**Assignment # 02:**

**Discerete Structures:**

**Usman Manzoor**

**19P-0068**

**Section-CS-19(B)**

**#include <iostream>**

**using namespace std;**

**void preposition(string p,string pt)**

**{**

**if(pt=="implication")**

**{**

**if(p[0]!='~' && p[3]!='~')**

**{**

**cout<<"converse is: "<<p[3]<<p[1]<<p[2]<<p[0]<<endl;**

**cout<<"contrapositive is: "<<"~"<<p[3]<<p[1]<<p[2]<<"~"<<p[0]<<endl;**

**cout<<"inverse is: "<<"~"<<p[0]<<p[1]<<p[2]<<"~"<<p[3]<<endl;**

**}**

**else if(p[0]=='~' && p[4]!='~')**

**{**

**cout<<"converse is: "<<p[4]<<p[2]<<p[3]<<p[0]<<p[1]<<endl;**

**cout<<"contrapositive is: "<<p[0]<<p[4]<<p[2]<<p[3]<<p[1]<<endl;**

**cout<<"inverse is: "<<p[1]<<p[2]<<p[3]<<p[0]<<p[4]<<endl;**

**}**

**else if(p[0]!='~' && p[3]=='~')**

**{**

**cout<<"converse is: "<<p[3]<<p[4]<<p[1]<<p[2]<<p[0]<<endl;**

**cout<<"contrapositive is: "<<p[4]<<p[1]<<p[2]<<p[3]<<p[0]<<endl;**

**cout<<"inverse is: "<<p[3]<<p[0]<<p[1]<<p[2]<<p[4]<<endl;**

**}**

**else if(p[0]=='~' && p[4]=='~')**

**{**

**cout<<"converse is: "<<p[4]<<p[5]<<p[2]<<p[3]<<p[0]<<p[1]<<endl;**

**cout<<"contrapositive is: "<<p[5]<<p[2]<<p[3]<<p[0]<<endl;**

**cout<<"inverse is: "<<p[0]<<p[2]<<p[3]<<p[5]<<endl;**

**}**

**}**

**if(pt=="inverse")**

**{**

**if(p[0]!='~' && p[3]!='~')**

**{**

**cout<<"implication is: "<<"~"<<p[0]<<p[1]<<p[2]<<"~"<<p[3]<<endl;**

**cout<<"contrapositive is: "<<p[3]<<p[1]<<p[2]<<p[0]<<endl;**

**cout<<"converse is: "<<"~"<<p[3]<<p[1]<<p[2]<<"~"<<p[0]<<endl;**

**}**

**else if(p[0]=='~' && p[4]!='~')**

**{**

**cout<<"implication is: "<<p[1]<<p[2]<<p[3]<<p[0]<<p[4]<<endl;**

**cout<<"contrapositve is: "<<p[4]<<p[2]<<p[3]<<p[0]<<p[1]<<endl;**

**cout<<"converse is: "<<p[0]<<p[4]<<p[2]<<p[3]<<p[1]<<endl;**

**}**

**else if(p[0]!='~' && p[3]=='~')**

**{**

**cout<<"implication is: "<<p[3]<<p[0]<<p[1]<<p[2]<<p[4]<<endl;**

**cout<<"contrapositive is: "<<p[3]<<p[4]<<p[1]<<p[2]<<p[0]<<endl;**

**cout<<"converse is: "<<p[4]<<p[1]<<p[2]<<p[3]<<p[0]<<endl;**

**}**

**else if(p[0]=='~' && p[4]=='~')**

**{**

**cout<<"implication is: "<<p[1]<<p[2]<<p[3]<<p[5]<<endl;**

**cout<<"contrapositive is: "<<p[4]<<p[5]<<p[2]<<p[3]<<p[0]<<p[1]<<endl;**

**cout<<"converse is:**

**"<<p[5]<<p[2]<<p[3]<<p[1]<<endl;**

**}**

**}**

**if(pt=="converse")**

**{**

**if(p[0]!='~' && p[3]!='~')**

**{**

**cout<<"implication is: "<<p[3]<<p[1]<<p[2]<<p[0]<<endl;**

**cout<<"contrapositive is: "<<"~"<<p[0]<<p[1]<<p[2]<<"~"<<p[3]<<endl;**

**cout<<"inverse is: "<<"~"<<p[3]<<p[1]<<p[2]<<"~"<<p[0]<<endl;**

**}**

**else if(p[0]=='~' && p[4]!='~')**

**{**

**cout<<"implification is: "<<p[0]<<p[4]<<p[2]<<p[3]<<p[1]<<endl;**

**cout<<"contrapositive is: "<<p[0]<<p[1]<<p[2]<<p[3]<<p[4]<<endl;**

**cout<<"inverse is: "<<p[4]<<p[2]<<p[3]<<p[0]<<p[1]<<endl;**

**}**

**else if(p[0]!='~' && p[3]=='~')**

**{**

**cout<<"implication is: "<<p[3]<<p[4]<<p[1]<<p[2]<<p[0]<<endl;**

**cout<<"contrapositive is: "<<p[3]<<p[0]<<p[1]<<p[2]<<p[4]<<endl;**

**cout<<"inverse is: "<<p[4]<<p[1]<<p[2]<<p[3]<<p[0]<<endl;**

**}**

**else if(p[0]=='~' && p[4]=='~')**

**{**

**cout<<"implication is: "<<p[0]<<p[1]<<p[2]<<p[3]<<p[4]<<p[5]<<endl;**

**cout<<"contrapositive is: "<<p[1]<<p[2]<<p[3]<<p[5]<<endl;**

**cout<<"inverse is: "<<p[5]<<p[2]<<p[3]<<p[1]<<endl;**

**}**

**}**

**if(pt=="contrapositive")**

**{**

**if(p[0]!='~' && p[3]!='~')**

**{**

**cout<<"implication is: "<<"~"<<p[3]<<p[1]<<p[2]<<"~"<<p[0]<<endl;**

**cout<<"inverse is: "<<p[3]<<p[1]<<p[2]<<p[0]<<endl;**

**cout<<"converse is: "<<"~"<<p[0]<<p[1]<<p[2]<<"~"<<p[3]<<endl;**

**}**

**else if(p[0]=='~' && p[4]!='~')**

**{**

**cout<<"implification is: "<<p[0]<<p[4]<<p[2]<<p[3]<<p[1]<<endl;**

**cout<<"inverse is: "<<p[4]<<p[2]<<p[3]<<p[0]<<p[1]<<endl;**

**cout<<"converse is: "<<p[0]<<p[1]<<p[2]<<p[3]<<p[4]<<endl;**

**}**

**else if(p[0]!='~' && p[3]=='~')**

**{**

**cout<<"implication is: "<<p[4]<<p[1]<<p[2]<<p[3]<<p[0]<<endl;**

**cout<<"inverse is: "<<p[3]<<p[4]<<p[1]<<p[2]<<p[0]<<endl;**

**cout<<"converse is: "<<p[3]<<p[0]<<p[1]<<p[2]<<p[4]<<endl;**

**}**

**else if(p[0]=='~' && p[4]=='~')**

**{**

**cout<<"implication is: "<<p[5]<<p[2]<<p[3]<<p[1]<<endl;**

**cout<<"inverse is: "<<p[0]<<p[5]<<p[2]<<p[3]<<p[4]<<p[1]<<endl;**

**cout<<"converse is: "<<p[1]<<p[2]<<p[3]<<p[5]<<endl;**

**}**

**}**

**}**

**int main()**

**{**

**string pt;**

**string p;**

**-**

**cout<<"Enter preposition type: ";**

**cin>>pt;**

**cout<<endl;**

**cout<<"Enter preposition: ";**

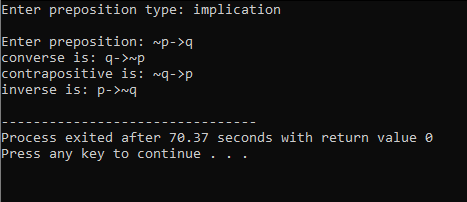
**cin>>p;**

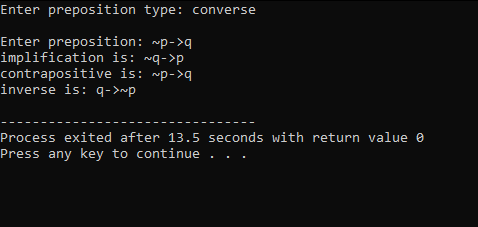
**preposition(p,pt);**

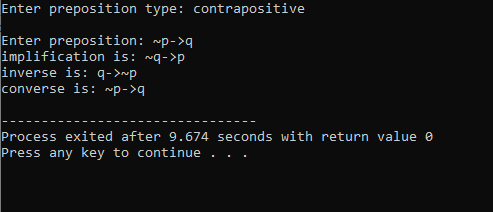
**return 0;**

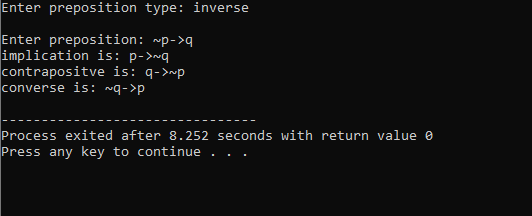
**}**

**Outputs:**

****

****

****

****