

3rd year cse lab programs

As per the anna university regulations - 2004, cs 1356 compilers lab and cs 1355 graphics and multimedia lab programs will be available here... u can also request for prog to this mail id cse.achievers@gmail.com...will be published soon...

Contributors

[kannan - admin](#)

[cselab](#)

Blog Archive

▼ 2010 (15)

► February (3)

▼ January (12)

[projection of 3d image](#)

[CODE GENERATION](#)

[cohen sutherland line](#)

[clipping](#)

[bresenhams line](#)

[drawing algorithm](#)

[intermediate code](#)

[generation](#)

[DDA LINE Drawing](#)

[Algorithm](#)

[two dimensional](#)

[transformation](#)

[midpoint circle](#)

[algorithm](#)

[midpoint ellipse](#)

[algorithm](#)

[shift reduce parser](#)

[recursive descent parser](#)

[in c](#)

[lexical analyser in c](#)

THURSDAY, JANUARY 28, 2010

CODE GENERATION

DOWNLOAD THIS FILE :

PROGRAM:

```
#include"stdio.h"
#include"conio.h"
#include"string.h"
#include"stdlib.h"
struct quadraple
{
    int pos;
    char op;
    char arg1[5];
    char arg2[5];
    char result[5];
}quad[15];
int n=0;
void assignment(int);
void uminus(int );
void explore();
void codegen(char op[5],int);
char tuple[15][15];
int main(void)
{
    FILE *src;
    int nRetInd,i;
    char str[15];
    clrscr();
    src=fopen("int.txt","r");
    fscanf(src,"%s",str);
    while(!feof(src))
    {
        strepy(tuple[n++],str);
        fscanf(src,"%s",str);
    }
    printf("INPUT:\nIntermiate codes:\n");
    for(i=0;i< n;i++)
        printf("%s\n",tuple[i]);
    explore();
    getch();
    clrscr();
    printf("OUTPUT:\n");
    printf("Quadruple: \n");
    printf("pos\topr\targ1\targ2\tresult\n");
    for(i=0;i< n;i++)
        printf("\n%d\t%c\t%s\t%s\t%s",quad[i].pos,quad[i].op,quad[i].arg1,quad[i].arg2,quad[i].result);
    i=0;
    printf("\n\ncode generated :\n");
    while(i< n)
    {
```

```

if(quad[i].op=='+')
    codegen("ADD",i);
if(quad[i].op=='=')
    assignment(i);
if(quad[i].op=='-')
    if(!strcmp(quad[i].arg2,"\0"))
        uminus(i);
    else
        codegen("SUB",i);
if(quad[i].op=='*')
    codegen("MUL",i);
if(quad[i].op=='/')
    codegen("DIV",i);
i++;
}
getch();
fcloseall();
return 0;
}

void codegen(char op[5],int t)
{
    char str[25];
    printf("MOV %s,Ro\n",quad[t].arg1);
    printf("%s %s,Ro\n",op,quad[t].arg2);
    printf("MOV Ro,%s\n",quad[t].result);
}

void assignment(int t)
{
    char str[25];
    printf("MOV %s,%s\n",quad[t].result,quad[t].arg1);
}

void uminus(int t)
{
    char str[25];
    printf("MOV Ro,o\n");
    printf("SUB %s,Ro\n",quad[t].arg1);
    printf("MOV Ro,%s\n",quad[t].result);
}

void explore()
{
    int i,j,t,t1,t2;
    for(i=0;i < n;i++)
    {
        quad[i].pos=i;
        for(j=0,t=0;j < strlen(tuple[i])&&tuple[i][j]!='';j++)
        {
            quad[i].result[t++]=tuple[i][j];
        }
        t1=j;
        quad[i].result[t]='\0';
        if(tuple[i][j]=='=')
        {
            quad[i].op='=';
        }
        if(tuple[i][j+1]=='+'||tuple[i][j+1]=='-'||tuple[i][j+1]=='*'||tuple[i][j+1]=='/'||tuple[i][j+1]=='%')
        {
            quad[i].op=tuple[i][j+1];
            t1=j+1;
        }
        for(j=t1+1,t=0;j < strlen(tuple[i])&&tuple[i][j]!='+'&&tuple[i][j]!='-'&&tuple[i][j]!='*'&&tuple[i][j]!='/'&&tuple[i][j]!='%';j++)
        {
            quad[i].arg1[t++]=tuple[i][j];
        }
        t2=j;
        quad[i].arg1[t]='\0';
        if(tuple[i][j]=='+'||tuple[i][j]=='-'||tuple[i][j]=='*'||tuple[i][j]=='/'||tuple[i][j]=='%')

```

```

{
    quad[i].op=tuple[i][j];
}
for(j=t2+1,t=0;j< strlen(tuple[i]);j++)
{
    quad[i].arg2[t++]=tuple[i][j];
}
quad[i].arg2[t]='\0';
}
}

```

INPUT:

INT.TXT

```

t0=c+d
t1=t0*c
b=t0/c
c=-t1
t2=t3

```

DOWNLOAD THIS FILE :

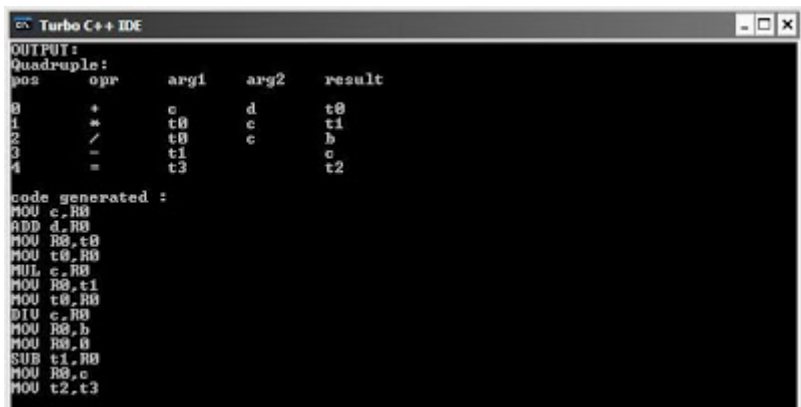
OUTPUT:



```

Turbo C++ IDE
INPUT:
intermediate codes:
t0=c+d
t1=t0*c
b=t0/c
c=-t1
t2=t3

```



```

Turbo C++ IDE
OUTPUT:
Quadruple:
pos   opr    arg1    arg2    result
0      +      c       d       t0
1      *      t0      c       t1
2      /      t0      c       b
3      -      t1      c       c
4      =      t3      c       t2

code generated :
MOV c,R0
ADD d,R0
MOV R0,t0
MOV t0,R0
MUL c,R0
MOV R0,t1
MOV t0,R0
DIV c,R0
MOV R0,b
MOV R0,0
SUB t1,R0
MOV R0,c
MOV t2,t3

```

Posted by cselab at [6:15 PM](#)

Labels: [code generation](#) , [code generation in c](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

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