Ahsanullah University of Science and Technology (AUST)
Department of Computer Science and Engineering

Assignment 4

Course No.: CSE4130

Course Title: Formal Language and Compiler

Date of Submission-
16 July, 2023

Submitted To-
Ms. Md. Aminur Rahman & Ms. Iffatun Nessa

Submitted By-
Dipon Deb Dipu
190204036
A-2
Year- 4th
Semester-1st
Department-CSE

.

Code:

```
#include<bits/stdc++.h>

using namespace std;

int else_if_error(char *ptr)
{
    int i = 0;
    while(ptr[i] != '\0')
    {
        if(ptr[i] == 'e' && ptr[i+1] == ' ' && ptr[i+2] == 'i' && ptr[i+3] == 'f')
        {
            i += 4;
            return 0;
        }
        else if(ptr[i] == 'i' && ptr[i+1] == 'f')
        {
            i += 2;
            return 1;
        }
        else if(ptr[i] == 'e' && ptr[i+1] == 'l' && ptr[i+2] == 's' && ptr[i+3] == 'e' && ptr[i+4] == ' ' && ptr[i+5] != 'i')
        {
            i += 6;
            return -1;
        }
        i++;
    }
}

int brac_error(char *ptr)
{
    int c = 0;
    int i = 0;
    while(ptr[i] != '\0')
    {
        if(ptr[i] == '{')
        {
            i++;
            c++;
        }
        if(ptr[i] == '}')
        {
            i++;
            c--;
        }
        i++;
    }
    return c;
}

int kw_error(char *ptr)
{
    int i = 0;
    int c = 0;
    while(ptr[i] != '\0')
    {
        if(ptr[i] = 'k' && ptr[i+1] == 'w' && ptr[i+2] == ' ')
        {
            c++;
        }
        if(ptr[i] = 'i' && ptr[i+1] == 'd' && ptr[i+2] == ' ')
        {
            c--;
        }
        i++;
    }

    if(c >= 1)
        return 1;

    return 0;
}

int semi_col_error(char *ptr)
{
    int i = 0;
    int c = 0;
    while(ptr[i+1] != '\0')
    {
        if(ptr[i] == ';' && ptr[i+2] == ';')
        {
                c++;
        }
        i++;
    }

    if(c >= 1)
        return 1;

    return 0;
}

int numbering(char *ptr, char (*str)[500])
{
    int i = 0;
    int j;
    int k = 0;
    while(ptr[i+1] != '\0')
    {
        j = 0;
        char a[500];
        while(ptr[i] != '\n')
        {
            a[j] = ptr[i];
            j++;i++;
        }
        a[j] = '\0';
        i++;

        strcpy(str[k], a);
        k++;
    }
    strcpy(str[k], "\n");

}

int kw(char *ptr, int n)
{
    char key[6][8] = {"int", "char", "float",
"double", "return", "void"};

    int i;
    for(i = 0; i < 6; i++)
    {
        if(strcmp(ptr, key[i]) == 0)
            return 1;
    }

    return 0;
}
int id(char *ptr, int n)
{
    int i, s;i = 0;s = 0;
    if((ptr[i] == 'i' && ptr[i+1] == 'f') || (ptr[i]
== 'e' && ptr[i+1] == 'l' && ptr[i+2] == 's' &&
ptr[i+3] == 'e'))
    {
        return 0;
    }
```

```
    else if((ptr[i] <= 'z' && ptr[i] >= 'a') || (ptr[i]
<= 'Z' && ptr[i] >= 'A') || ptr[i] == '_')
    {
        s = 1;i++;
    }
    else
    {
        return 0;
    }
    while(ptr[i] != '\0')
    {
        if((ptr[i] <= 'z' && ptr[i] >= 'a') || (ptr[i]
<= 'Z' && ptr[i] >= 'A') || ptr[i] == '_' ||
isdigit(ptr[i]))
        {
            s = 1;i++;
        }
        else
        {
            s = 0;break;
        }
    }
    return s;
}


int categorized(char *ptr, int n, char *ptr2)
{
    int i = 0;
    int j;
    int jt;
int k = 0;
int c = 0;
while(ptr[i] != '\0')
{
   char a[500];
   j = 0;

   while(ptr[i] != ' ')
   {
      if(ptr[i] == '\n')
      {
         ptr2[k] = ptr[i];
         k++;i++;
         c++;
      }
      else
      {
         if(ptr[i] == '\0')
            break;
         a[j] = ptr[i];j++;i++;
      }

   }
   a[j] = '\0';
   jt = 0;

   if(kw(a, j))
   {
      ptr2[k] = 'k';k++;
      ptr2[k] = 'w';k++;
      ptr2[k] = ' ';k++;
      while(a[jt] != '\0')
      {
         ptr2[k] = a[jt];jt++;k++;
      }
   }
   else if(id(a, j))
   {
      ptr2[k] = 'i';k++;
      ptr2[k] = 'd';k++;
      ptr2[k] = ' ';k++;
      while(a[jt] != '\0')
      {
         ptr2[k] = a[jt];jt++;k++;
      }
   }
   else
   {

      while(a[jt] != '\0')
      {
         ptr2[k] = a[jt];jt++;k++;
      }
   }

   ptr2[k] = ptr[i];
   k++;i++;
   }
   ptr2[k] = '\0';
   c++;
   return c;
}

int spacing(char *ptr, int n, char *ptr2)
{
   int i = 0;
   int j = 0;
   while(ptr[i] != '\0')
   {
      if((ptr[i] == '<' || ptr[i] == '>' || ptr[i] == '=' || ptr[i] == '!') && ptr[i+1] == '=')
      {
         if(ptr2[j-1] != ' ')
         {
            ptr2[j] = ' ';j++;
         }
         ptr2[j] = ptr[i];j++;i++;
         ptr2[j] = ptr[i];j++;i++;
         ptr2[j] = ' ';j++;

      }
      else if(ptr[i] == ',' || ptr[i] == ';' || ptr[i] == '=' || ptr[i] == '+' || ptr[i] == '-' || ptr[i] == '*'
            || ptr[i] == '/' || ptr[i] == '\'' || ptr[i] == '\"' || ptr[i] == '(' || ptr[i] == ')' || ptr[i] == '{'
            || ptr[i] == '}' || ptr[i] == '<' || ptr[i] == '>')
      {
         if(ptr2[j-1] != ' ' && ptr2[j-1] != '\n')
         {
            ptr2[j] = ' ';j++;
         }

         ptr2[j] = ptr[i];j++;i++;
         ptr2[j] = ' ';j++;
      }
      else
      {
         ptr2[j] = ptr[i];j++;i++;
      }
   }
   ptr2[j] = '\0';
   return j;
}

int token_check(char *ptr, int c, char *token_char, int t_len)
{
   int tc1 = c - 1;
   int ic = t_len - 1;
   int flag = 1;
   while(ic != 0)
   {
      if(token_char[ic] != ptr[tc1])
         flag = 0;
      ic--;tc1--;
   }
   if(flag == 1)
      return 1;
   return 0;
}

int file_write(char *ptr, int t, char *ptr2)
{

   char ichar[7][8] = {"int","return","double","char","void","float","else"};
   int back_c;
   int i = 0;
   int a = 0;
   int lex_t = 0;
   int cit = -1;
   while(ptr[i] != '\0')
   {
      if(ptr[i] == '"')
         cit *= -1;
      if(ptr[i] == ' ' && cit == -1)
      {
         for(int j = 0; j < 7; j++)
         {
            a = token_check(ptr, i, ichar[j], strlen(ichar[j]));
            if(a == 1)
            {
               ptr2[lex_t] = ptr[i];lex_t++;
            }
```

```
         }
      }
      else
      {
         ptr2[lex_t] = ptr[i];
         lex_t++;
      }
      i++;
   }
   ptr2[lex_t] = '\0';
   return lex_t;
}

int file_read(char *m)
{
   char x;
   char z;
   FILE *in;
   int m_count = 0;
   int s_count = 0;
   in = fopen("input.txt", "r");
   if(in == NULL)
      return 0;
   while(!feof(in))
   {
      x = fgetc(in);
      if(x == '\n')
         s_count++;
      if(x != '\t')
      {
         if(x == '/')
         {
            z = fgetc(in);

            if(z == '*')
            {
               while(x != '\n')
                  x = fgetc(in);

               m[m_count] = '\n';m_count++;s_count++;
            }

            else if(z == '/')
            {

               while(x != '\n')
                  x = fgetc(in);

               m[m_count] = '\n';m_count++;s_count++;
            }
            else
            {
               m[m_count] = x;m_count++;
               m[m_count] = z;m_count++;
            }
         }
         else
         {
            m[m_count] = x;m_count++;
         }
      }
   }
   m[m_count] = '\0';
   s_count++;

   fclose(in);
   return s_count;
}
int main()
{
   char m[2000];
   char lex[2000];
   char sep[2000];
   char cate[2000];
   char idn[2000];

   int m_count = 0;
   int lex_count = 0;
   int sep_count = 0;
   int c;

   m_count = file_read(m);
   //puts(m);
   //cout << endl;
   lex_count = file_write(m, m_count, lex);
   //puts(lex);
   //cout << endl;
   sep_count = spacing(lex, lex_count, sep);
   //puts(sep);
   //cout << endl;
   c = categorized(sep, sep_count, cate);
   //puts(cate);
   //cout << endl;
   char num[c][500];
   numbering(cate,num);


   for(int i = 0; i < c; i++)
   {
      cout << i+1 << " " << num[i] << endl;
   }

   cout << endl;
   int br_cout = 0;
   int ef_cout = 0;
   for(int i = 0; i < c; i++)
   {
      int a = 0;
      a = brac_error(num[i]);
      br_cout += a;

      if(a < 0 )
      {
         br_cout = 0;
         cout << "Misplaced } at line " << i+1 << endl;
      }
   }
   for(int i = 0; i < c; i++)
   {
      int a = 0;
      a = else_if_error(num[i]);
      ef_cout += a;

      if(a < 0)
         cout << "Unmatched else at line " << i+1 << endl;
   }
   for(int i = 0; i < c; i++)
   {
      int a = 0;
      a = semi_col_error(num[i]);

      if(a == 1)
         cout << "Duplicate ; at line " << i+1 << endl;
   }
   for(int i = 0; i < c; i++)
   {
      int a = 0;
      a = kw_error(num[i]);

      if(a == 1)
         cout << "Duplicate Keyword at line " << i+1 << endl;
   }
}
```