**Program 1a**

%{

#include<stdio.h>

int id=0,op=0,br=0;

%}

%%

[a-zA-Z0-9]+ {id++; printf("\nthe identifier is:\n"); ECHO;}

[+|\*|/]+ {op++; printf("\nthe operator is :\n"); ECHO;}

[-] {op++; printf("\nthe operator is :\n"); ECHO;}

"(" {br++;}

")" {if(br==0) {br=1; return 0;} else br--;}

%%

int main()

{

printf("enter the expression:\n");

yylex();

if((op+1)==id && br==0)

{

printf("the entered expression is valid\n");

printf("the total number of identifiers %d:\n",id);

printf("the total number of operators %d:\n",op);

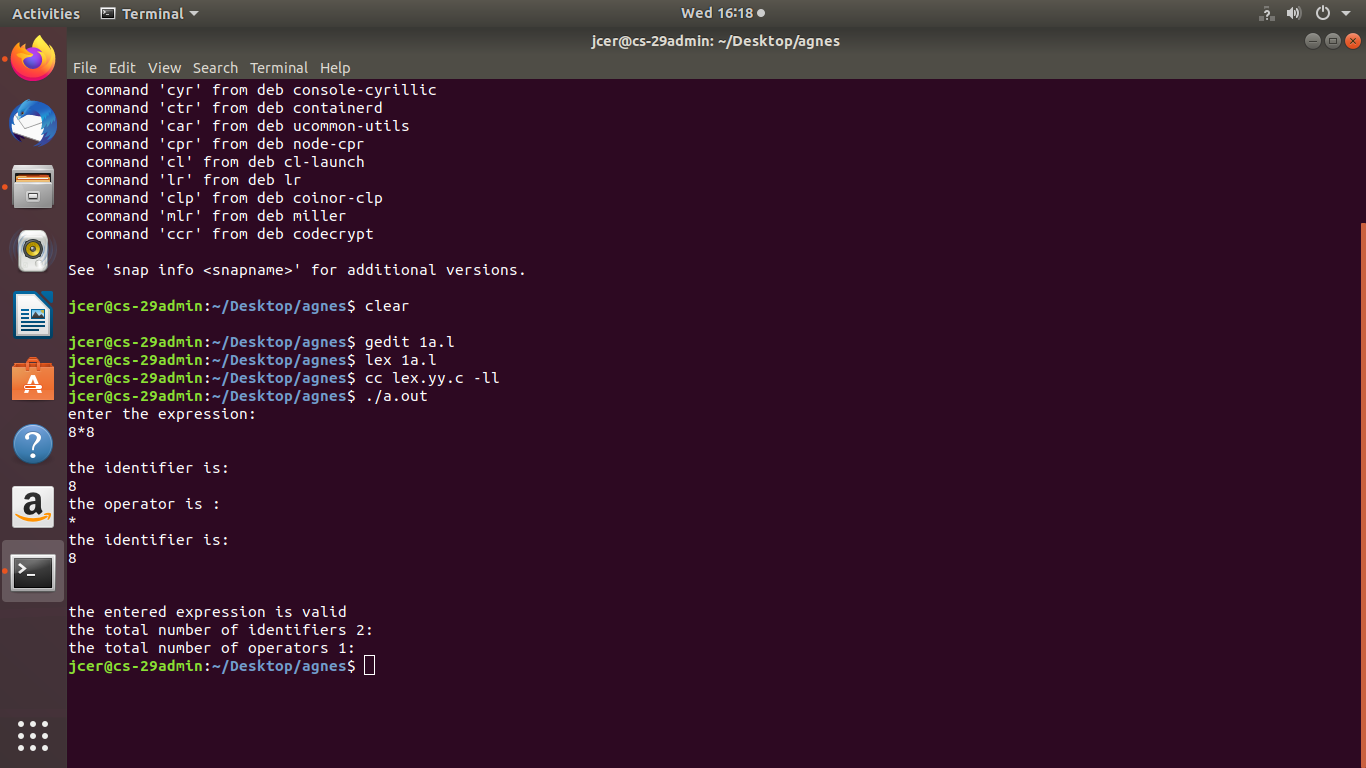
}

else

printf("the entered expression is invalid\n");

}

Program 1 Output



**Program 1b**

%{

#include<stdio.h>

#include"y.tab.h"

extern int yylval;

%}

%%

[0-9]+ { yylval=atoi(yytext); return NUM;}

. {return yytext[0];}

%%

**Program 1b.l**

%{

#include<stdio.h>

#include<stdlib.h> intyylex(); void yyerror(const char \*s);

%}

%token NUM

%left '+''-'

%left '\*''/'

%nonassoc UMINUS %%

expr:NUM { $$ = $1; yylval=$$;} |

expr'+'expr {$$=$1+$3; yylval=$$;}

| expr'-'expr {$$=$1-$3; yylval=$$;}

| expr'\*'expr {$$=$1\*$3; yylval=$$;}

| expr'/'expr {if($3==0) { printf("Error : Divided by zero.\n"); exit(0);}

else{

$$=$1/$3; yylval=$$;}}

|'('expr')' {$$=$2;}

| '-'expr %prec UMINUS {$$=-$2;yylval=$$; }; %%

int main() {

printf("Enter a expression to evaluate: \n");

yyparse(); printf("The expression is valid.\n");

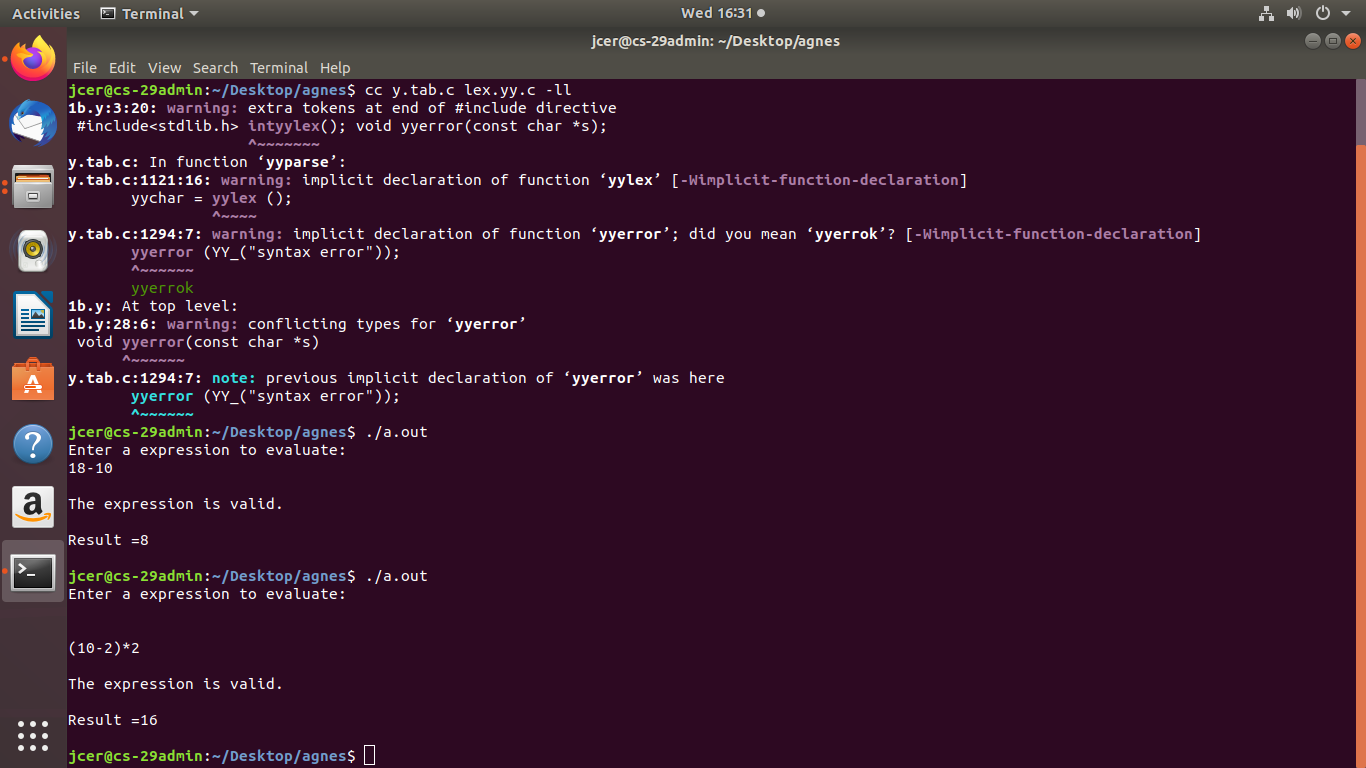
printf("\nResult =%d\n\n",yylval);

exit(0);}

void yyerror(const char \*s){

printf("The expression is invalid.\n");

exit(0); }



**Program 2**

%{

#include"y.tab.h"

%}

%%

[a] {return A;}

[b] {return B;}

%%

**Program 2y**

%{

#include<stdio.h>

#include<stdlib.h>

int valid;

%}

%token A B

%%

str: A expr B

| expr1

;

expr: A expr|;

expr1: B ;

%%int main()

{

printf("\nEnter the Expression : \n");

yyparse();

printf("The Grammar is valid.\n");

exit(0);

}

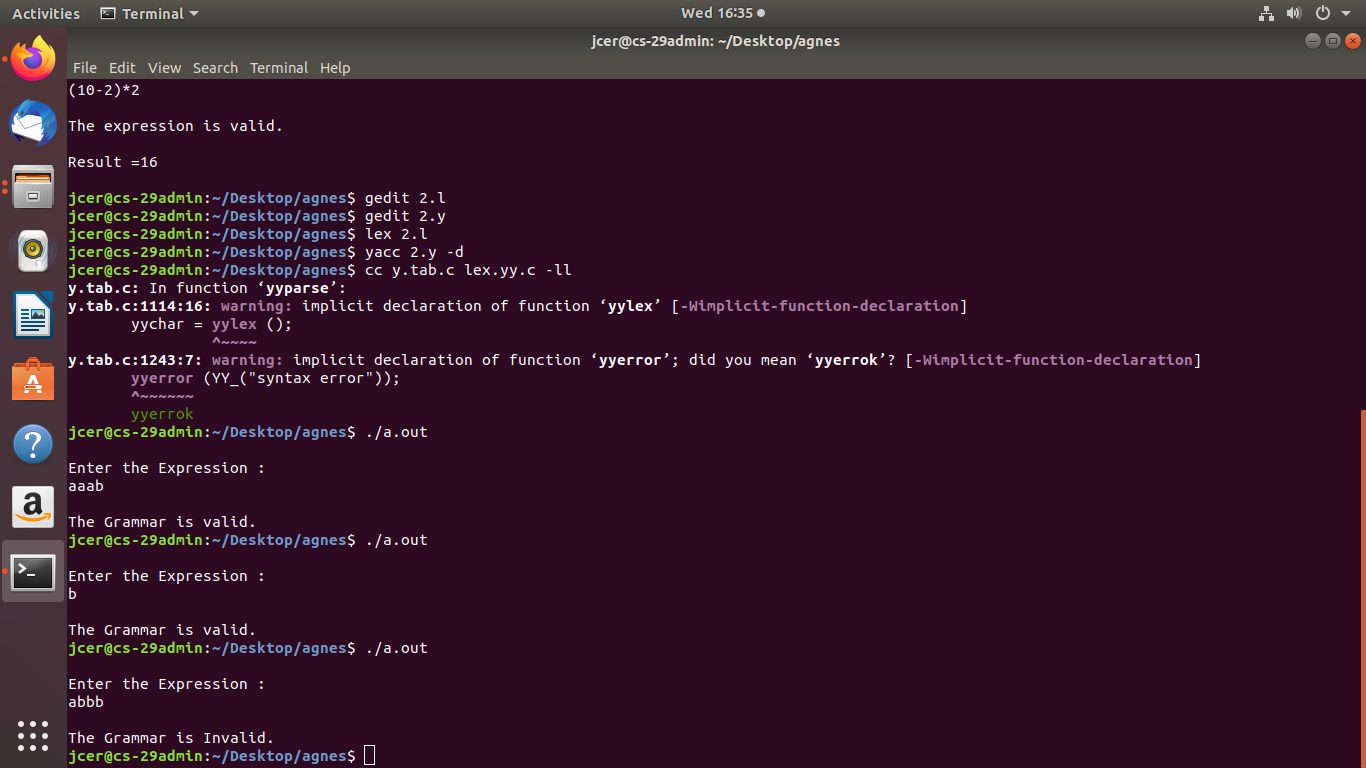
int yyerror()

{

printf("\nThe Grammar is Invalid.\n");

exit(0);

}



Program 3

#include<stdlib.h>

#include<string.h>

#include<stdio.h>

char prod[3][10]={"A->aBa","B->bB","B->@"},input[10],stack[25];

int top=-1;int j=0,k,l;

void push(char item)

{

stack[++top]=item;

}

void pop()

{

top=top-1;

}

void display()

{

int j;

for(j=top;j>=0;j--)

printf("%c",stack[j]);

}

void stackpush(char p)

{

if(p=='A')

{

pop();

for(j=strlen(prod[0])-1;j>=3;j--)

push(prod[0][j]);

}

else

{

pop();

for(j=strlen(prod[1])-1;j>=3;j--)

push(prod[1][j]);

}

}

void main()

{

char c;

int i;

printf("enter the input string to parse:-");

scanf("%s",input);

for(i=0;input[i]!='\0';i++)

{

if((input[i]!='a')&&(input[i]!='b')&&(input[i]!='$'))

{

printf("invalid string");

exit(0);

}

}

if(input[i-1]!='$')

{

printf("\n\n input string entered without end marker $");

exit(0);

}

push('$');

push('A');

i=0;

printf("\n\n");

printf("stack\tInput\taction");

while(i!=strlen(input)&&stack[top]!='$')

{

printf("\n");

for(l=top;l>=0;l--)

printf("%c",stack[l]);

printf("\t");

for(l=i;l<strlen(input);l++)

printf("%c",input[l]);

printf("\t");

if(stack[top]=='A')

{

printf("A->aBa");

stackpush('A');

}

else if(stack[top]=='B')

{

if(input[i]!='b')

{

printf("B->@");

printf("\t matched E");

pop();

}

else

{

printf("B->bB");

stackpush('B');

}

}

else

{

if(stack[top]==input[i])

{

printf("pop%c",input[i]);

printf("\tmatched %c",input[i]);

pop();

i++;

}

else

break;

}

}

if(stack[top]=='$' && input[i]=='$')

{

printf("\n$\t$");

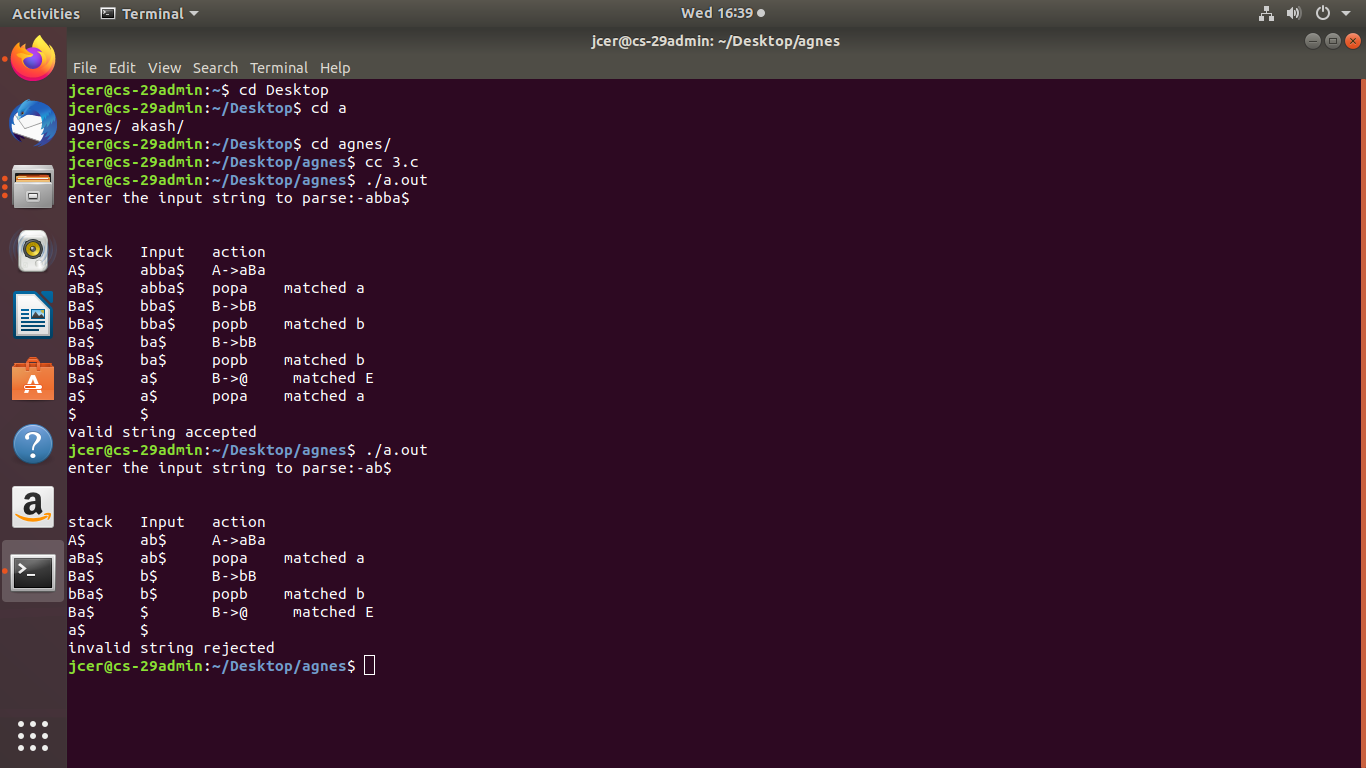
printf("\nvalid string accepted\n");

}

else

printf("\ninvalid string rejected\n");

}



**Program 4**

#include<stdio.h>

#include<string.h>

int k=0,z=0,i=0,j=0,c=0;

char a[16],ac[20],stk[15],act[10];

void check();

int main()

{

puts("GRAMMAR is E->E+E \n E->E\*E \n E->(E) \n E->id");

puts("\nEnter input string :");

gets(a);

c=strlen(a);

strcpy(act,"SHIFT->");

puts("stack \t input \t action");

for(k=0,i=0; j<c; k++,i++,j++)

{

if(a[j]=='i' && a[j+1]=='d')

{

stk[i]=a[j];

stk[i+1]=a[j+1];

stk[i+2]='\0';

a[j]=' ';

a[j+1]=' ';

printf("\n$%s\t%s$\t%sid",stk,a,act);

check();

}

else

{

stk[i]=a[j];

stk[i+1]='\0';

a[j]=' ';

printf("\n$%s\t%s$\t%ssymbols",stk,a,act); check();

}

}

}

void check()

{

strcpy(ac,"REDUCE TO E");

for(z=0; z<c; z++)

if(stk[z]=='i' && stk[z+1]=='d')

{

stk[z]='E';

stk[z+1]='\0';

printf("\n$%s\t%s$\t%s",stk,a,ac);

j++;

}

for(z=0; z<c; z++)

if(stk[z]=='E' && stk[z+1]=='+' && stk[z+2]=='E')

{

stk[z]='E';

stk[z+1]='\0';

stk[z+2]='\0';printf("\n$%s\t%s$\t%s",stk,a,ac);

i=i-2;

}

for(z=0; z<c; z++)

if(stk[z]=='E' && stk[z+1]=='\*' && stk[z+2]=='E')

{

stk[z]='E';

stk[z+1]='\0';

stk[z+2]='\0';

printf("\n$%s\t%s$\t%s",stk,a,ac);

i=i-2;

}

for(z=0; z<c; z++)

if(stk[z]=='(' && stk[z+1]=='E' && stk[z+2]==')')

{

stk[z]='E';

stk[z+1]='\0';

stk[z+1]='\0';

printf("\n$%s\t%s$\t%s",stk,a,ac);

i=i-2;

}

}



**Program 5**

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<string.h>

char op[2], arg1[5], arg2[5], result[5];

void main()

{

FILE \*fp1, \*fp2;

fp1=fopen("input.txt","r");

fp2=fopen("output.txt","w");

while(!feof(fp1))

{

fscanf(fp1, "%s%s%s%s", result,arg1,op,arg2);

if(strcmp(op,"+")==0)

{

fprintf(fp2,"\nMOV R0,%s",arg1);

fprintf(fp2,"\nADD R0,%s",arg2);

fprintf(fp2,"\nMOV %s,R0",result);

}

if(strcmp(op,"\*")==0)

{

fprintf(fp2,"\nMOV R0,%s",arg1);

fprintf(fp2,"\nMUL R0,%s",arg2);

fprintf(fp2,"\nMOV %s,R0",result);

}

if(strcmp(op,"-")==0)

{

fprintf(fp2,"\nMOV R0,%s",arg1);

fprintf(fp2,"\nSUB R0,%s",arg2);

fprintf(fp2,"\nMOV %s,R0",result);

}

if(strcmp(op,"=")==0)

{

fprintf(fp2,"\nMOV R0,%s",arg1);

fprintf(fp2,"\nMOV %s,R0",result);

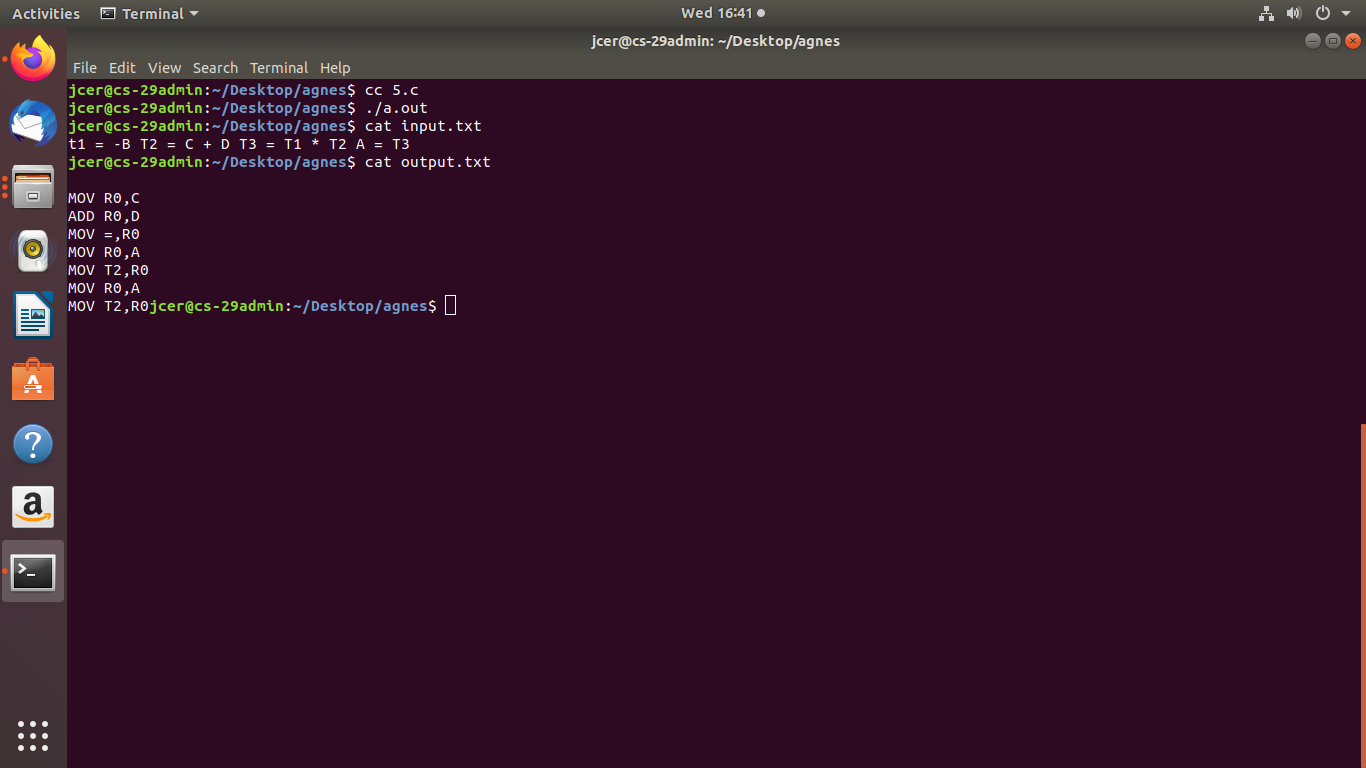
}

}

fclose(fp1);

fclose(fp2);

}



**Program 6a**

%{

#include<stdio.h>

int cc=0;

%}

%%

"/\*"[^\*/]\*"\*/" {cc++;}

"//".\* {cc++;}

%%

int main()

{

yyin=fopen("1.c","r");

yyout=fopen("2.c","w");

yylex();

fclose(yyin);

fclose(yyout);

printf("number of comment lines in the given file:%d\n",cc);

}

**Program 6b**

%{

#include<stdio.h>

#include "y.tab.h"

%}

%%

[+|-|\*|/|=|<|>|(|)|{|}|] {printf("Operator is:%s\n",yytext);return OP;}

int|char|float|void|if|else|return|break|printf|scanf|double|main {printf("Keyword is:%s\n",yytext);return KEY;}

[a-z|A-Z|0-9]\* {printf("Identifier is:%s\n",yytext);return ID;}

.

%%

**Program 6b.y**

%{

#include<stdio.h>

#include<stdlib.h>

int id=0,key=0,op=0;

%}

%token ID KEY OP %%

input:ID input {++id;}

|KEY input {++key;}

|OP input {++op;}

|ID {++id;}

|KEY {++key;}

|OP {++op;};

%%

extern FILE \*yyin;

main(){

yyin=fopen("ex.c","r");

yyparse();

printf("Keywords=%d\n",key);

printf("Identifiers=%d\n",id);

printf("Operators=%d\n",op);}

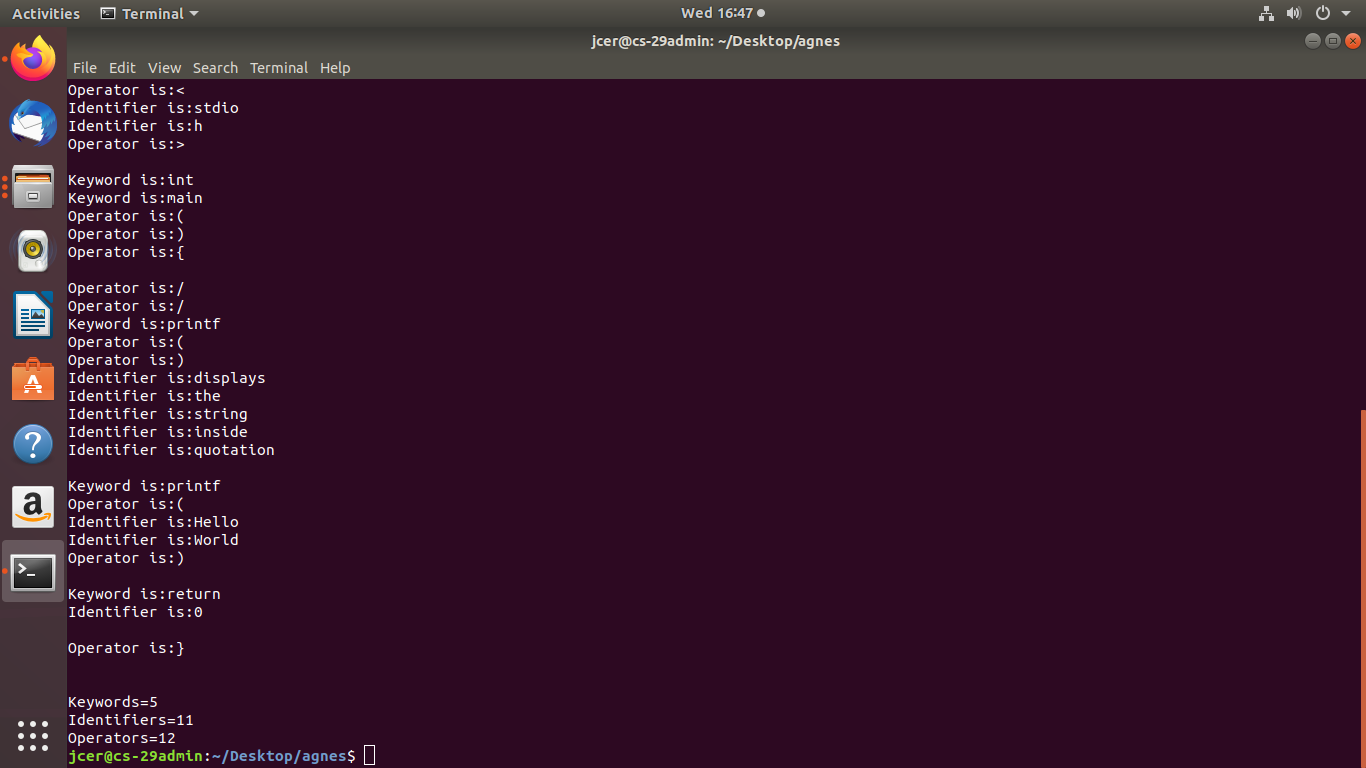
yyerror(){

printf("Error");

exit(0);

}

6a.png



**Program 7**

#include<stdio.h>

int main()

{

int i, count, n, tq, temp1, sq=0, swt=0, stat=0;

int st[10], tat[10], bt[10], wt[10];

float awt=0.0, atat=0.0;

printf("Enter the total number of processes : \n");

scanf("%d",&n);

printf("Enter the Time Quantum : \n");

scanf("%d",&tq);

for(i=0;i<n;i++)

{

printf("Enter Burst Time :\n");

scanf("%d",&bt[i]);

st[i]=bt[i];

}

while(1)

{ // Write the variables

for(i=0,count=0;i<n;i++)

{

temp1=tq;

if(st[i]==0)

{

count++;

continue;

}

if(st[i]>tq)

st[i]=st[i]-tq;

else

if(st[i]>=0)

{

temp1=st[i];

st[i]=0;

}

sq=sq+temp1;

tat[i]=sq;

}

if(n==count)

break;

}

for(i=0;i<n;i++)

{wt[i]=tat[i]-bt[i];

swt=swt+wt[i];

stat=stat+tat[i];

}

awt=(float)swt/n;

atat=(float)stat/n;

printf("Process No. Burst Time  Wait Time   Turn Around Time\n");

for(i=0;i<n;i++)

{

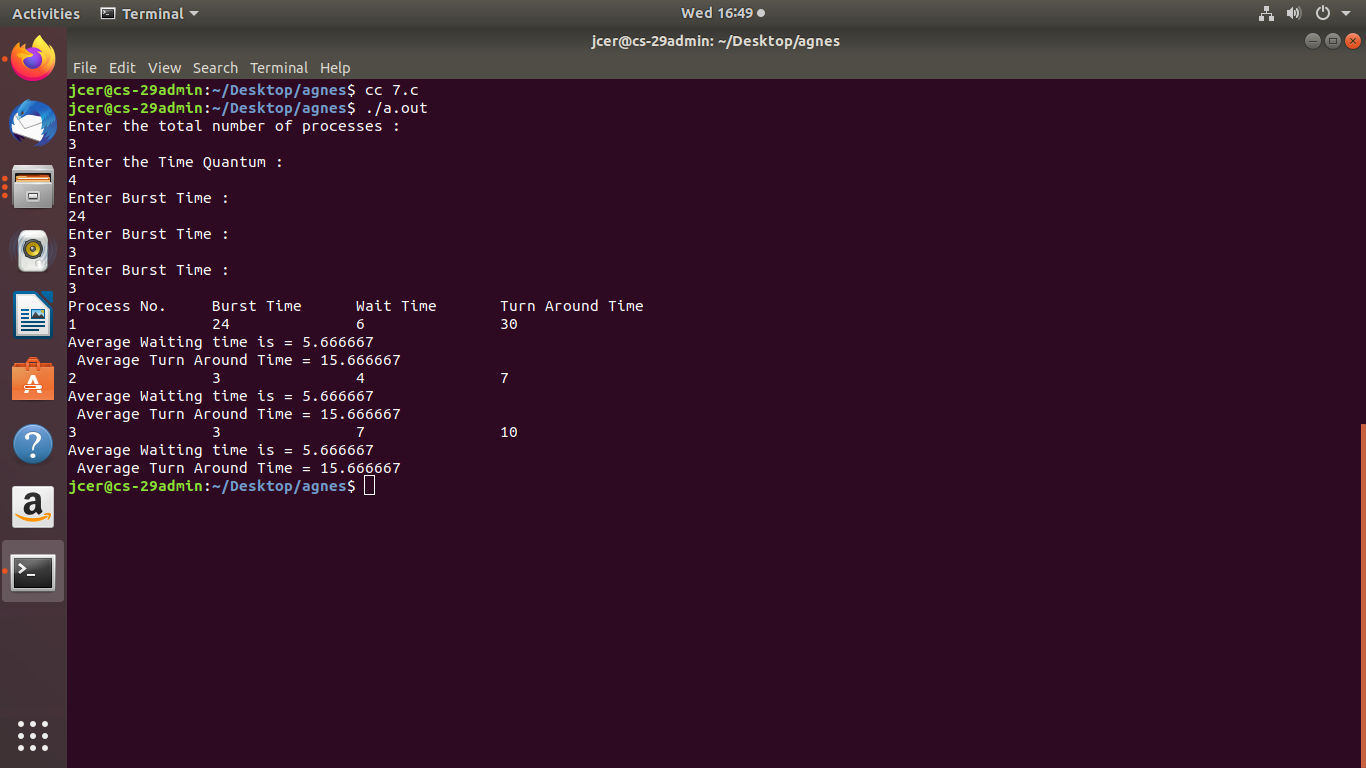
printf("%d\t\t%d\t\t%d\t\t%d\n",i+1,bt[i],wt[i],tat[i]);

printf("Average Waiting time is = %f \n Average Turn Around Time = %f\n",awt,atat);

}

return 0;

}



**Program 7b**

#include<stdio.h>

int main()

{

int at[10],bt[10],temp[10];

int i,smallest,count=0,time,n;

double wt=0,tt=0,end;

float awt,att;

printf("Enter the total number of processes:\n");

scanf("%d",&n);

for(i=0;i<n;i++)

{

printf("Enter Arrival Time:\n");

scanf("%d",&at[i]);

printf("Enter Burst Time:\n");

scanf("%d",&bt[i]);

temp[i]=bt[i];

}

bt[9]=9999;

for(time=0;count!=n;time++)

{

smallest=9;

for(i=0;i<n;i++)

{

if(at[i] <= time && bt[i] < bt[smallest] && bt[i] > 0)

{

smallest=i;

}

}

bt[smallest]--;

if(bt[smallest]==0)

{

count++;

end=time+1;

wt=wt+end-at[smallest]-temp[smallest];

tt=tt+end-at[smallest];

}

}

awt=wt/n;

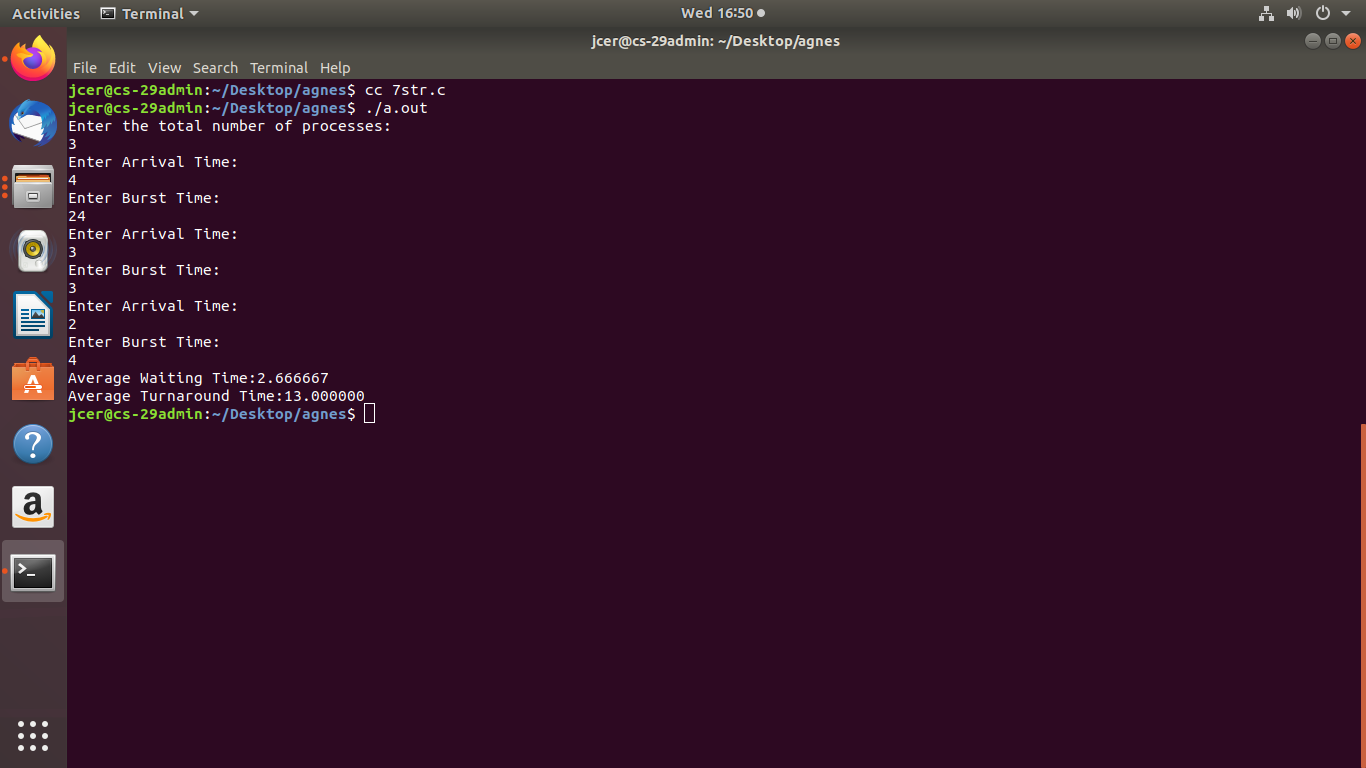
att=tt/n;

printf("Average Waiting Time:%f\n",awt);

printf("Average Turnaround Time:%f\n",att);

return 0;

}



**Program 8**

#include<stdio.h>

#include<stdlib.h>

int

available[10],allocation[10][10],maxi[10][10],need[10][10],c[10],i,j,k,l=

0,n,r,seq[10];

int grant()

{

int x;

for(x=0;x<r;x++)

{

if(!(need[i][x]<=available[x]))

return 0;

}

return 1;

}

void insert(int ele)

{

seq[l++]=ele;

}

void display()

{

printf("Safe sequence:<");

for(i=0;i<l;i++)

printf("%d,",seq[i]);

printf("%d>\n",seq[i]);

}

void main()

{

int sum,p[10],z=0,allow;

printf("Enter no. of processes:");

scanf("%d",&n);

printf("Enter no. of resources:");

scanf("%d",&r);

printf("Enter available resource matrix:\n");

for(i=0;i<r;i++)

scanf("%d",&available[i]);

printf("Enter allocation matrix:\n");

for(i=0;i<n;i++)

{

for(j=0;j<r;j++)

{

scanf("%d",&allocation[i][j]);

}

}

printf("Enter max. matrix:\n");

for(i=0;i<n;i++)

{

for(j=0;j<r;j++){

scanf("%d",&maxi[i][j]);

need[i][j]=maxi[i][j]-allocation[i][j];

}

}

for(k=0;k<n;k++)

{

for(i=0;i<n;i++)

{

allow=grant();

if(allow!=0 && p[i]!=1)

{

insert(i);

p[i]=1;

for(j=0;j<r;j++)

{

available[j]=available[j]+allocation[i][j];

}

}

}

}

for(i=0;i<n;i++)

{

if(p[i]==0)

{

printf("Unsafe state\n");

exit(0);

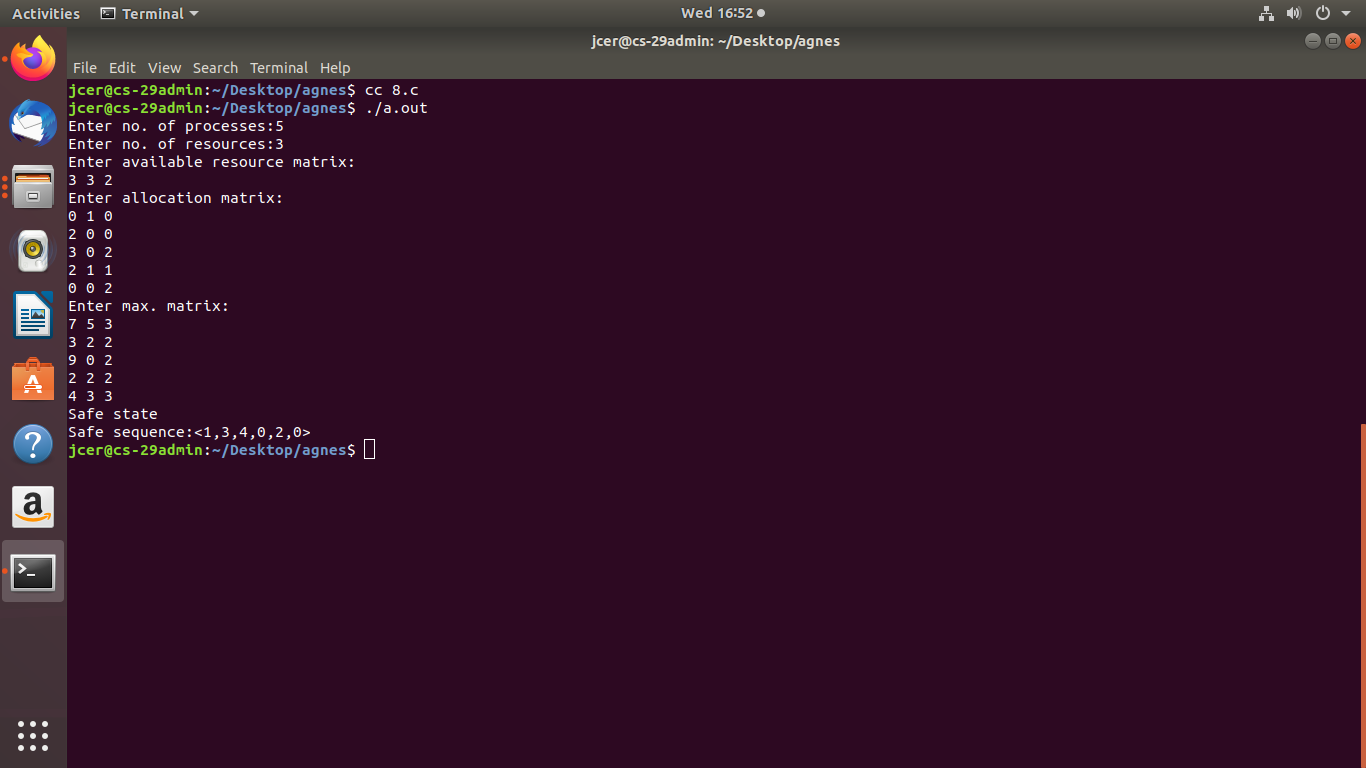
}

}

printf("Safe state\n");

display();

}



**Program 9a**

#include<stdio.h>

int main()

{

int i, j, n, a[50], frame[10], no, k, avail, count=0, hit;

printf("Enter the number of pages : \n");

scanf("%d",&n);

printf("Enter the page number : \n");

for(i=1;i<=n;i++)

scanf("%d",&a[i]);

printf("Enter the number of frames :\n");

scanf("%d",&no);

for(i=0;i<no;i++)

frame[i]=-1;

j=0;

printf("Reference string \t Page Frames\n");

for(i=1;i<=n;i++)

{

printf("%d\t\t\t",a[i]);

avail=0;

for(k=0;k<no;k++)

if(frame[k]==a[i])

avail=1;

if(avail==0)

{

frame[j]=a[i];

j=(j+1)%no;

count++;

for(k=0;k<no;k++)

printf("%d\t",frame[k]);

}

printf("\n");

}

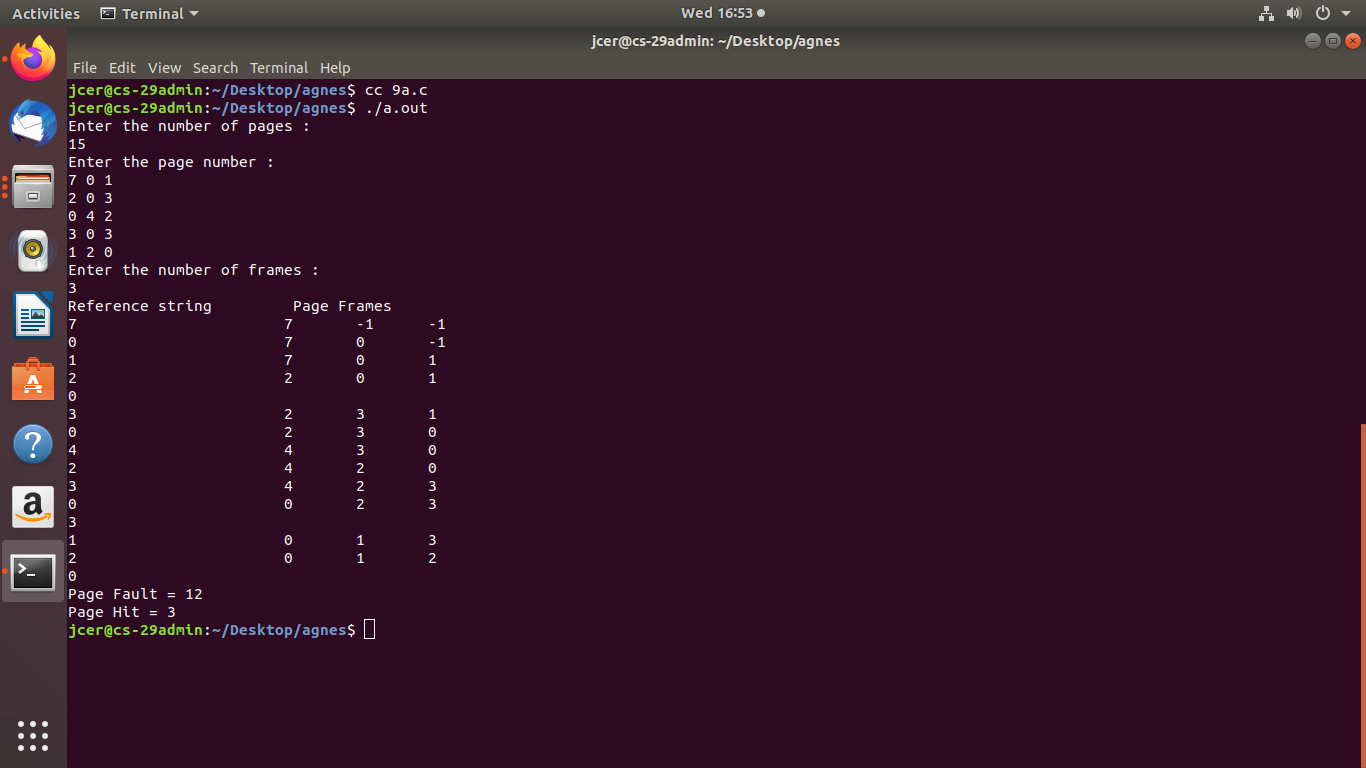
printf("Page Fault = %d\n",count);

hit=n-count;

printf("Page Hit = %d\n",hit);

return 0;

}



**Program 9b**

#include<stdio.h>

int main()

{

int q[20],p[50],c=0,c1,d,f,i,j,k=0,n,r,t,b[20],c2[20];

printf("Enter no of pages:");

scanf("%d",&n);

printf("Enter the refrences string:");

for(i=0;i<n;i++)

scanf("%d",&p[i]);

printf("Enter no of frames:");

scanf("%d",&f);

p[i]-f;

q[k]=p[k];

printf("\n\t%d\n",q[k]);

c++;

k++;

for(i=1;i<n;i++)

{

c1=0;

for(j=0;j<f;j++)

{

if(p[i]!=q[j])

c1++;

}

if(c1==f)

{

c++;

if(k<f)

{

q[k]=p[i];

k++;

for(j=0;j<k;j++)

printf("\t%d",q[j]);

printf("\n");

}else

{

for(r=0;r<f;r++)

{

c2[r]=0;

for(j=i-1;j<n;j--)

{

if(q[r]!=p[j])

c2[r]++;

else

break;

}

}

for(r=0;r<f;r++)

b[r]=c2[r];for(r=0;r<f;r++)

{

for(j=r;j<f;j++)

{

if(b[r]<b[j])

{

t=b[r];

b[r]=b[j];

b[j]=t;

}

}

}

for(r=0;r<f;r++)

{

if(c2[r]==b[0])

q[r]=p[i];

printf("\t%d",q[r]);

}

printf("\n");

}

}

}

printf("\n The no of page faults is %d",c);

}

