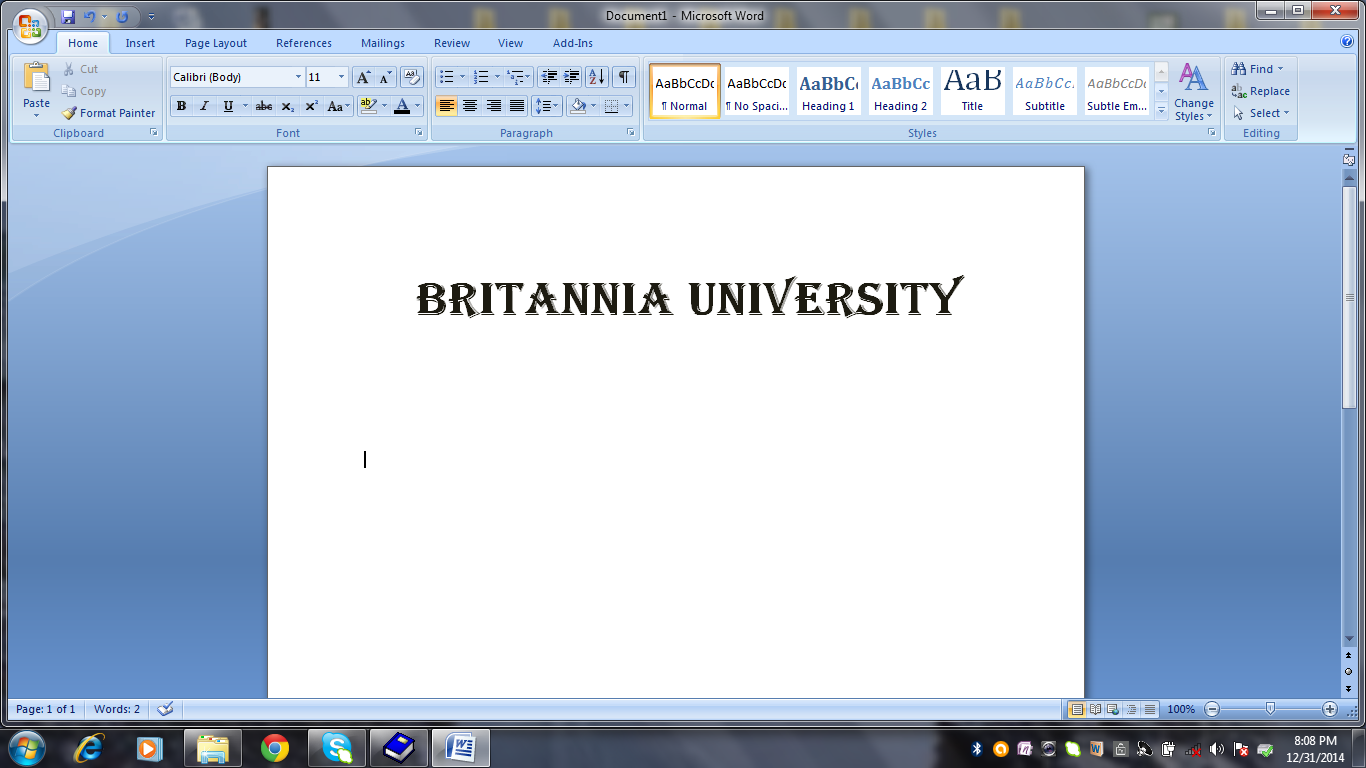


****

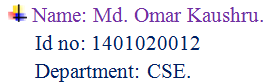
**Course title:** Compiler Design.

**Assignment on:** A C program to implement context free grammar.



Date of Submission: 08-08-2016

Submitted by:

****

* A C++ program to implement context free grammar.

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<process.h>

#include<stdlib.h>

**char** gramm**[**3**][**10**]={"SS+","SS\*"** **,"a"};**

**char** x**=**'a'**;**

**struct** comp

**{**

**char** grammer**[**3**][**10**];**

**char** solution**[**10**][**10**];**

**}**gr**;**

**int** main**()**

**{**

**char** temp**[**23**],**temp1**[**30**];**

**char** ques**[**30**];**

gets**(**ques**);**

**int** j**,**l**;**

**int** i**,**c**;**

**for(** i**=**0**;** i**<**3**;** i**++)**

**{**

strcpy**(**gr**.**grammer**[**i**],**gramm**[**i**]);**

**}**

l**=**strlen**(**ques**);**

**if(**ques**[**l**-**1**]==**'\*'**)**

**{**

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

**{**

c**=**0**;**

**if(**i**==**0**)**

strcpy**(**gr**.**solution**[**i**],**gr**.**grammer**[**i**+**1**]);**

**else** **if(**i**==**1**)**

**{**

strcpy**(**gr**.**solution**[**i**],**gr**.**grammer**[**i**-**1**]);**

strcat**(**gr**.**solution**[**i**],**gr**.**grammer**[**i**]);**

strcpy**(**temp**,**gr**.**solution**[**i**]);**

l**=**strlen**(**temp**);**

**for(**j**=**3**;** i**<**j**-**1**;** j**++)**

**{**

temp**[**j**]=**temp**[**j**+**1**];**

**}**

temp**[**l**-**1**]=**'\0'**;**

strcpy**(**gr**.**solution**[**i**],**temp**);**

**}**

**else**

**{**

l**=**strlen**(**gr**.**solution**[**1**]);**

strcpy**(**gr**.**solution**[**i**],**gr**.**solution**[**i**-**1**]);**

**for(** j**=**0**;** j**<**l**;** j**++)**

**{**

**if(**gr**.**solution**[**i**][**j**]==**'S' **||** gr**.**solution**[**i**][**j**]==**'s'**)**

**{**

gr**.**solution**[**i**][**j**]=**x**;**

**break;**

**}**

**}**

**}**

**if(**i**==**4**)**

strcpy**(**temp**,**gr**.**solution**[**4**]);**

**}**

**}**

**else** **if(**ques**[**l**-**1**]==**'+'**)***///checking 2nd condition*

**{**

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

**{**

c**=**0**;**

**if(**i**==**0**)**

strcpy**(**gr**.**solution**[**i**],**gr**.**grammer**[**i**+**1**]);**

**else** **if(**i**==**1**)**

**{**

strcpy**(**gr**.**solution**[**i**],**gr**.**grammer**[**i**]);**

strcat**(**gr**.**solution**[**i**],**gr**.**grammer**[**i**-**1**]);**

strcpy**(**temp**,**gr**.**solution**[**i**]);**

l**=**strlen**(**temp**);**

**for(**j**=**3**;** i**<**j**-**1**;** j**++)**

**{**

temp**[**j**]=**temp**[**j**+**1**];**

**}**

temp**[**l**-**1**]=**'\0'**;**

strcpy**(**gr**.**solution**[**i**],**temp**);**

**}**

**else**

**{**

l**=**strlen**(**gr**.**solution**[**1**]);**

strcpy**(**gr**.**solution**[**i**],**gr**.**solution**[**i**-**1**]);**

**for(** j**=**0**;** j**<**l**;** j**++)**

**{**

**if(**gr**.**solution**[**i**][**j**]==**'S' **||** gr**.**solution**[**i**][**j**]==**'s'**)**

**{**

gr**.**solution**[**i**][**j**]=**x**;**

**break;**

**}**

**}**

**}**

**if(**i**==**4**)**

strcpy**(**temp1**,**gr**.**solution**[**4**]);**

**}**

**}**

l**=**strlen**(**ques**);**

**if(**strcmp**(**ques**,**temp**)==**0**)**

**{**

printf**("the string aa+a\* can be generated by this grammar\n");**

printf**("Solution\n\n");**

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

printf**("%s \n",**gr**.**solution**[**i**]);**

**}**

**else** **if(**strcmp**(**ques**,**temp1**)==**0**)**

**{**

printf**("the string aa\*a+ can be generated by this grammar\n");**

printf**("Solution\n\n");**

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

printf**("%s \n",**gr**.**solution**[**i**]);**

**}**

**else** **if(**ques**[**0**]==**'a' **&&** l**==**1**)**

printf**("the string a can be generated by this grammar\n");**

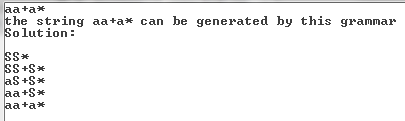
**else**

printf**("Invalid");**

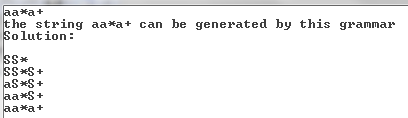
**return** 0**;**

**}**

**Sample input/output-1:**



**Sample input/output-2:**



**Sample input/output-3:**

