
}

ilab1()
{
    lnum++;
    strcpy(temp,"t");
    strcat(temp,i_); // prints negation of operands
    printf("%s = not %s\n",temp,st[top]);
    //checks for condition and jumps to target statement
    printf("if %s goto L%d\n",temp,lnum);
    i_[0]++;
    label[++ltop]=lnum; //push lnum in array
}

ilab2()
{
    int x;
    lnum++;
    x=label[ltop--];
    printf("goto L%d\n",lnum); //prints jump label and target statement
    printf("L%d: \n",x);
    //push lnum in array
    label[++ltop]=lnum;
}

ilab3()
{
    int y;
    y=label[ltop--];
    //prints label
    printf("L%d: \n",y);
}

lab1()
{
    printf("L%d: \n",lno++);
}

lab2()
{
    strcpy(temp,"t");

```

cannon bal using appletl in java

check if a binary tree is sub tree of other binary tree

Check if All Leaf Nodes are at same Level in Binary tree

Check if Expression is Balanced or Not using Stack in Python

Check if two given strings are isomorphic

COMPARE 2 TREES in C++

computer networks

CONWAY's GAME OF LIFE USING BUTTON in JAVA

creation and deletion

cycle in graph

Data structure

depth first search in graph

detect cycle in graph

DFS

DFS and BFS of undirected graph

diameter of tree

digital clock in assembly language

Distance between two vertices in Graph

Doubly Linked List in Data Structure C++ code program

dynamic programming

Evaluating postfix expression using queue in C++

Evaluating the postfix expression using stack in C++

Find Itinerary from a given list of tickets

Find sum of nodes vertically for a given binary tree

first come first serve Operating system c code

Foldable binary tree

geeksforgeeks

Generic stack in C++

Given a N\*N square matrix

graph

graphics display in ASSEMBLY LANGUAGE

heapsort algorithm

Heapsort in Python

histogram implementation using applet

how to check if given tree is BST

Implementation of Bubble sort using Queue in C++

Implementation of linked list and all operations in C++

```

    strcat(temp,i_);
    printf("%s = not %s\n",temp,st[top]);
    printf("if %s goto L%d\n",temp,lno);
    i_[0]++;
    label[++ltop]=lno;
    lno++;
    printf("goto L%d\n",lno);
    label[++ltop]=lno;
    printf("L%d: \n",++lno);
}
lab3()
{
    int x;
    x=label[ltop--];
    printf("goto L%d \n",start);
    printf("L%d: \n",x);
}

wlab1()
{
    //prints while lebel 1
    printf("L%d: \n",lnum++);
}

wlab2()
{
    strcpy(temp,"t");
    strcat(temp,i_);
    printf("%s = not %s\n",temp,st[top]);
    printf("if %s goto L%d\n",temp,lnum);
    i_[0]++;
}

wlab3()
{
    printf("goto L%d \n",start);
    printf("L%d: \n",lnum);
}

lab4()
{
    int x;
    x=label[ltop--];
    //prints goto lebel
    printf("goto L%d \n",lno);
    printf("L%d: \n",x);
}

/*

We save yacc code with .y extension for example "xyz.y".

save the above program.*/

/*LEX PROGRAM
*/
ALPHA [A-Za-z]
DIGIT [0-9]
%%
while          return WHILE;
for            return FOR;
if             return IF;
then          return THEN;
else          return ELSE;
{ALPHA}{(ALPHA)|(DIGIT)}*    return ID;
{DIGIT}+      {yylval=atoi(yytext);    return NUM;}
"<="        return LE;
">="        return GE;
"=="        return EQ;
"!="        return NE;
"|"         return OR;
"&&"        return AND;
[ \t]       ;
\n          yyterminate();
.           return yytext[0];
%%

```

Implementation of stack containing queues as elements data structure in C++

implementation of stack using linked list in C++

in data strcuture

infix evaluation

Infix to Postfix using stack in C++

insertion sort in Python

int to string c++

intermediate code generation program in c

intermediate code generator in c++

inverted pendulum using applet in java

java

java programming

Kth smallest element

Kth smallest element in Binary search tree

Large number addition multiplication using Linked Lists in data structure c++

Left View of a Binary Tree

lex

linked list all operations in C++

longest prefix among strings

maximum width of BST

Merge a linked list into another linked list at alternate positions

Minimum Height in binary tree

Mirror Binary tree

mouse in maze

moving ball in java

moving ball inassembly language

moving car in assembly language 8086

moving object in assembly language

multithreading in JAVA

my laptop beeps when i start it

narasimha karumanchi

Node at K distance from root Binary Tree

node swap in linked list

Number of paths with exactly k coins

Point to next higher value node in a linked list with an arbitrary pointer

POTATO (circular queue) using LINKEDLIST using C++

print ancestor of a node in binary tree

print common nodes in BST

Projectile motion implementation using

save the above program with .l extension

now to execute the Intermediate code Generation

open your terminal and type these commands in order

```
lex filename.l lex.yy.c
```

Posted by Pawan Kumar at 5:03 PM

Labels: [c++](#), [Data structure](#), [geeksforgeeks](#), [intermediate code generation program in c](#), [intermediate code generator in c++](#), [lex](#), [yacc](#)

No comments yet



Add a comment as Avishek Roy

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

[applet java](#)

[python programming](#)

[queue](#)

[Queue implementation and operation in C++](#)

[Radix Bucket sorting algorithm and program in C++](#)

[Rearrange a given linked list in-place](#)

[redundancy check](#)

[Repetition of an element in array within k distance](#)

[return an array of its anti-diagonals](#)

[Reverse a String Word wise](#)

[Reverse an array without affecting special characters](#)

[reverse of linked list](#)

[similar tree](#)

[Solve beeping problem on startup in laptops](#)

[Solve beeping problem while booting in laptops](#)

[Sort a linked list that is sorted alternating ascending and descending orders](#)

[Sorting using two stacks in C++](#)

[Sprial in java](#)

[stack](#)

[stack implementation in C++](#)

[stack in python](#)

[Stack of stack in C++](#)

[string in assembly language](#)

[String to int c++](#)

[strings](#)

[Swap nodes in a linked list without swapping data](#)

[system programming code](#)

[TOWER OF HANOI USING APPLET in JAVA](#)

[Trailing zeroes in factorial](#)

[UBUNTU 15.05 update is out](#)

[ubuntu update](#)

[validation of infix expression in C++](#)

[Vector in C++ simple functions](#)

[vertical redundancy check](#)

[Vertical redundancy check computer networks error handling and error checking](#)

[vivid vervet launched ubuntu 15.05](#)

[Width of Binary Tree](#)

[write a prgram for to find x^n\(x to power of n\) using bitwise operators only](#)

**Contributors**

Pawan Samarwal

Pawan Kumar

yacc

**Blog Archive**

## ▼ 2015 (72)

► November (10)

► October (22)

► September (3)

► August (1)

► July (9)

## ▼ April (5)

UBUNTU 15.05 update is out

Vector in C++ simple functions

Vertical redundancy check computer networks error ...

Intermediate code generator

Solve beeping problem while booting in laptops

► February (22)

## ► 2014 (54)

**Translate**

Select Language ▼

Powered by  Google Translate**Google+ Badge****Populars****Intermediate code generator**

```
%{ // yacc program #include <stdio.h>
#include <stdlib.h> %} %token ID NUM
IF THEN ELSE WHILE FOR LE GE EQ
NE OR AND //valu...
```

**TSR DIGITAL CLOCK in ASSEMBLY LANGUAGE 8086**

I've been trying to write TSR (Terminate-Stay-Resident) programs (in general) in Assembly (16-bit) for MS-DOS. I've read through a...

**POSTFIX EXPRESSION solution using STACK in C++**

```
#include<iostream> using namespace
std; class stack { public: int top,sz; int
a[50]; stack (int c) { sz=c; t...
```

**color change of screen in 8086**

```
color change of screen if press 1
.model small .stack 64 .data clor db
10h .code main proc near mov
ax,@data mov ds,ax mov ax,3h...
```

**screensaver in SP using keyboard input 8086**

```
pawan macro mov ax,0600h mov
cx,0000h mov dx,184fh mov bh,a int
10h endm .model small .stack 64 .data
ctr db 10h tik db 'pa...
```

**UBUNTU 15.05 update is out**

VIVID VERVET has launched linux based operating system UBUNTU's new upgrade. Till now 14.04 was available as latest version of ubu...

**Moving CAR in ASSEMBLY LANGUAGE**

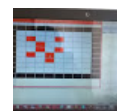
int 10h is a video BIOS interrupt that is older than dirt and cannot be accessed from user mode in current operating systems .model smal...

**File operation in Assembly language 8086**

```
;random access from file .model small
.stack 64 .data name1 label byte
maxlen db 5 actlen db ? record db 5
dup(' '), '$&#...
```

**AVL tree in C++**

AVL tree is a balanced tree. DEFINITION: An AVL tree is a binary search tree with the additional balance property that, for any node in ...

**CONWAY's GAME OF LIFE USING BUTTON in JAVA**

GAME OF LIFE is a very common problem in which you have to create a grid of button ( so called cells) . Som...

Awesome Inc. template. Template images by [molotovcocktail](#). Powered by [Blogger](#).