**Code**

#include<bits/stdc++.h>

using namespace std;

void CheckReflexive(char \*s1,char \*s2)

{

int c=0,j=0;

for(int i=0;i<strlen(s2);i=i+2)

if(s2[i]==s1[j])

if(s2[i+1]==s1[j])

{

c++;

j++;

continue;

}

if(j==strlen(s1))

cout<<"aRb = Reflexive"<<endl;

else

cout<<"aRb = Not Reflexive"<<endl;

}

void CheckSymmetric(char \*s2)

{

char temp\_a,temp\_b;

int j=0,i,k,flag=0,ASFlagCount=0;

for(i=0;i<strlen(s2);i=i+2)

{

flag = 0;

temp\_a = s2[j];

temp\_b = s2[j+1];

j=j+2;

for(k=0;k<strlen(s2)&&temp\_a!=temp\_b;k=k+2)

if(s2[k]==temp\_b)

if(s2[k+1]==temp\_a)

flag = 1;

if(flag == 0 && (temp\_a != temp\_b))

ASFlagCount++;

}

if(ASFlagCount != 0) {

cout<<"aRb = Not Symmetric"<<endl;

return;

}

if(flag == 1)

cout<<"aRb = Symmetric"<<endl;

}

void CheckAntiSymmetric(char \*s2)

{

char temp\_a,temp\_b;

int j=0,i,k,flag=0;

for(i=0;i<strlen(s2);i=i+2)

{

temp\_a = s2[j];

temp\_b = s2[j+1];

j=j+2;

for(k=0;k<strlen(s2)&&temp\_a!=temp\_b;k=k+2)

if(s2[k]==temp\_b)

if(s2[k+1]==temp\_a)

flag = 1;

}

if(flag != 1)

cout<<"aRb = Anti-symmetric"<<endl;

else

cout<<"aRb = Not Anti-symmetric"<<endl;

}

bool pair\_exist(char left, char right, char \*s2, int len2)

{

for (int i=0; i<len2; i+=2)

{

if (left==s2[i] && right==s2[i+1]) return true;

}

return false;

}

bool transitive(char \*s2, int len1)

{

for (int i=0; i<len1; i+=2)

{

char e = s2[i];

char f = s2[i+1];

for (int j=0; j<len1; j+=2)

{

if (i == j)

continue;

if (s2[j] != f)

continue;

if (!pair\_exist(e, s2[j+1], s2, len1))

return false;

}

}

}

void CheckTransitive(char \*s2,int l)

{

if((transitive(s2,l))==true)

cout<<"aRb = Transitive"<<endl;

else

cout<<"aRb = Not Transitive"<<endl;

}

void menu(char \*s1, char \*s2)

{

int choice,flag=0;

int len = strlen(s2);

cout<<"\n1. Reflexive Test\n2. Symmetric Test\n3. Anti-symmetric Test\n4. Transitive Test\n5. Exit"<<endl;

while(flag == 0)

{

cout<<"\nEnter your choice: ";

cin>>choice;

switch(choice)

{

case 1:

CheckReflexive(s1,s2);

break;

case 2:

CheckSymmetric(s2);

break;

case 3:

CheckAntiSymmetric(s2);

break;

case 4:

CheckTransitive(s2,len);

break;

case 5:

cout<<"Program Finished. . . "<<endl;

flag = 1;

break;

}

}

}

int main()

{

int ActLen1,ActLen2,c=0,d=0;

char set1[50],set2[60];

char str1[20],str2[20];

cout<<"Enter Set A: ";

gets(set1);

cout<<"Corresponding Relation ";

gets(set2);

ActLen1=((strlen(set1)-2)/2)+1;

ActLen2=((strlen(set2)-2)/5)\*2;

for(int i=0,j=0;i<strlen(set1);i++)

{

if(j==ActLen1){

str1[j]='\0';

break;

}

else if(set1[i]!='{' && set1[i]!='}' && set1[i]!=','){

str1[j]=set1[i];

j++;

}

}

for(int i=0,j=0;i<strlen(set2);i++)

{

if(j==ActLen2){

str2[j]='\0';

break;

}

else if(set2[i]!='{' && set2[i]!='}' && set2[i]!=',' && set2[i]!='(' && set2[i]!=')'){

str2[j]=set2[i];

j++;

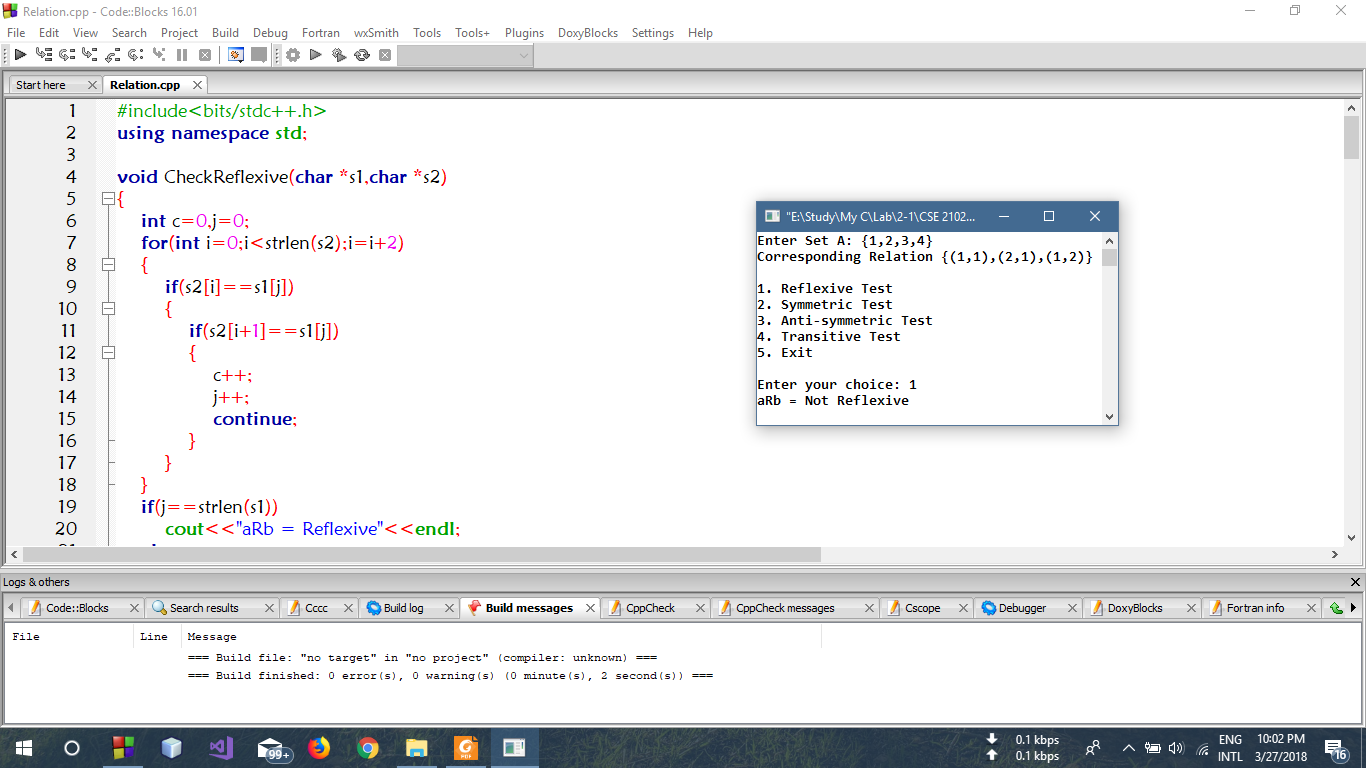
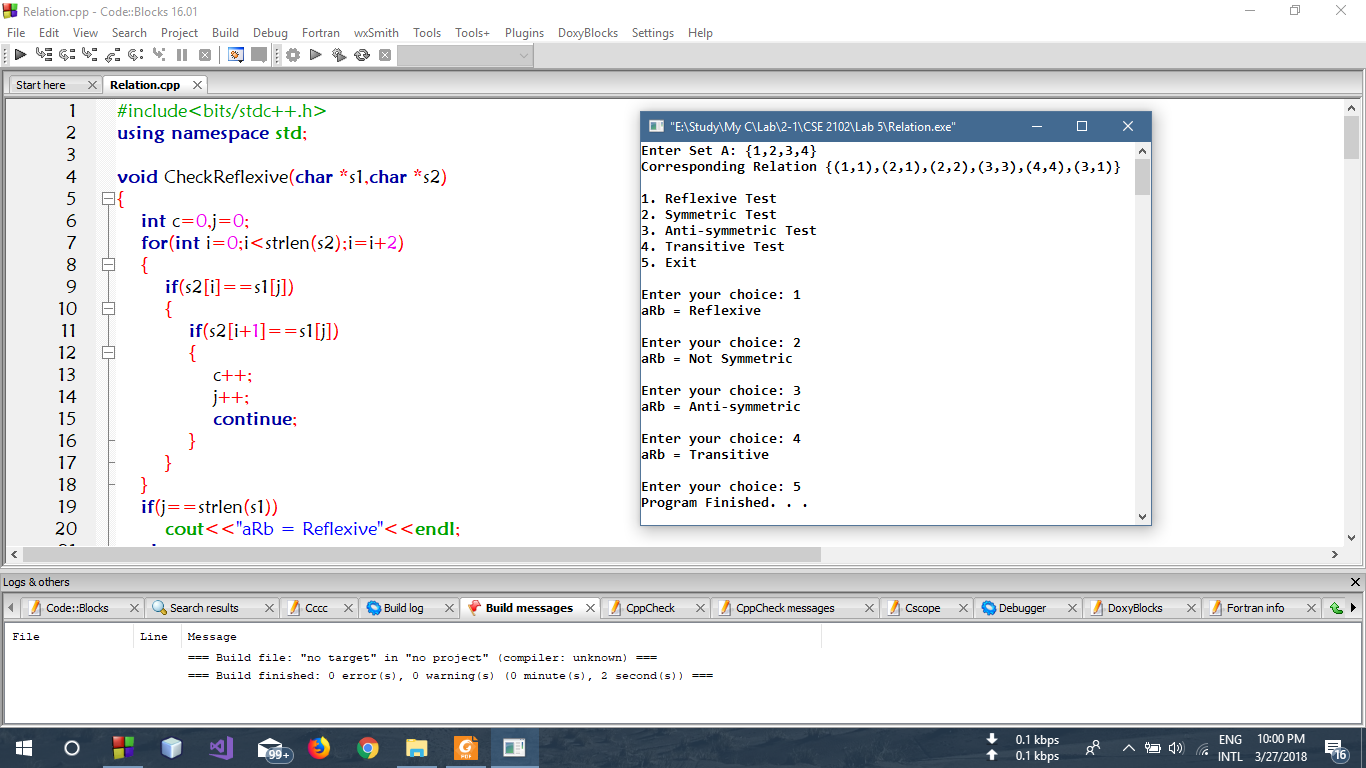
}

}

menu(str1,str2);

return 0;

}

**Output**

