|  |
| --- |
| **Lab 8:CFG for Iterative Statements and Declaration** |

|  |
| --- |
|  |

**Exercise 1**

Write down the CFG of Declaration statement valid for C++ language also implement it.

**TASK\_01:**

#include<iostream>

#include<string.h>

using namespace std;

bool decl();

bool init();

bool list();

bool Const();

string str[] = {"DT","ID","=","int",";"};

int i=0;

int main()

{

cout<<"string = ";

for(int i=0;i<5;i++)

cout<<str[i]<<" ";

if(decl())

cout<<"valid string";

else

cout<<"Invalid string\nsyntax error";

return 0;

}

bool decl()

{

bool flag=false;

if(str[i]=="DT")

{

i++;

if(str[i]=="ID")

{

i++;

if(init())

{

i++;

if(list())

flag= true;

}

}

}

else

return flag;

}

bool init()

{

bool flag=false;

if(str[i] == "=")

{

i++;

if(str[i]=="ID")

{

i++;

if(init())

flag= true;

}

else if(Const())

flag= true;

}

else if(str[i]==" ")

flag= true;

else

return flag;

}

bool list()

{

bool flag=false;

if(str[i]==";")

flag= true;

else if(str[i]==",")

{

i++;

if(str[i]=="ID")

{

i++;

if(init())

{

i++;

if(list())

flag= true;

}

}

}

else

return flag;

}

bool Const()

{

bool flag=false;

if(str[i]=="int")

flag=true;

else if(str[i]=="char")

flag= true;

else if(str[i]=="float")

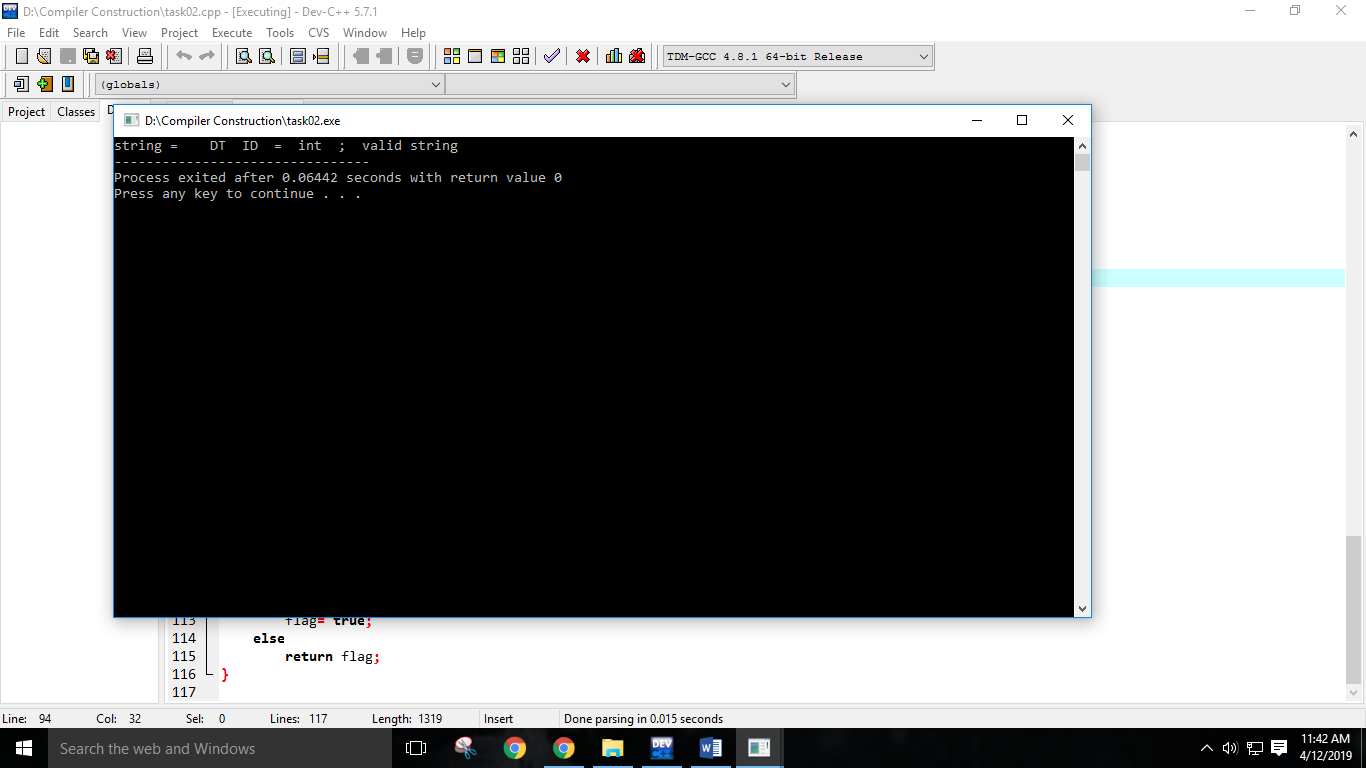
flag= true;

else

return flag;

}

**SCREENSHOT:**



**Exercise 2**

Write down the CFG of for loop statement valid for C++ language also implement it.

**TASK\_02:**

#include<iostream>

#include<string.h>

using namespace std;

int i = 0;

string w[]={"for","(","DT","ID","=","int",";","ID","relop","int",";","ID","indec",")",";"};

bool For();

bool AD();

bool decl();

bool con();

bool c();

bool AI();

bool id\_cons();

bool inc();

bool assign();

bool body();

bool ml\_st();

bool s\_st();

bool Const();

bool init();

bool list();

bool init()

{

if(w[i] == "=")

{

i++;

if(w[i]=="ID")

{

i++;

if(init())

return true;

}

else if(Const())

return true;

}

else if(w[i]==" ")

return true;

else return false;

}

bool list()

{

if(w[i]==";")

return true;

else if(w[i]==",")

{

i++;

if(w[i]=="ID")

{

i++;

if(init())

{

i++;

if(list())

return true;

}

}

}

else return false;

}

bool Const()

{

if(w[i]=="int")

return true;

else if(w[i]=="char")

return true;

else if(w[i]=="float")

return true;

else return false;

}

bool decl()

{

if(w[i]=="DT")

{

i++;

if(w[i]=="ID")

{

i++;

if(init())

{

i++;

if(list())

return true;

}

}

}

else return false;

}

bool For()

{

if(w[i]=="for")

{

i++;

if(w[i] == "(")

{

i++;

if(AD())

{

i++;

if(con())

{

i++;

if(w[i]==";")

{

i++;

if(AI())

{

i++;

if(w[i]==")")

{

i++;

if(body())

return true;

}

}

}

}

}

}

}

else return false;

}

bool AD()

{

if(decl())

return true;

else if(assign())

{

i++;

if(w[i]==";")

return true;

}

else if(w[i]==";")

return true;

else return false;

}

bool con()

{

if(c())

return true;

else if(w[i]==" ")

return true;

else return false;

}

bool c()

{

if(id\_cons())

{

i++;

if(w[i]=="relop")

{

i++;

if(id\_cons())

return true;

}

}

else return false;

}

bool AI()

{

if(inc())

return true;

else if(assign())

return true;

else if(w[i]==" ")

return true;

else return false;

}

bool id\_cons()

{

if(w[i]=="ID")

return true;

else if(w[i]=="int")

return true;

else return false;

}

bool inc()

{

if(w[i]=="ID")

{

i++;

if(w[i]=="indec")

return true;

}

else if(w[i]=="indec")

{

i++;

if(w[i]=="ID")

return true;

}

else return false;

}

bool assign()

{

if(w[i]=="ID")

{

i++;

if(w[i]=="asop")

{

i++;

if(id\_cons())

{

i++;

if(w[i]==";")

return true;

}

}

}

else return false;

}

bool body()

{

if(w[i]==";")

return true;

else if(s\_st())

return true;

else if(w[i]=="{")

{

i++;

if(ml\_st())

{

i++;

if(w[i]=="}")

return true;

}

}

else return false;

}

bool ml\_st()

{

if(s\_st())

{

i++;

if(ml\_st())

return true;

}

else if(w[i]==" ")

return true;

else return false;

}

bool (s\_st())

{

if(w[i]=="any single line commment")

return true;

else return false;

}

int main()

{

if(For())

cout<<"Valid";

else

cout<<"invalid";

}

**SCREENSHOT:**

