Programs in Different Languages

THURSDAY, JANUARY 21, 2010

OPERATOR PRECEDENCE PARSER

```
#include stdio.h
#include string.h
#include conio.h
char stack[20],stack1[20],next,s[10];
int top=-1;
char prod[9][10]={
">><<<<<>>",
">><<<<<>>",
">>>><<<>>",
">>>><<<>>",
">>>><<<>>",
">>>>>ee>>",
"<<<<<<<=e",
">>>>>ee>>",
"<<<<<< };
char G[7][6]={
"E->E+E",
" /E-E",
" /E*E",
" /E/E",
" /(E)",
" /i "
};


int main()
{
char symbol;
int i=0,flag=0;
int j,k;
clrscr();
printf("Grammar\n");
for(j=0;j<7;j++)
{
for(k=0;k<6;k++)
printf("%c",G[j][k]);
printf("\n");
}
printf("\n\n OPERATOR PRECEDENCE RELATIONS \n");
printf("\n ------------------------------------------------------- \n");
printf("%c\t%c\t%c\t%c\t%c\t%c\t%c\t%c\t%c\t",'+','-','*','/','^','i','(',')','$');
printf("\n-----------------------------------------------------------------\n");
for(j=0;j<9;j++)
{
for(k=0;k<10;k++)
printf("%c\t",prod[j][k]);
printf("\n");
}
printf("Enter the string : ");
gets(s);
++top;
stack[top]='$';
next=s[i];
while(1)
{
if(stack[top]=='$'&& next=='$'||next=='\0')
break;
```

LABELS

- Books (1)
- C Language (1)
- C++ Program (1)
- Cheats in Maths (1)
- Compiler Design (7)
- Compiler Design Prog (7)
- Data Base Management Systems (DBMS) (3)
- English (3)
- Flex (1)
- Java (1)
- MASM Programs (MPI) (1)
- Maths (1)
- MS-Dos (1)
- Sports (1)
- Unix (1)

BLOG ARCHIVE

- ▼ 2010 (22)
  - ▼ January (22)
    - Records Of Sachin Tendukar
    - Role Play Topics
    - Role Play Topics
    - ROLE PLAY (Eg)
    - YACC
    - LEX TOOL
    - TOP DOWN PARSER
    - RECURSIVE DESCENT PARSER
    - OPERATOR PRECEDENCE PARSER
    - LR PARSER
    - LEXICAL ANALYSER
    - MS-Dos Programming
    - ADOBE Flex Programming & Applications
    - DBMS Lab Manual
    - Detail Description of C language
    - Tricks to Solve Maths Easy
    - DML/DDL COMMANDS & More (DBMS)
    - DBMS Programs
    - All Data Structure Programs in C++
    - MASM Programs (MPI)
    - Unix Shell Programs
    - Java Lab Manual

ABOUT

Xpert

View my complete profile

```
else
{
symbol=prod[f(stack[top])][f(next)];
if(symbol=='<'||symbol=='=')
{
stack[++top]=symbol;
stack[++top]=next;
}
else if(symbol=='>')
{
do
{
top--;
}while(stack[top]!='<');
stack[++top]=next;
if(next!='$')
{
for(j=0;j<=top;j++)
stack1[j]=stack[j];
stack1[j]=symbol;
}
}
else
flag=1;
next=s[++i];
}
}

printf("\n STACK : ");
for(j=0;j<=top;j++)
printf("%c",stack1[j]);
printf("%c",'$');
if(flag==0)
printf("\n\n Accepted");
else
printf("Rejected");
return 0;
}


int f(char ch)
{
switch(ch)
{
case '+':return 0;
case '-':return 1;
case '*':return 2;
case '/':return 3;
case '^':return 4;
case 'i':return 5;
case '(':return 6;
case ')':return 7;
case '$':return 8;
default :
{
printf("\n ERROR ");
exit(0);
}
}
}
```

OUTPUT:

```
OPERATOR PRECEDENCE RELATIONS

--------------------------------------------------------
+ - * / ^ i ( ) $
-------------------------------------------------------------------
```

```
> > < < < < < > >

> > < < < < < > >

> > > > < < < > >

> > > > < < < > >

> > > > < < < > >

> > > > > e e > >

< < < < < < < = e

> > > > > e e > >

< < < < < < < e e

Enter the string : i + i @ i $
ERROR

Enter the string : i +i*i$
STACK : $<+<*> $
Accepted
```

Posted by Xpert at 6:54 AM
Labels: Compiler Design, Compiler Design Prog

NO COMMENTS:

POST A COMMENT

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)