**Lab 6: RECURSIVE DESCENT PARSER**

Write a recursive descent parser for the following simple grammars.

1.)

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int curr=0;

char str[100];

void S();

void T();

void Tprime();

void reject()

{

printf("reject\n");

exit(0);

}

void valid()

{

printf("accept\n");

exit(0);

}

void S()

{

if(str[curr]=='a')

{

curr++;

return;

}

else if(str[curr]=='>')

{

curr++;

return;

}

else if(str[curr]=='(')

{

curr++;

T();

if(str[curr]==')')

{

curr++;

return;

}

else

reject();

}

else

reject();

}

void T()

{

S();

Tprime();

}

void Tprime()

{

if(str[curr]==',')

{

curr++;

S();

return;

}

}

int main()

{

printf("enter string\n");

scanf("%s",str);

S();

if(str[curr]=='$')

{

valid();

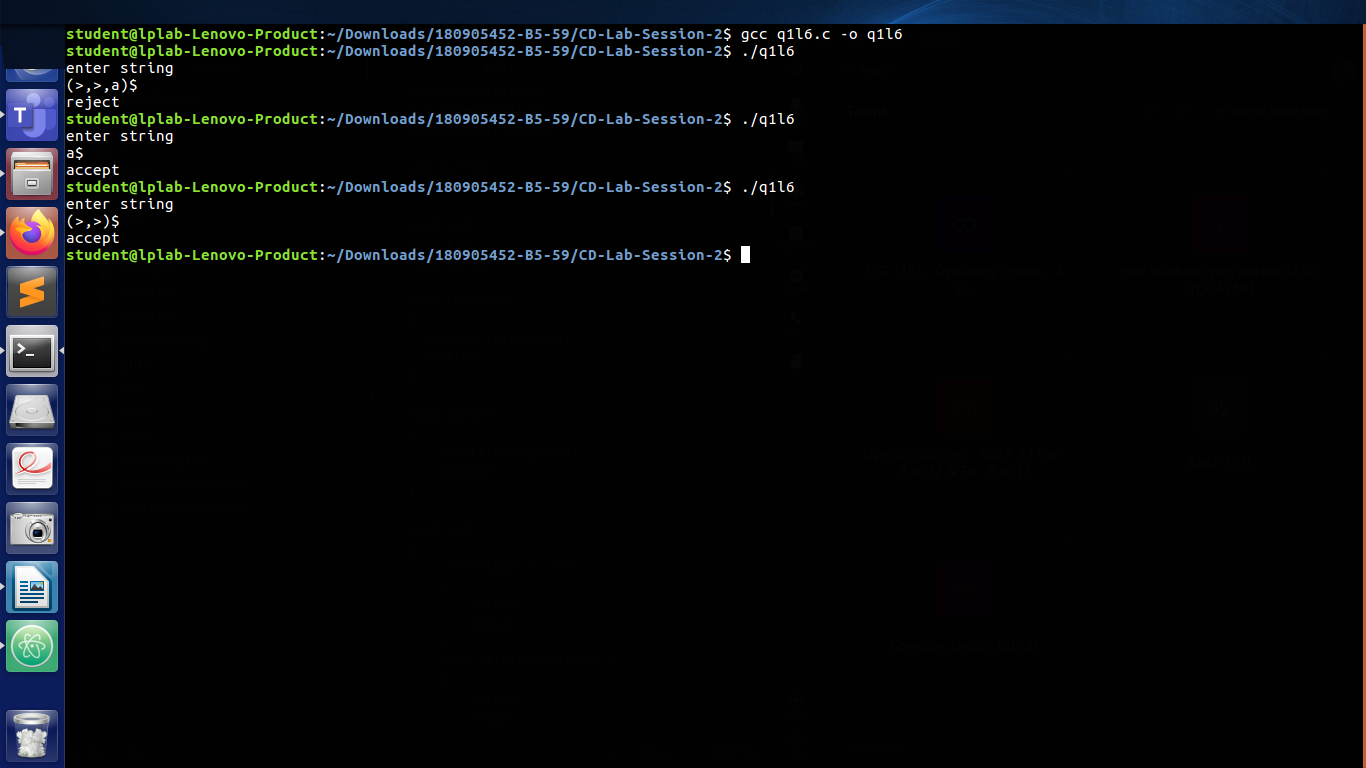
}

else

reject();

}

Output



2.)

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int curr=0;

char str[100];

void S();

void U();

void V();

void W();

void reject()

{

printf("reject\n");

exit(0);

}

void accept()

{

printf("accept\n");

exit(0);

}

void S()

{

U();

V();

W();

}

void U()

{

if(str[curr]=='(')

{

curr++;

S();

if(str[curr]==')')

{

curr++;

return;

}

else

reject();

}

else if(str[curr]=='a')

{

curr++;

S();

if(str[curr]=='b')

{

curr++;

return;

}

else

reject();

}

else if(str[curr]=='d')

{

curr++;

return;

}

else

reject();

}

void V()

{

if(str[curr]=='a')

{

curr++;

V();

return;

}

}

void W()

{

if(str[curr]=='c')

{

curr++;

W();

return;

}

}

int main()

{

printf("enter string\n");

scanf("%s",str);

S();

if(str[curr]=='$')

{

accept();

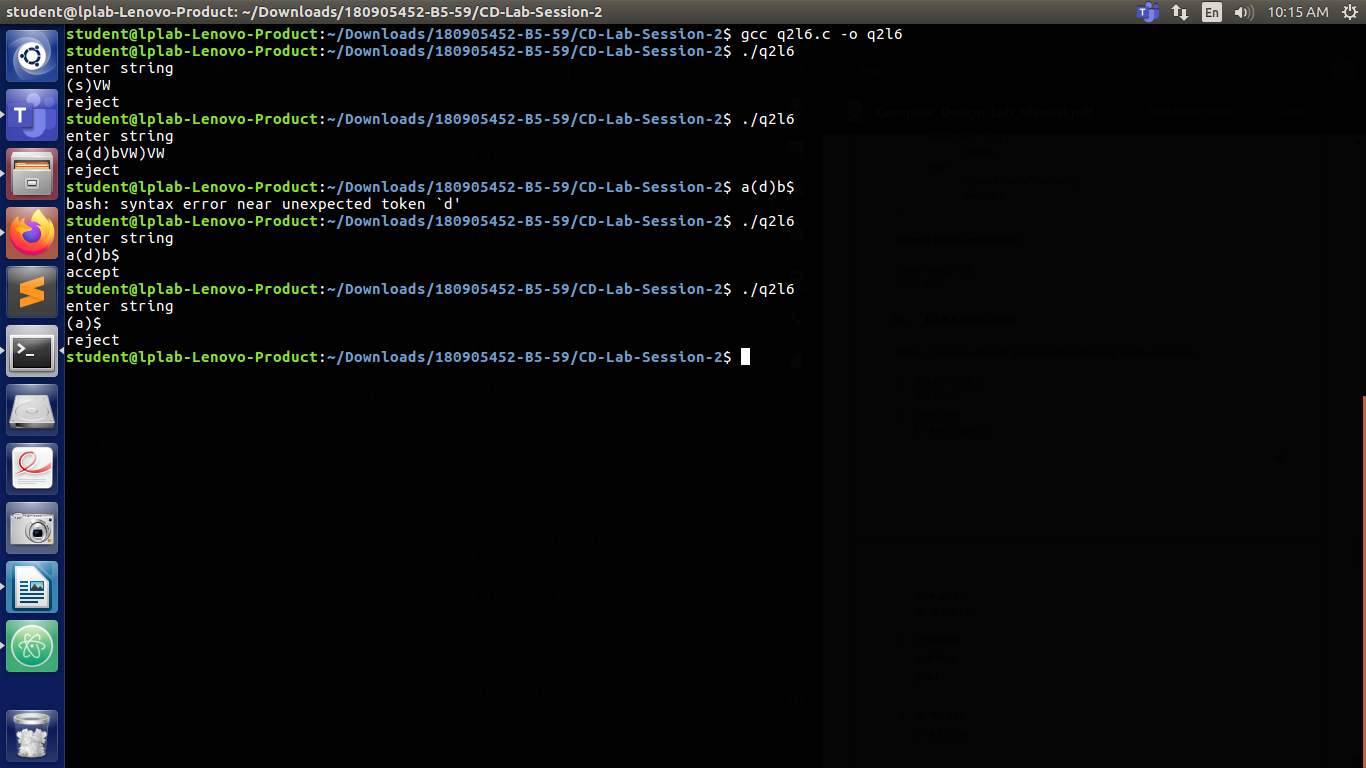
}

else

reject();

}

Output



3.)

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int curr=0;

char str[100];

void S();

void A();

void Aprime();

void B();

void reject()

{

printf("reject\n");

exit(0);

}

void accept()

{

printf("accept\n");

exit(0);

}

void S()

{

if(str[curr]=='a')

{

curr++;

A();

if(str[curr]=='c')

{

curr++;

B();

if(str[curr]=='e')

{

curr++;

return;

}

else

reject();

}

else reject();

}

else reject();

}

void A()

{

if(str[curr]=='b')

{

curr++;

Aprime();

}

else

reject();

}

void Aprime()

{

if(str[curr]=='b')

{

curr++;

Aprime();

return;

}

}

void B()

{

if(str[curr]=='d')

{

curr++;

return;

}

}

int main()

{

printf("enter string\n");

scanf("%s",str);

S();

if(str[curr]=='$')

{

accept();

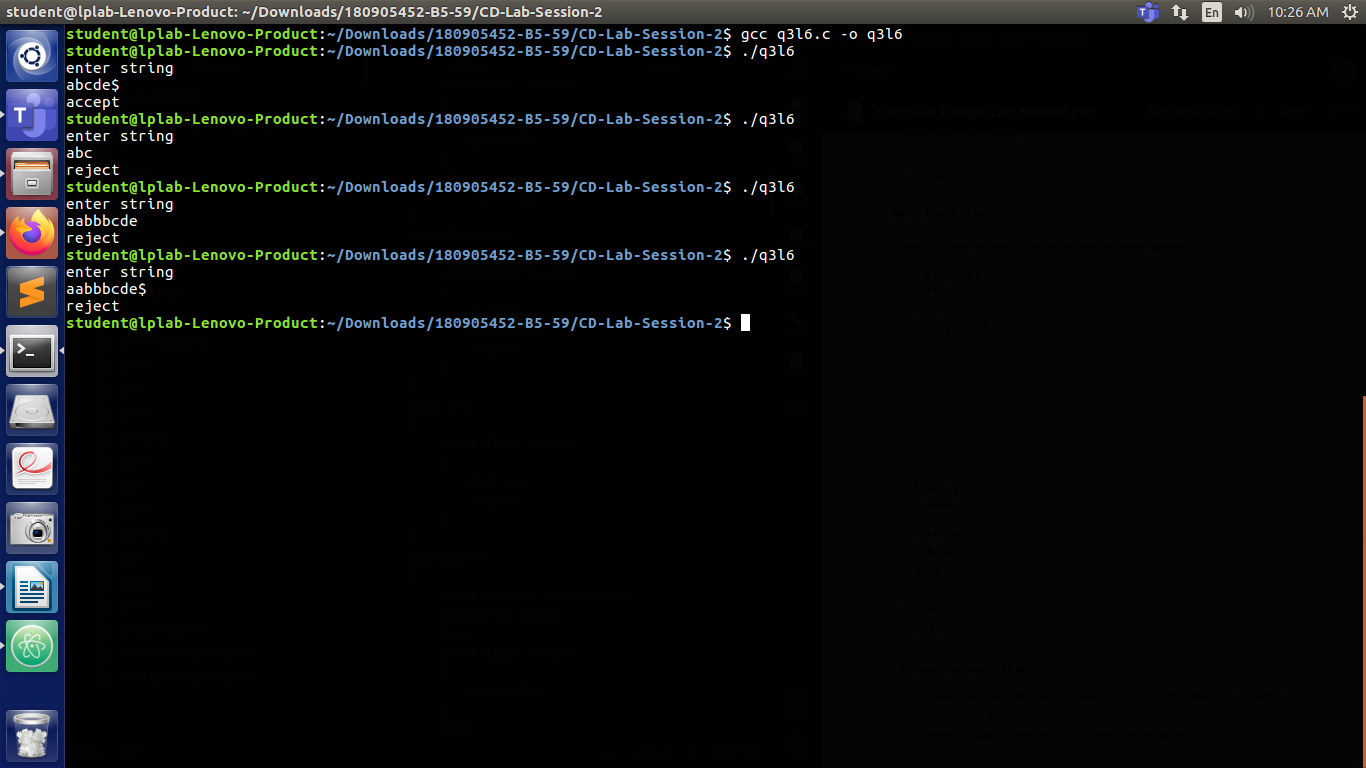
}

else

reject();

}

Output



4.)

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int curr=0;

char str[100];

void S();

void L();

void Lprime();

void reject()

{

printf("reject\n");

exit(0);

}

void accept()

{

printf("accept\n");

exit(0);

}

void S()

{

if(str[curr]=='(')

{

curr++;

L();

if(str[curr]==')')

{

curr++;

return;

}

else

reject();

}

else if(str[curr]=='a')

{

curr++;

return;

}

else

reject();

}

void L()

{

S();

Lprime();

}

void Lprime()

{

if(str[curr]==',')

{

curr++;

S();

Lprime();

return;

}

}

int main()

{

printf("enter string\n");

scanf("%s",str);

S();

if(str[curr]=='$')

{

accept();

}

else

reject();

}

Output

