**PERSONALITY QUIZ**

**ACKNOWLEDGEMENTS**

We would like to thank Dhanesh sir for his sincere helping hand and support.

We would also like to thank Shri. G Devan for providing us the oppurtunity to do this project, and our friends and teachers for being a sturdy pillar of support even when the going got tough.

**CONTENTS**

1. INTRODUCTION -4
2. SYNOPSIS -5
3. HEADER FILES USED -6
4. PROGRAM SOURCE CODE -7
5. PROGRAM SAMPLE OUTPUT -51
6. BIBLIOGRAPHY -82

**Introduction**

When Dhanesh Sir first told us about doing our own computer project, the first thing that came to our minds was a personality quiz, since we felt that the concept of different personalities was underrated and not well known. We had a theory that 16 types of personalities existed in this world, and any person would fit into one of these or very rarely, be a hybrid. We were quite sceptical at first when we thought about taking it as a project. There were many challenges facing us: Would people even take our 25 questions quiz? Will they be interested in our theory and knowing about their own type?

There were a plethora of advantages of knowing a person’s (true) type, of course. First, the thrill of reading about your type and thinking, “Wow, the description is perfect!” (Or “This baloney would have Socrates facepalming from his grave!”, of course. Nothing is perfect.). Knowing a person’s type would help them figure out their weaknesses and thus lead to more self awareness. Knowing about their close ones’ type could help them in knowing what they dislike, and result in improving the bond between them. Also, many rare types could be heartened that people similar to them exist in this world we live in.

In the end, we decided to do it. 40 pages of programming ended in what we hope is an accurate, humorous program that will help you understand more about yourself. We sincerely hope that our program delights you and helps you evaluate yourself.

**SYNOPSIS**

The aim of our project is to create a user-friendly personality quiz, that accurately displays your Personality type(which comes in the form of four letters). Also, the result of the quiz includes an introduction to your type, strengths, weaknesses, famous people of your type and the rough percentage of population of your type. We have created a user-friendly program that allows you to take our quiz, read our theory, see the results of the people who have already taken the quiz, read about the different types and see the credits. We hope sincerely that our quiz is of great benefit to you.

**HEADER FILES USED**

1. #include<iostream.h> -for cout and cin
2. #include<conio.h> -for clrscr() abd getch() functions
3. #include<string.h> -for strcpy() function
4. #include<stdio.h> -for gets() function
5. #include<fstream.h> -to read and write into text and binary files

**PROGRAM**

**SOURCE**

**CODE**

#include<iostream.h>

#include<conio.h>

#include<string.h>

#include<stdio.h>

#include<fstream.h>

class participant

{

public:

char name[100];

char type[4];

int a[25];

char study[4];

void display()

{

cout<<name<<"--"<<type<"\n";

}

}p;

void main()

{

int v1,v6,var;

int i=0,I=0,E=0,N=0,S=0,P=0,J=0,F=0,T=0;

char t17[1000];

char s1[20],s2[20],s3[20],s4[20];

strcpy(s1," ");

strcpy(s2," ");

strcpy(s3," ");

strcpy(s4," ");

ifstream iff1("quest.txt");

label27:

{

clrscr();

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PERSONALITY QUIZ\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n BY SRILAKSHMI AJITH(SRI)";

cout<<"\n AND IYER RAJKUMAR VARSHA(VIOLET)";

cout<<"\n\n\n1. Take the quiz\n2. Our Theory\n3. See saved results\n4. Read about different types\n5. Credits";

cout<<"\n\nEnter Choice : ";

cin>>v1;

goto label28;

}

label28:

{

ifstream formal("SR.dat",ios::app);

ifstream iff2("INTJ.txt");

ifstream iff3("INTP.txt");

ifstream iff5("ENTP.txt");

ifstream iff8("ENFJ.txt");

ifstream iff9("ENFP.txt");

ifstream iff10("ISTJ.txt");

ifstream iff11("ISFJ.txt");

ifstream iff12("ESTJ.txt");

ifstream iff14("ISTP.txt");

ifstream iff16("ESTP.txt");

ifstream iff17("ESFP.txt");

iff1.seekg(0);

iff2.seekg(0);

iff3.seekg(0);

iff5.seekg(0);

iff8.seekg(0);

iff9.seekg(0);

iff10.seekg(0);

iff11.seekg(0);

iff12.seekg(0);

iff14.seekg(0);

iff16.seekg(0);

iff17.seekg(0);

clrscr();

switch(v1)

{

case 1:

{

char s[800];

cout<<"\nEnter your name:";

gets(p.name);

cout<<"Press Enter to continue";

getch();

clrscr();

cout<<"\nA Word of Caution!";

cout<<"\nRemember! No personality type is superior to another. Each type has their own \npros and cons. Also, seeing yourself as you'd like to be rather than as you \nactually are will skew results immensely. Answer honestly, even if you don't \nlike the answer.";

cout<<"\nPress Enter to continue";

getch();

clrscr();

iff1.getline(s,800,'$');

cout<<s;

label1:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

N=N+3;

break;

}

case 2:

{

N=N+2;

break;

}

case 3:

{

S=S+3;

break;

}

case 4:

{

S=S+2;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label1;

} //default close

} //switch close

} //label1 close

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label2:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

T=T+2;

break;

}

case 2:

{

F=F+2;

break;

}

case 3:

{

T=T+1;

break;

}

case 4:

{

F=F+1;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label2;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label3:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

P=P+4;

break;

}

case 2:

{

J=J+4;

break;

}

case 3:

{

P=P+2;

break;

}

case 4:

{

J=J+2;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label3;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label4:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

S=S+4;

break;

}

case 2:

{

N=N+1;

break;

}

case 3:

{

N=N+4;

break;

}

case 4:

{

S=S+2;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label4;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label5:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

I=I+4;

break;

}

case 2:

{

I=I+1;

break;

}

case 3:

{

E=E+4;

break;

}

case 4:

{

E=E+2;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label5;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label6:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

S=S+4;

P=P+2;

break;

}

case 2:

{

N=N+4;

J=J+2;

break;

}

case 3:

{

N=N+2;

J=J+1;

break;

}

case 4:

{

S=S+2;

P=P+1;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label6;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label7:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

J=J+4;

break;

}

case 2:

{

strcpy(s1,"Perfectionistic");

break;

}

case 3:

{

P=P+2;

break;

}

case 4:

{

P=P+4;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label7;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label8:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

T=T+4;

break;

}

case 2:

{

F=F+4;

break;

}

case 3:

{

F=F+3;

break;

}

case 4:

{

T=T+2;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label8;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label9:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

S=S+4;

break;

}

case 2:

{

S=S+2;

break;

}

case 3:

{

N=N+2;

break;

}

case 4:

{

N=N+4;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label9;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label10:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

strcpy(p.study,"NY");

break;

}

case 2:

{

strcpy(p.study,"YN");

break;

}

case 3:

{

strcpy(p.study,"YY");

break;

}

case 4:

{

strcpy(p.study,"NN");

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label10;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label11:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

E=E+4;

break;

}

case 2:

{

I=I+2;

break;

}

case 3:

{

I=I+4;

break;

}

case 4:

{

E=E+4;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label11;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label12:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

S=S+4;

break;

}

case 2:

{

N=N+4;

break;

}

case 3:

{

S=S+2;

break;

}

case 4:

{

N=N+2;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label12;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label13:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

E=E+4;

break;

}

case 2:

{

E=E+2;

break;

}

case 3:

{

I=I+2;

break;

}

case 4:

{

I=I+4;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label13;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label14:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

F=F+4;

break;

}

case 2:

{

T=T+4;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label14;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label15:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

P=P+4;

break;

}

case 2:

{

J=J+4;

break;

}

case 3:

{

J=J+2;

break;

}

case 4:

{

P=P+2;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label15;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label16:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

J=J+2;

break;

}

case 2:

{

J=J+1;

T=T+2;

break;

}

case 3:

{

F=F+2;

break;

}

case 4:

{

F=F+2;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label16;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label17:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

F=F+4;

break;

}

case 2:

{

T=T+4;

break;

}

case 3:

{

F=F+4;

break;

}

case 4:

{

T=T+2;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label17;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label18:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

J=J+4;

break;

}

case 2:

{

P=P+4;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label18;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label19:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

I=I+4;

break;

}

case 2:

{

I=I+1;

strcpy(s4,"Competitive");

break;

}

case 3:

{

E=E+4;

break;

}

case 4:

{

E=E+2;

F=F+2;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label19;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label20:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

T=T+4;

break;

}

case 2:

{

T=T+3;

strcpy(s2,"Arrogant");

break;

}

case 3:

{

I=I+2;

break;

}

case 4:

{

F=F+2;

strcpy(s3,"Turbulent");

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label20;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label21:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

I=I+4;

T=T+4;

break;

}

case 2:

{

E=E+4;

break;

}

case 3:

{

I=I+2;

F=F+2;

break;

}

case 4:

{

T=T+4;

strcpy(s2,"Arrogant");

break;

}

case 5:

{

E=E+4;

strcpy(s2,"Arrogant");

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label21;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label22:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

F=F+4;

break;

}

case 2:

{

F=F+2;

break;

}

case 3:

{

T=T+2;

break;

}

case 4:

{

T=T+4;

break;

}

case 5:

{

E=E+4;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label22;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label23:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

E=E+4;

break;

}

case 2:

{

I=I+2;

break;

}

case 3:

{

E=E+2;

break;

}

case 4:

{

I=I+4;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label23;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label24:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

T=T+4;

strcpy(s2,"Arrogant");

break;

}

case 2:

{

T=T+2;

break;

}

case 3:

{

F=F+2;

break;

}

case 4:

{

F=F+4;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label24;

}

}

}

clrscr();

i++;

iff1.getline(s,800,'$');

cout<<s;

label25:

{

cout<<"\n Your answer is : ";

cin>>p.a[i];

switch(p.a[i])

{

case 1:

{

N=N+4;

break;

}

case 2:

{

S=S+4;

break;

}

default:

{

cout<<"\n Wrong choice! Enter again";

goto label25;

}

}

}

int x1,x2,x3,x4;

if(E>I)

x1=1;

else if(I>E)

x1=2;

else if(I==E)

{

if((p.a[12]==1)||(p.a[12]==2))

x1=1;

else if((p.a[12]==3)||(p.a[12]==4))

x1=2;

}

if(S>N)

x2=1;

else if(N>S)

x2=2;

else if(N==S)

{

if((p.a[5]==1)||(p.a[5]==4))

x2=1;

else if((p.a[5]==3)||(p.a[5]==2))

x2=2;

}

if(F>T)

x3=1;

else if(T>F)

x3=2;

else if(T==F)

{

if((p.a[21]==1)||(p.a[21]==2))

x3=1;

else if((p.a[21]==3)||(p.a[21]==4))

x3=2;

}

if(P>J)

x4=1;

else if(J>P)

x4=2;

else if(P==J)

{

if(p.a[17]==2)

x4=1;

else if(p.a[17]==1)

x4=2;

}

if((x1==1)&&(x2==1)&&(x3==1)&&(x4==1))

{strcpy(p.type,"ESFP");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff17.getline(t17,500,'$'))

{cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

if((x1==1)&&(x2==1)&&(x3==2)&&(x4==1))

{strcpy(p.type,"ESTP");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff16.getline(t17,500,'$'))

{ cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

if((x1==2)&&(x2==1)&&(x3==1)&&(x4==1))

{strcpy(p.type,"ISFP");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nISFP: The Artists";

cout<<"\nISFPs are the true artists, though not necessarily in the literal sense.";

cout<<"\nThey use aesthetics and design to pop societal conventions.";

cout<<"\nThey can be very laid back, and are happy to be who they are.";

cout<<"\nSentence:Don't box me in!";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nStrengths:";

cout<<"\n1.Sensitive to others: They can easily relate to others' emotions.";

cout<<"\n2.Imaginative and creative";

cout<<"\n3.Passionate: Beneath the quiet shyness beats an intensely feeling heart.";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nWeaknesses:";

cout<<"\n1.They actively avoid/are unable to plan for the future";

cout<<"\n2.They can be easily stressed";

cout<<"\n3.They have fluctuating self esteem if their efforts are dismissed.";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nISFPs you may know:";

cout<<"\n1.Michael Jackson";

cout<<"\n2.Britney Spears";

cout<<"\n3.Avril Lavigne";

cout<<"\n4.Beatrix Kiddo(Kill Bill)";

cout<<"\n5.Jesse Pinkman(Breaking Bad)";

cout<<"\nPercentage of population:8.8%";

}

if((x1==2)&&(x2==1)&&(x3==2)&&(x4==1))

{strcpy(p.type,"ISTP");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff14.getline(t17,500,'$'))

{cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

if((x1==1)&&(x2==1)&&(x3==1)&&(x4==2))

{strcpy(p.type,"ESFJ");

cout<<"\nYour result is : "<<p.type;

cout<<"\nESFJ: The Consul";

cout<<"\nThey are extraordinarily caring people who are always eager to help.";

cout<<"\nThey are warm hearted, popular and tend to put needs of others over their";

cout<<"\nown needs.";

cout<<"\nThey value tradition and security.";

cout<<"\nSentence:Lift, encourage and strengthen each other! Let's all share the";

cout<<"\npositiveness spread.";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nStrengths:";

cout<<"\n1.They are extrememly loyal,";

cout<<"\n2.They are good at connecting with others.";

cout<<"\n3.They are sensitive and warm.";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nWeaknesses:";

cout<<"\n1.They worry about their social status";

cout<<"\n2.They are reluctant to innovate or improvise.";

cout<<"\n3.They are vulnerable to criticism ";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nESFJs you may know:";

cout<<"\n1.Bill Clinton";

cout<<"\n2.Taylor Swift";

cout<<"\n3.Jennifer Lopez";

cout<<"\n4.Sansa Stark(Game of thrones)";

cout<<"\n5.Monica(FRIENDS)";

cout<<"\nPercentage of population:12.3%";

}

if((x1==1)&&(x2==1)&&(x3==2)&&(x4==2))

{strcpy(p.type,"ESTJ");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff12.getline(t17,500,'$'))

{ cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

if((x1==2)&&(x2==1)&&(x3==1)&&(x4==2))

{strcpy(p.type,"ISFJ");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff11.getline(t17,500,'$'))

{

cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

if((x1==2)&&(x2==1)&&(x3==2)&&(x4==2))

{strcpy(p.type,"ISTJ");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff10.getline(t17,500,'$'))

{cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

if((x1==1)&&(x2==2)&&(x3==1)&&(x4==1))

{strcpy(p.type,"ENFP");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff9.getline(t17,500,'$'))

{cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

if((x1==1)&&(x2==2)&&(x3==1)&&(x4==2))

{strcpy(p.type,"ENFJ");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff8.getline(t17,500,'$'))

{cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

if((x1==2)&&(x2==2)&&(x3==1)&&(x4==1))

{strcpy(p.type,"INFP");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nINFP: The Writers and Creators";

cout<<"\nThey are true idealists.";

cout<<"\nThey have a high risk of being misunderstood by people.";

cout<<"\nThey have a talent for self-expression, revelaing themselves through";

cout<<"\n metaphors and fictional characters.";

cout<<"\nThey listen to many, but talk to few.";

cout<<"\nSentence:Mostly misunderstood and alone.";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nStrengths:";

cout<<"\n1.Seek and value harmony";

cout<<"\n2.Open minded and flexible";

cout<<"\n3.Very creative due to their intuitive nature.";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nWeaknesses:";

cout<<"\n1.Altruistic: They sometimes see themselves as too selfish only because";

cout<<"\nthey want to give much more than they are able to.";

cout<<"\n2.Difficult to get to know";

cout<<"\n3.Dislike dealing with data";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nINFPs you may know:";

cout<<"\n1.J.R.R Tolkein";

cout<<"\n2.William Shakespeare";

cout<<"\n3.Johnny Depp";

cout<<"\n4.Frodo Baggins(Lord of the rings)";

cout<<"\n5.Tom Hiddleson";

cout<<"\nPercentage of population:4.4%";

}

if((x1==2)&&(x2==2)&&(x3==1)&&(x4==2))

{strcpy(p.type,"INFJ");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nINFJ: Yin-yang enforcer";

cout<<"\nSuch types have an inborn sense of idealism and morality. ";

cout<<"\nThey are dreamers, but are capable of taking concrete steps to realise";

cout<<"\ntheir dreams.";

cout<<"\nThey share a very unique combination of traits: Soft spoken yet have ";

cout<<"\nstrong opinions which they will fight tirelessly for.";

cout<<"\nKarma is a very attractive idea to INFJs and they tend to act not for ";

cout<<"\npersonal gain, but to create balance.";

cout<<"\nSentence:The world is a place full of inequity!";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nStrengths:";

cout<<"\n1.They are inspiring and convincing.";

cout<<"\n2.Determined and passionate";

cout<<"\n3.They truly act just to make the world a better place.";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nWeaknesses:";

cout<<"\n1.Sensitive and highly vulnerable to criticism.";

cout<<"\n2.They need to remember to take care of themselves while making the";

cout<<"\nworld better.";

cout<<"\n3.Extremely private: Trusting a new friend can even be a challenge.";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nINFJs you might know:";

cout<<"\n1.Martin Luther King";

cout<<"\n2.Nelson Mandela";

cout<<"\n3.Mother Teresa";

cout<<"\n4.Jon Snow(Game of Thrones)";

cout<<"\n5.Aragorn/Strider(Lord of the rings)";

cout<<"\nPercentage of population:1.5%, rarest.";

}

if((x1==1)&&(x2==2)&&(x3==2)&&(x4==1))

{strcpy(p.type,"ENTP");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff5.getline(t17,500,'$'))

{cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

if((x1==1)&&(x2==2)&&(x3==2)&&(x4==2))

{strcpy(p.type,"ENTJ");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nENTJ: The commander";

cout<<"\nRuthless. Challenging. Anything but timid. That's what ENTJs are.";

cout<<"\nThey project authority, charisma and confidence like a light-emitting ";

cout<<"\ndiode.(No nerdism intended)";

cout<<"\nThey love any sort of challenge and believe that given enough time and";

cout<<"\n resources, they can achieve any goal.";

cout<<"\nENTJs form many of the businesses and institutions we take for";

cout<<"\ngranted everyday.";

cout<<"\nSentence:My way or the highway!";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nStrengths:";

cout<<"\n1.Highly effficient";

cout<<"\n2.Self confident: they trust their abilities.";

cout<<"\n3.Strong willed: They never give up if the going gets tough.";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nWeaknesses:";

cout<<"\n1.They can be quite intolerant at times. ";

cout<<"\n2.Cold and ruthless: They can be extraordinarily insensitive. Like asking";

cout<<"\nfor the wifi password at a funeral sort of insensitive.";

cout<<"\n3.Arrogant: They may look down on those they deem inferior.";

cout<<"\n(That's a lot of people.)";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nENTJs you may know:";

cout<<"\n1.Steve Jobs";

cout<<"\n2.Gordon Ramsay";

cout<<"\n3.Francis J Underwood(House of cards)";

cout<<"\n4.Milady De Winter(The three musketeers)";

cout<<"\n5.Margaret Thatcher";

cout<<"\nPercentage of population:2%";

}

if((x1==2)&&(x2==2)&&(x3==2)&&(x4==1))

{

strcpy(p.type,"INTP");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff3.getline(t17,500,'$'))

{cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

if((x1==2)&&(x2==2)&&(x3==2)&&(x4==2))

{

strcpy(p.type,"INTJ");

cout<<"\nYour result is : "<<p.type;

cout<<"\nPress enter to continue";

getch();

clrscr();

while(iff2.getline(t17,500,'$'))

{

cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

}

cout<<"\nBonus Points:(if any:)";

cout<<"\n"<<s1<<"\n";

cout<<s2<<"\n";

cout<<s3<<"\n";

cout<<s4<<"\n";

ofstream formal("SR.dat",ios::app);

formal.write((char\*)&p,sizeof(p));

cout<<"\nPress Enter to go to the main menu";

getch();

goto label27;

}

case 2:

{

ifstream iff18("thor.txt");

iff18.open("thory.txt");

while(iff18.getline(t17,1000,'$'))

{

cout<<t17;

cout<<"\nPress enter to continue";

getch();

clrscr();

}

cout<<"\nPress Enter to go to the main menu";

getch();

goto label27;

}

case 3:

{

while(formal.read((char\*)&p,sizeof(p)))

{

p.display();

cout<<"\nPress Enter to continue\n";

getch();

}

cout<<"\nPress Enter to go to main menu";

getch();

clrscr();

goto label27;

}

case 4:

{

label26:

{

clrscr();

cout<<"\nThere are sixteen types in total. \nThe Analysts:";

cout<<"\n1.INTJ\n2.INTP\n3.ENTJ\n4.ENTP";

cout<<"\nThe explorers:\n5.ESFP\n6.ISFP\n7.ESTP\n8.ISTP";

cout<<"\nThe sentinels:\n9.ISFJ\n10.ESFJ\n11.ISTJ\n12.ESTJ";

cout<<"\nThe Diplomats:\n13.INFP\n14.INFJ\n15.ENFJ\n16.ENFP";

cout<<"\nEnter number of which type you want to read about:";

cin>>v6;

clrscr();

switch(v6)

{

case 1:

{

while(iff2.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

case 2:

{

while(iff3.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

case 3:

{

cout<<"\nENTJ: The commander";

cout<<"\nRuthless. Challenging. Anything but timid. That's what ENTJs are.";

cout<<"\nThey project authority, charisma and confidence like a light-emitting ";

cout<<"\ndiode.(No nerdism intended)";

cout<<"\nThey love any sort of challenge and believe that given enough time and";

cout<<"\n resources, they can achieve any goal.";

cout<<"\nENTJs form many of the businesses and institutions we take for";

cout<<"\ngranted everyday.";

cout<<"\nSentence:My way or the highway!";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nStrengths:";

cout<<"\n1.Highly effficient";

cout<<"\n2.Self confident: they trust their abilities.";

cout<<"\n3.Strong willed: They never give up if the going gets tough.";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nWeaknesses:";

cout<<"\n1.They can be quite intolerant at times. ";

cout<<"\n2.Cold and ruthless: They can be extraordinarily insensitive. Like asking";

cout<<"\nfor the wifi password at a funeral sort of insensitive.";

cout<<"\n3.Arrogant: They may look down on those they deem inferior.";

cout<<"\n(That's a lot of people.)";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nENTJs you may know:";

cout<<"\n1.Steve Jobs";

cout<<"\n2.Gordon Ramsay";

cout<<"\n3.Francis J Underwood(House of cards)";

cout<<"\n4.Milady De Winter(The three musketeers)";

cout<<"\n5.Margaret Thatcher";

cout<<"\nPercentage of population:2%";

break;}

case 4:

{

while(iff5.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

case 5:

{

while(iff17.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

case 6:

{

cout<<"\nISFP: The Artists";

cout<<"\nISFPs are the true artists, though not necessarily in the literal sense.";

cout<<"\nThey use aesthetics and design to pop societal conventions.";

cout<<"\nThey can be very laid back, and are happy to be who they are.";

cout<<"\nSentence:Don't box me in!";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nStrengths:";

cout<<"\n1.Sensitive to others: They can easily relate to others' emotions.";

cout<<"\n2.Imaginative and creative";

cout<<"\n3.Passionate: Beneath the quiet shyness beats an intensely feeling heart.";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nWeaknesses:";

cout<<"\n1.They actively avoid/are unable to plan for the future";

cout<<"\n2.They can be easily stressed";

cout<<"\n3.They have fluctuating self esteem if their efforts are dismissed.";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nISFPs you may know:";

cout<<"\n1.Michael Jackson";

cout<<"\n2.Britney Spears";

cout<<"\n3.Avril Lavigne";

cout<<"\n4.Beatrix Kiddo(Kill Bill)";

cout<<"\n5.Jesse Pinkman(Breaking Bad)";

cout<<"\nPercentage of population:8.8%";

break;}

case 7: {

while(iff16.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

case 8:

{

while(iff14.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

case 9:

{

while(iff11.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

case 10:

{

cout<<"\nESFJ: The Consul";

cout<<"\nThey are extraordinarily caring people who are always eager to help.";

cout<<"\nThey are warm hearted, popular and tend to put needs of others over their";

cout<<"\nown needs.";

cout<<"\nThey value tradition and security.";

cout<<"\nSentence:Lift, encourage and strengthen each other! Let's all share the";

cout<<"\npositiveness spread.";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nStrengths:";

cout<<"\n1.They are extrememly loyal,";

cout<<"\n2.They are good at connecting with others.";

cout<<"\n3.They are sensitive and warm.";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nWeaknesses:";

cout<<"\n1.They worry about their social status";

cout<<"\n2.They are reluctant to innovate or improvise.";

cout<<"\n3.They are vulnerable to criticism ";

cout<<"\nPress enter to continue";

getch();

clrscr();

cout<<"\nESFJs you may know:";

cout<<"\n1.Bill Clinton";

cout<<"\n2.Taylor Swift";

cout<<"\n3.Jennifer Lopez";

cout<<"\n4.Sansa Stark(Game of thrones)";

cout<<"\n5.Monica(FRIENDS)";

cout<<"\nPercentage of population:12.3%";

break;}

case 11:

{

while(iff10.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

case 12:

{

while(iff12.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

case 13:

{

cout<<"\nINFP: The Writers and Creators";

cout<<"\nThey are true idealists.";

cout<<"\nThey have a high risk of being misunderstood by people.";

cout<<"\nThey have a talent for self-expression, revelaing themselves through";

cout<<"\n metaphors and fictional characters.";

cout<<"\nThey listen to many, but talk to few.";

cout<<"\nSentence: Mostly misunderstood and alone.";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nStrengths:";

cout<<"\n1.Seek and value harmony";

cout<<"\n2.Open minded and flexible";

cout<<"\n3.Very creative due to their intuitive nature.";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nWeaknesses:";

cout<<"\n1.Altruistic: They sometimes see themselves as too selfish only because";

cout<<"\nthey want to give much more than they are able to.";

cout<<"\n2.Difficult to get to know";

cout<<"\n3.Dislike dealing with data";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nINFPs you may know:";

cout<<"\n1.J.R.R Tolkein";

cout<<"\n2.William Shakespeare";

cout<<"\n3.Johnny Depp";

cout<<"\n4.Frodo Baggins(Lord of the rings)";

cout<<"\n5.Tom Hiddleson";

cout<<"\nPercentage of population:4.4%";

break;}

case 14:

{

cout<<"\nINFJ: Yin-yang enforcer";

cout<<"\nSuch types have an inborn sense of idealism and morality. ";

cout<<"\nThey are dreamers, but are capable of taking concrete steps to realise";

cout<<"\ntheir dreams.";

cout<<"\nThey share a very unique combination of traits: Soft spoken yet have ";

cout<<"\nstrong opinions which they will fight tirelessly for.";

cout<<"\nKarma is a very attractive idea to INFJs and they tend to act not for ";

cout<<"\npersonal gain, but to create balance.";

cout<<"\nSentence:The world is a place full of inequity!";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nStrengths:";

cout<<"\n1.They are inspiring and convincing.";

cout<<"\n2.Determined and passionate";

cout<<"\n3.They truly act just to make the world a better place.";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nWeaknesses:";

cout<<"\n1.Sensitive and highly vulnerable to criticism.";

cout<<"\n2.They need to remember to take care of themselves while making the";

cout<<"\nworld better.";

cout<<"\n3.Extremely private: Trusting a new friend can even be a challenge.";

cout<<"\nPress enter to continue.";

getch();

clrscr();

cout<<"\nINFJs you might know:";

cout<<"\n1.Martin Luther King";

cout<<"\n2.Nelson Mandela";

cout<<"\n3.Mother Teresa";

cout<<"\n4.Jon Snow(Game of Thrones)";

cout<<"\n5.Aragorn/Strider(Lord of the rings)";

cout<<"\nPercentage of population:1.5%, rarest.";

break;}

case 15:

{

while(iff8.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

case 16:

{

while(iff9.getline(t17,500,'$'))

{

cout<<"\n"<<t17;

cout<<"\nPress enter to continue.";

getch();

clrscr();

}

break;}

default:{cout<<"\nWrong choice, enter again!";

goto label26;}

} //switchv6 close

iff2.seekg(0,ios::cur);

iff3.seekg(0,ios::cur);

iff5.seekg(0,ios::cur);

iff8.seekg(0,ios::cur);

iff9.seekg(0,ios::cur);

iff10.seekg(0,ios::cur);

iff11.seekg(0,ios::cur);

iff12.seekg(0,ios::cur);

iff14.seekg(0,ios::cur);

iff16.seekg(0,ios::cur);

iff17.seekg(0,ios::cur);

cout<<"\n\nTo go back to the Different Personality types, enter 1.";

cout<<"\nTo go back to main menu, enter 2.";

cout<<"\nEnter choice:";

cin>>var;

switch(var)

{case 1: {

goto label26;

}

case 2: {clrscr();

goto label27;

}

}

} //label 26 close

} //case 4 close

case 5: {cout<<"\nCREDITS\n";

cout<<"\nTo Sri the ESFP(Entertainer) and Violet the INTJ(Architect) mainly, of course. \n9 3/4 out of 10, they deserve this credit.";

cout<<"\n(Obviously. We slaved for days to make this a reality.)";

cout<<"\n\nBut puting our big head aside, \nThanks to all those who took our quiz and helped us find out our mistakes.";

cout<<"\nSpecial thanks to Neill Mathew,Gwen,Joe Bobby, Razi, Suhaim Ahmed Shaiju(Sushi),\nNeeraj Krishnan, Patrick Wicks and Varun Iyer.";

cout<<"\nP.S You're all idiots.";

cout<<"\n\nViolet's note: To Moo and Sarah. Thanks for not helpng at all.";

cout<<"\n\nTo go back to main menu, press Enter.";

getch();

clrscr();

goto label27;

}

default:{cout<<"\nWrong choice, you dunderhead. Snape disapproves. Press Enter to continue and get the menu.";

getch();

goto label27;

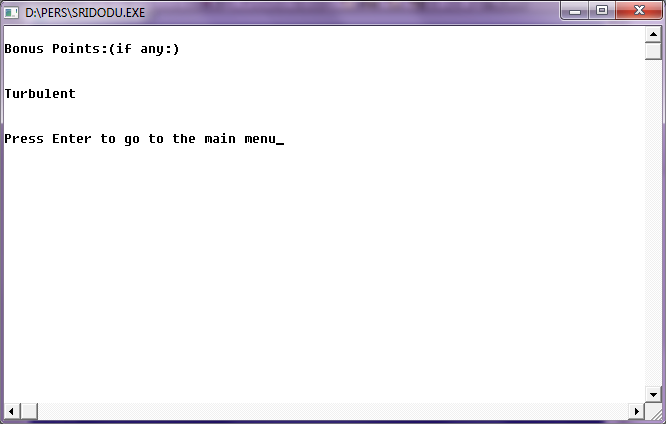
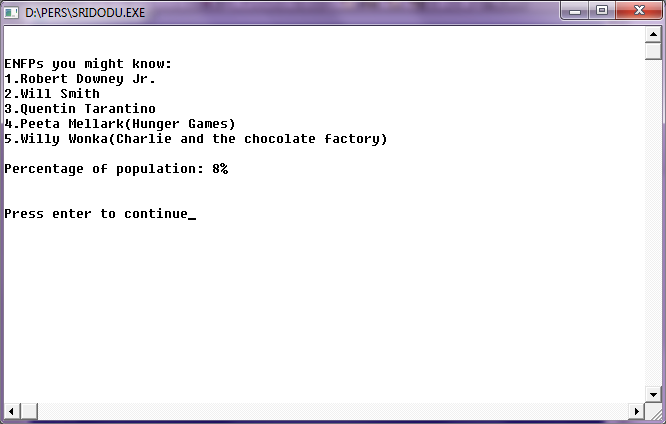
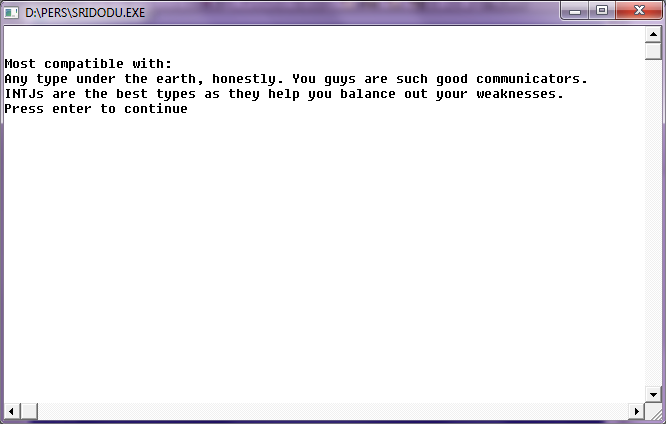
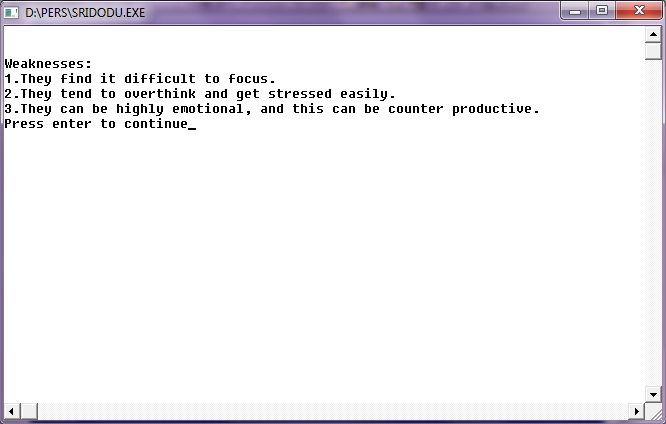
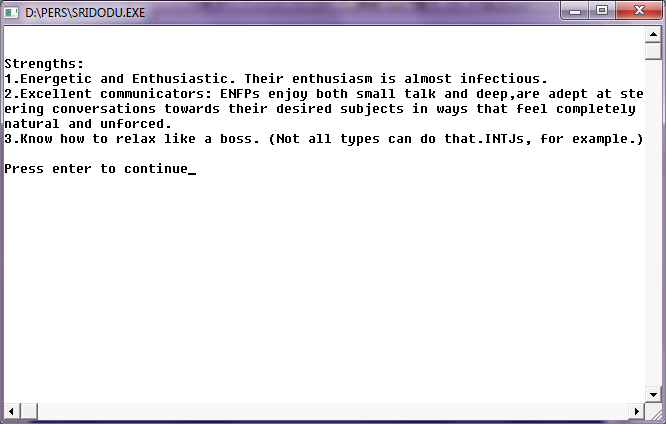
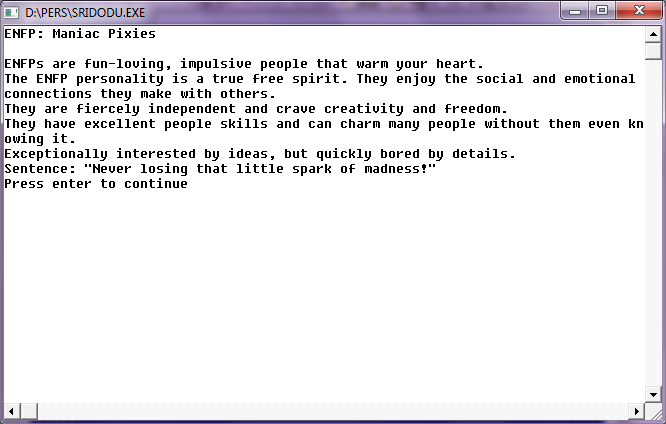
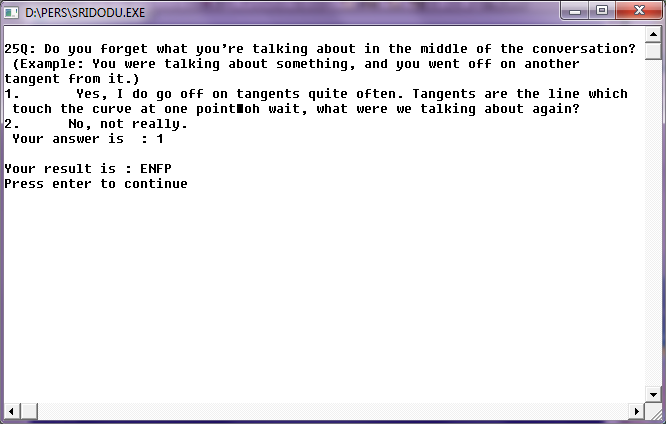
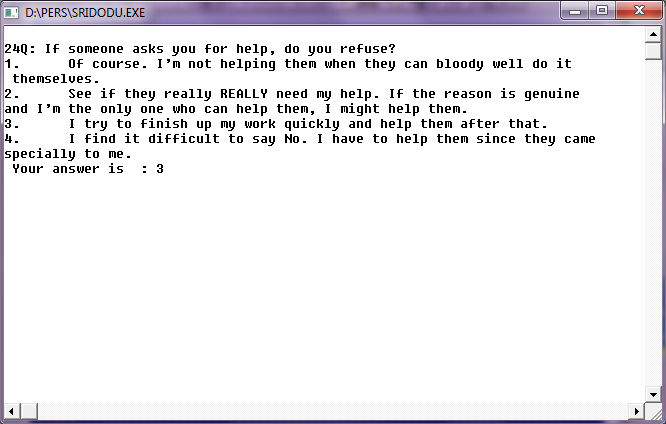
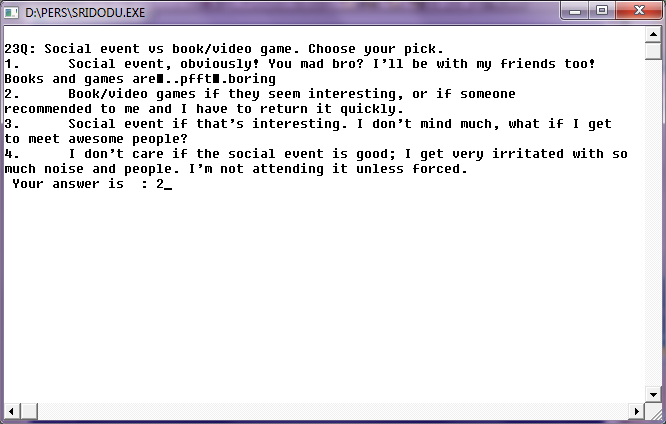
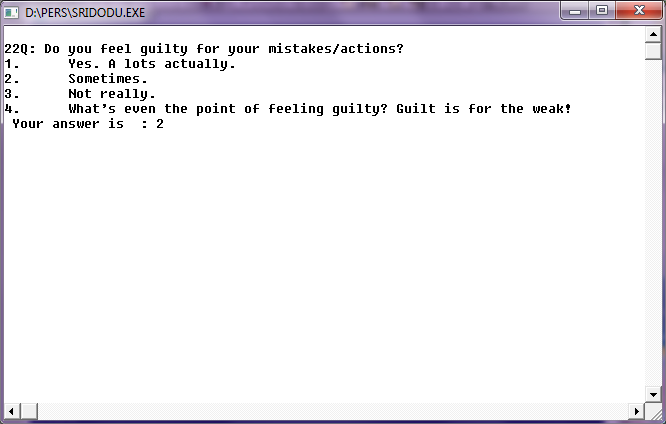
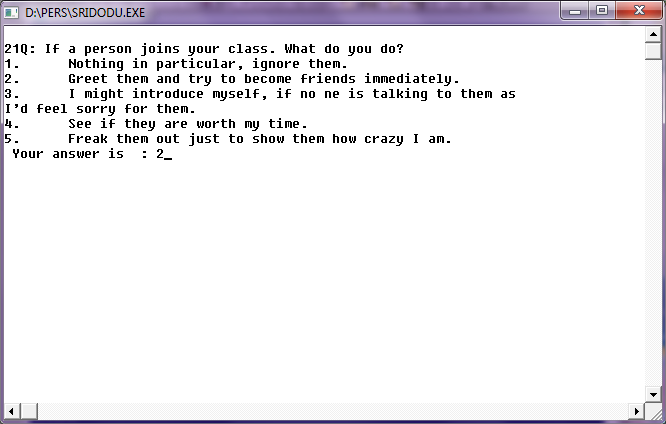
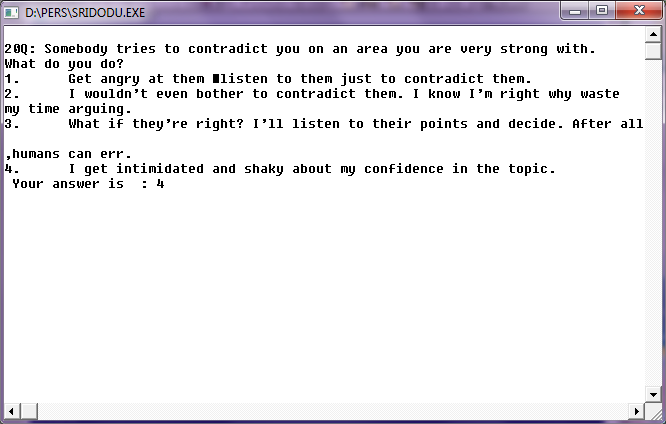
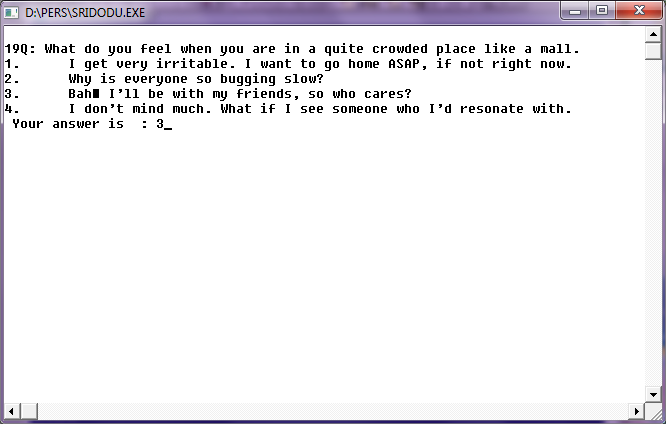
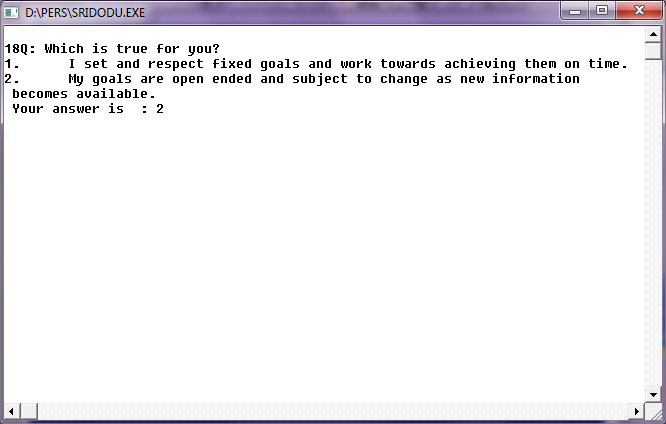
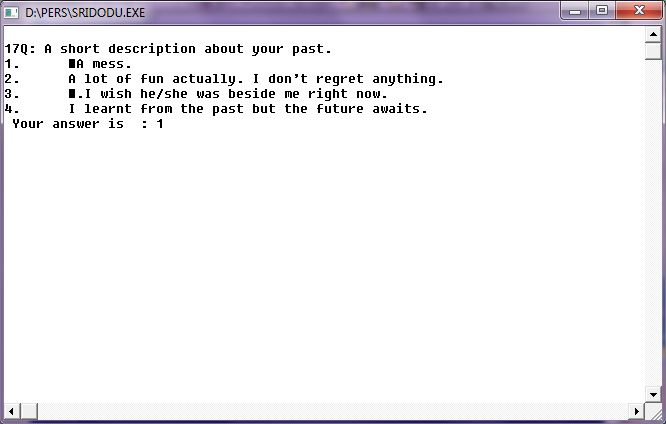
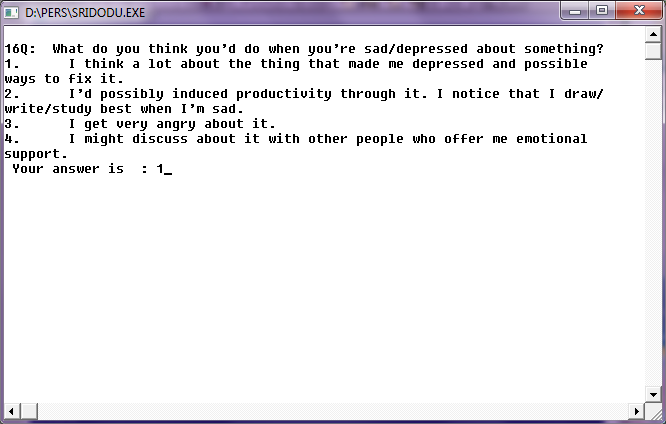
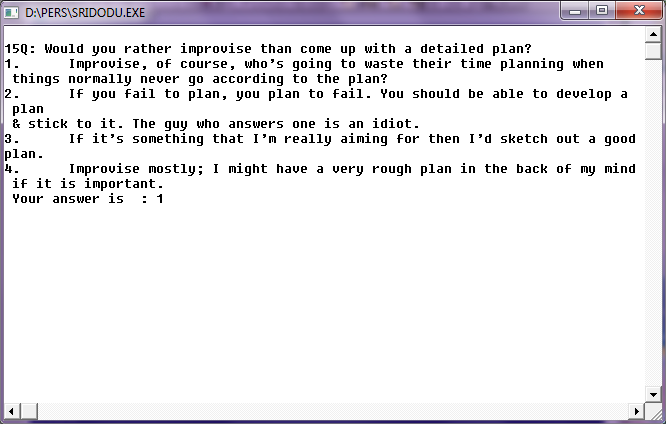
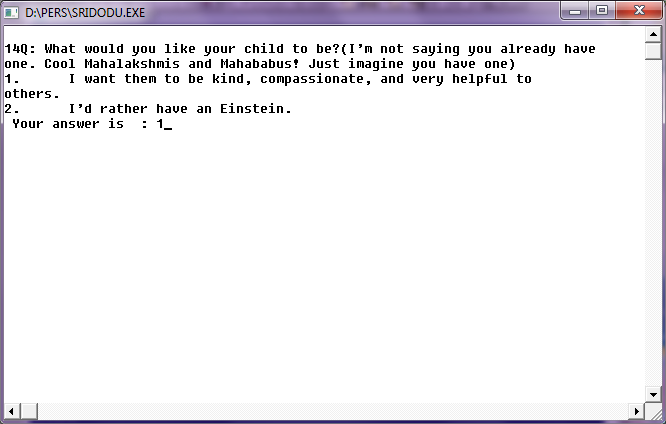
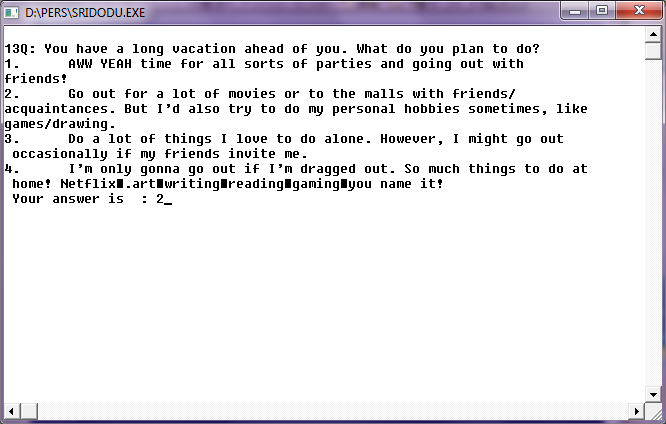
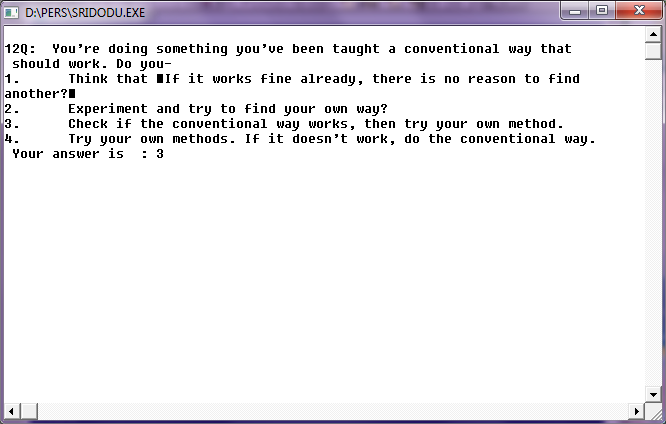
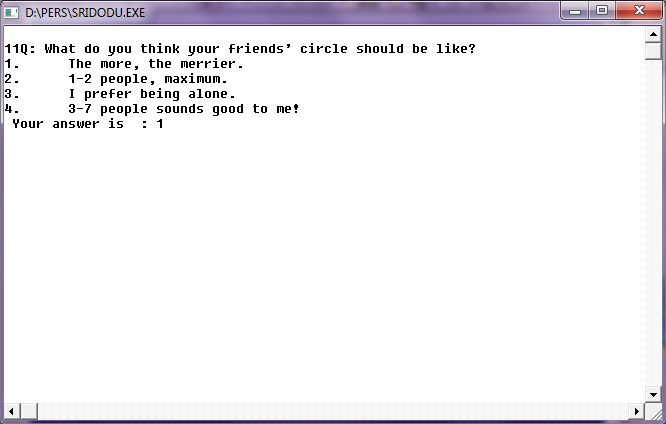
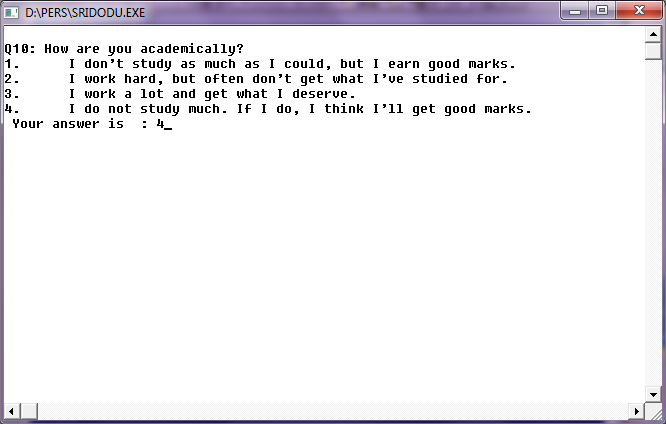
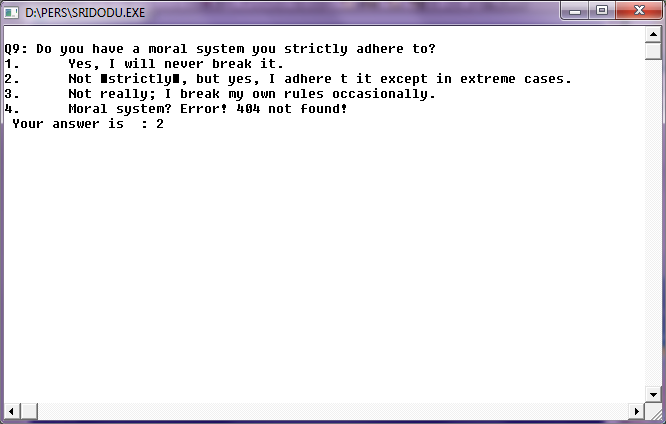
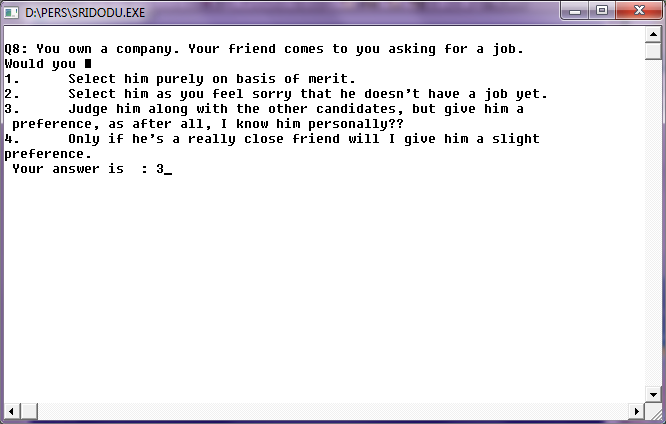
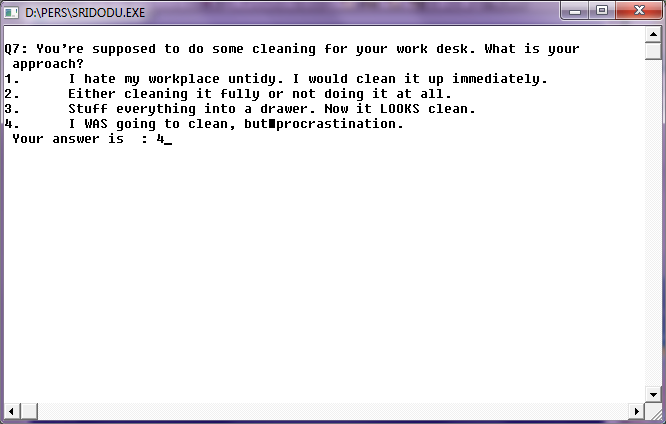
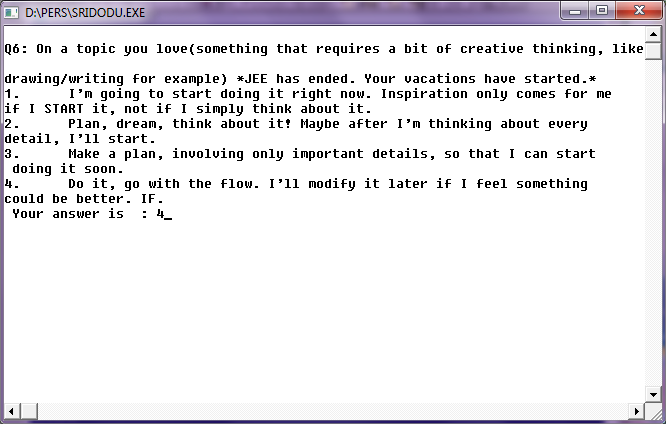
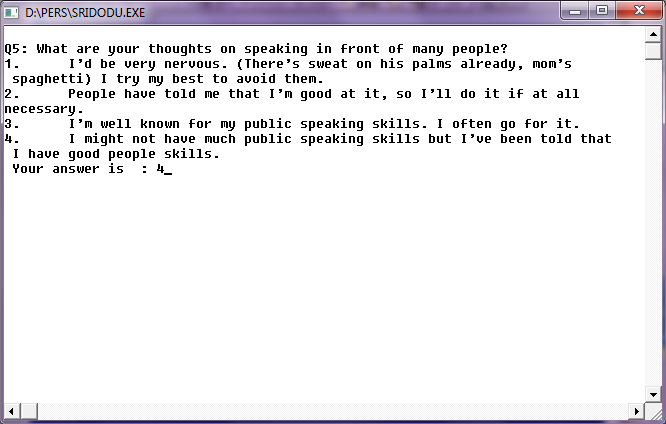
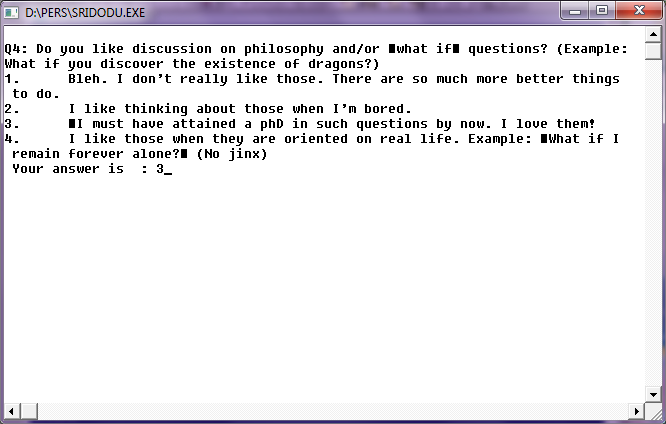
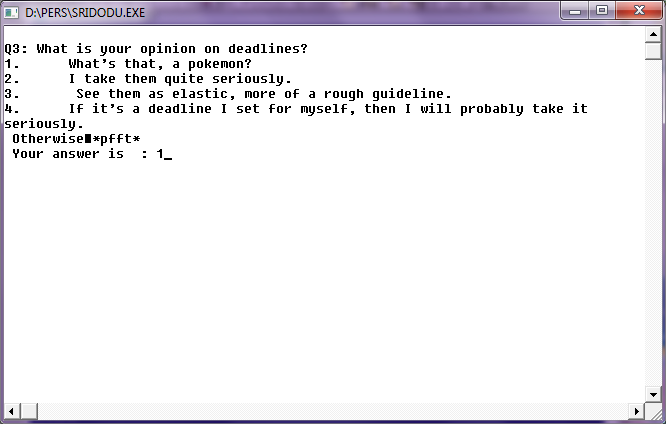
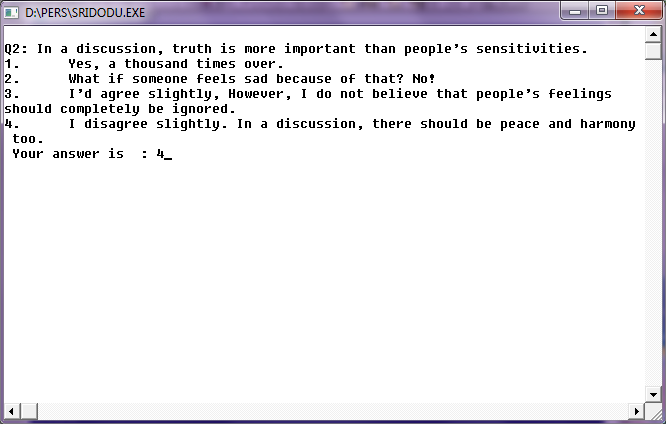
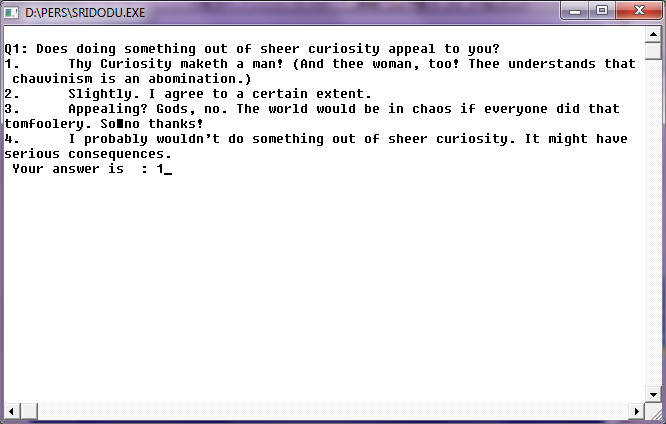
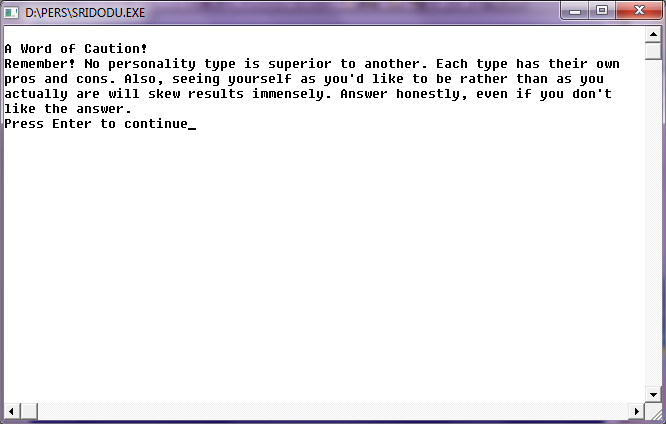
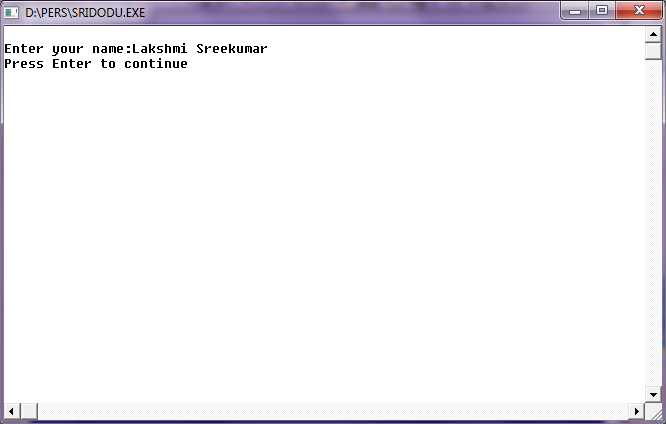
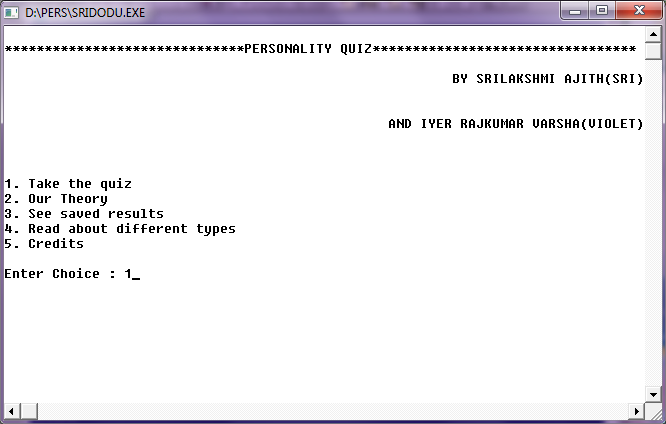
}

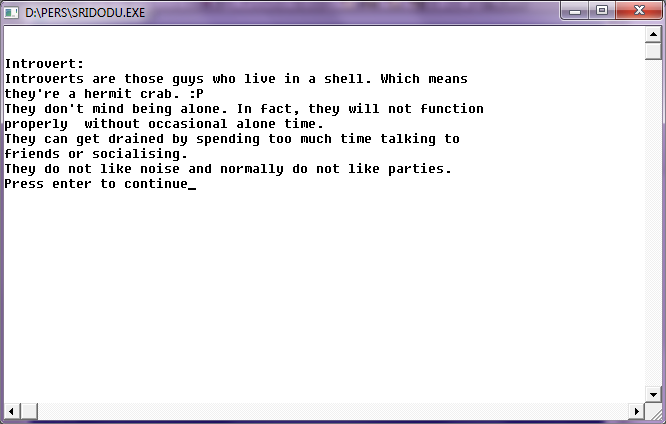
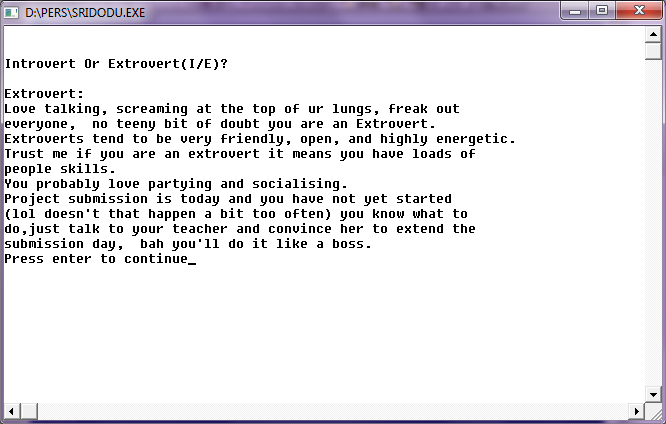
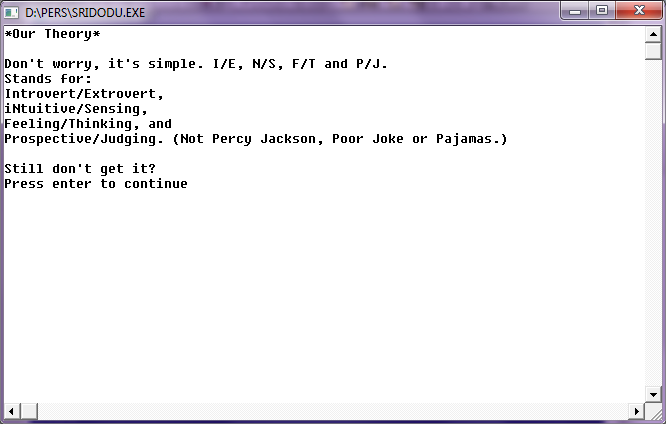
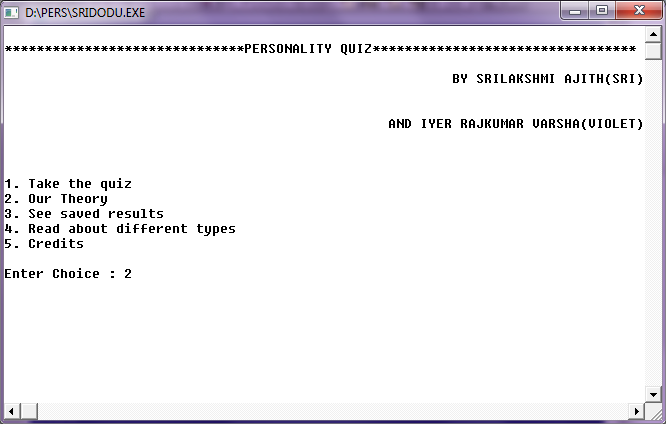
} //switchv1 close

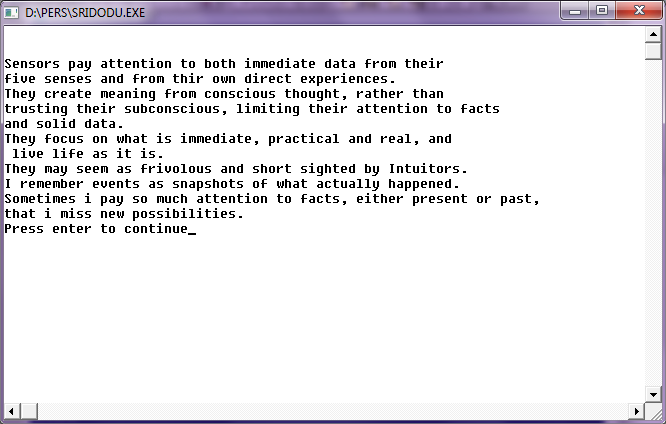
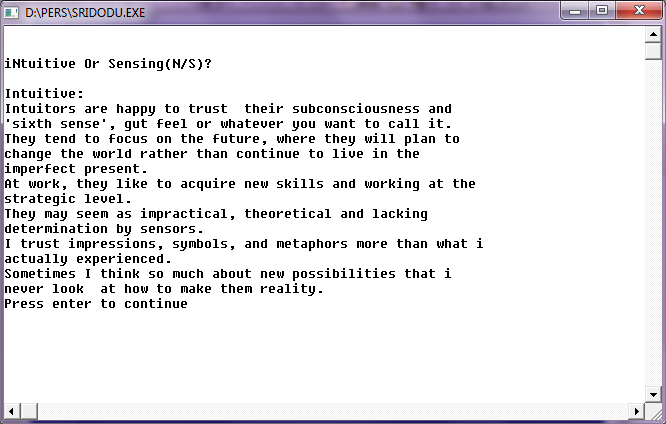
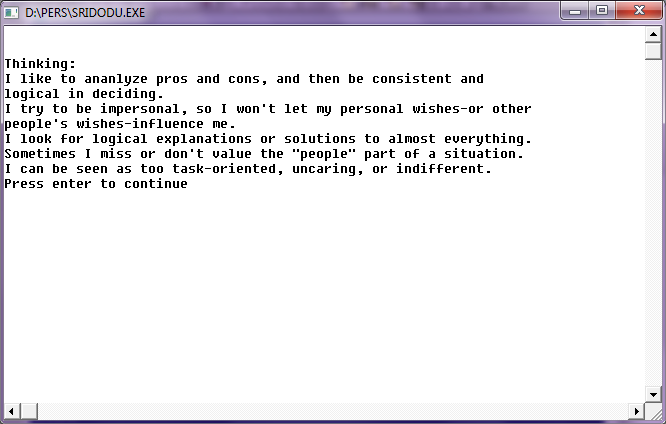
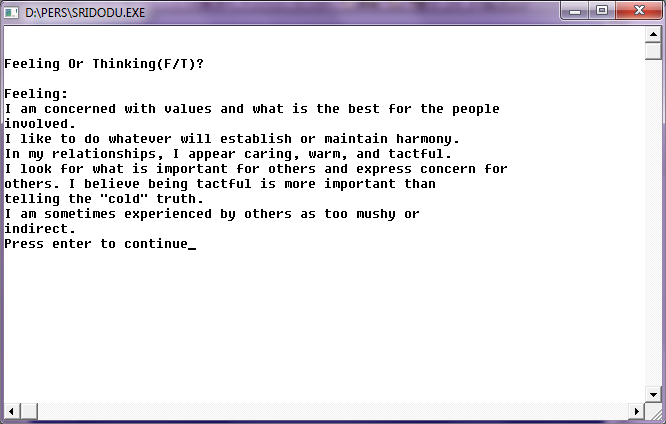
} //label28 close

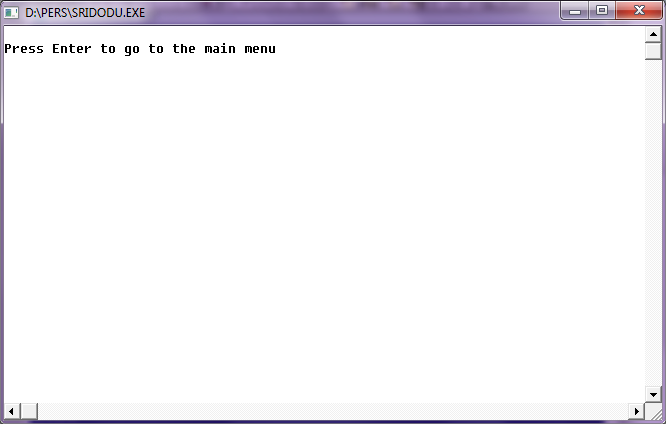
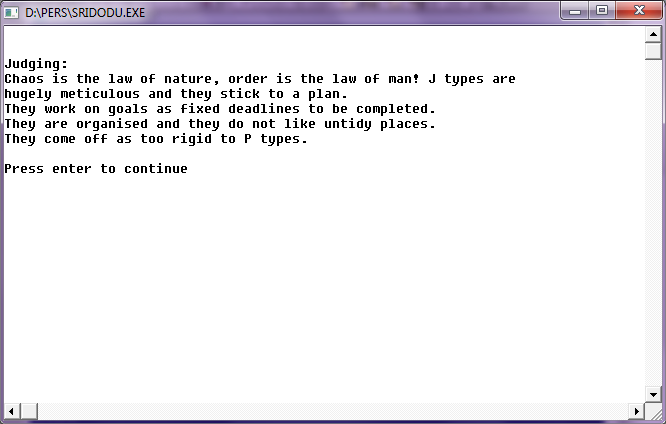
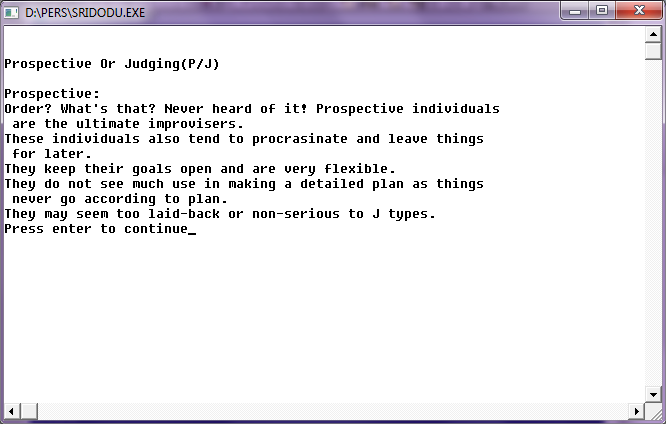
} //void main close

**PROGRAM SAMPLE OUTPUT**

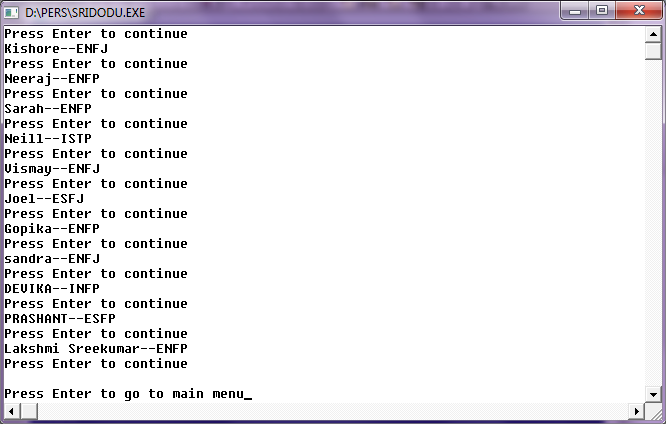
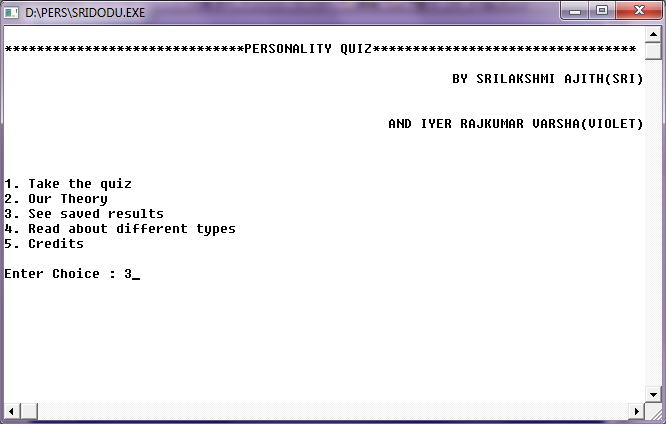


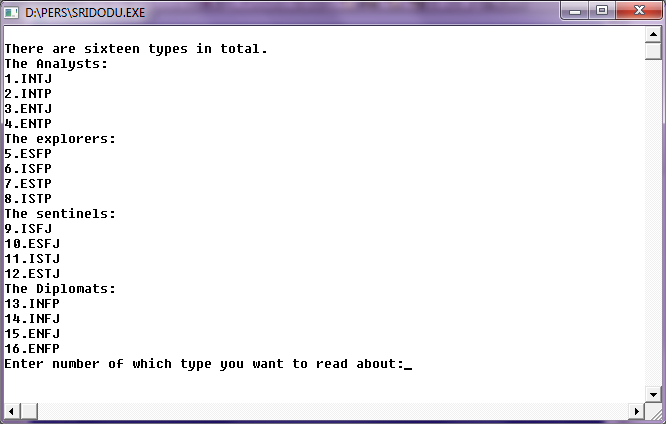
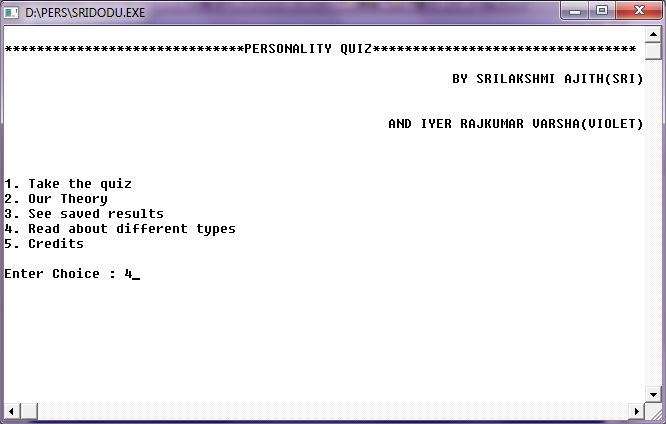


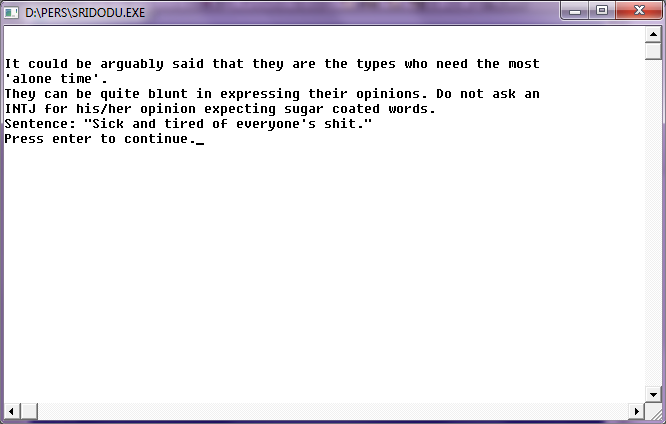
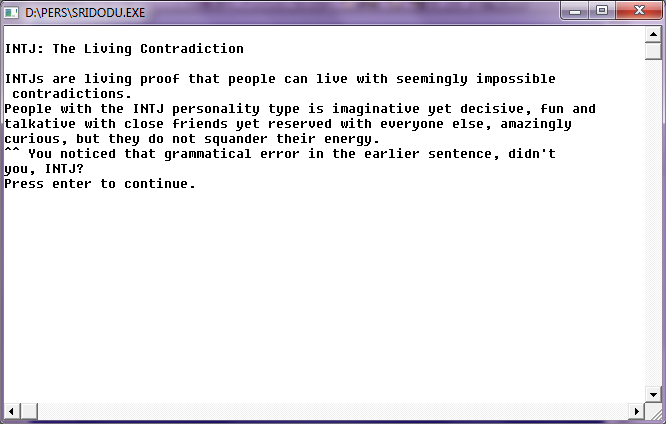


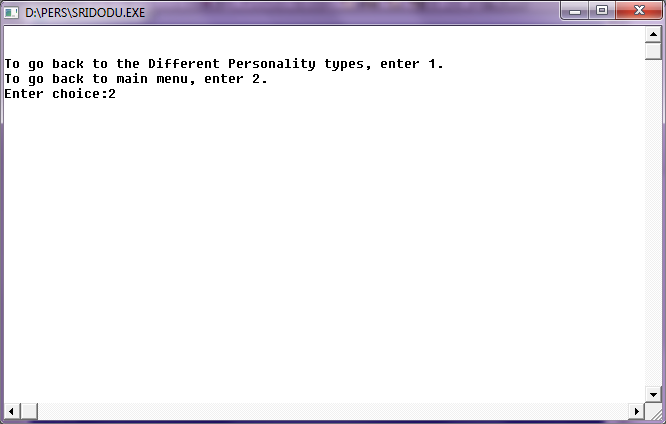
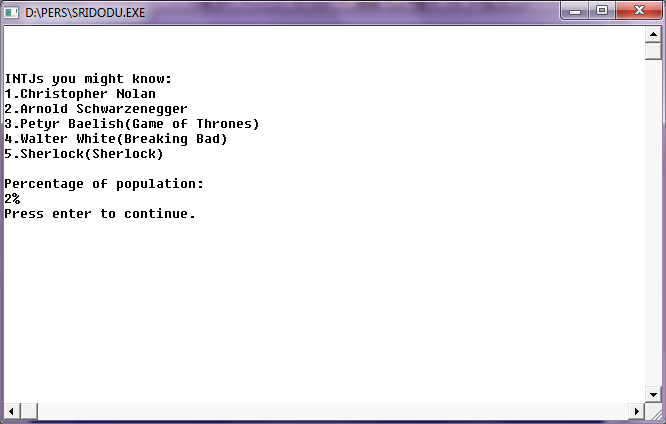
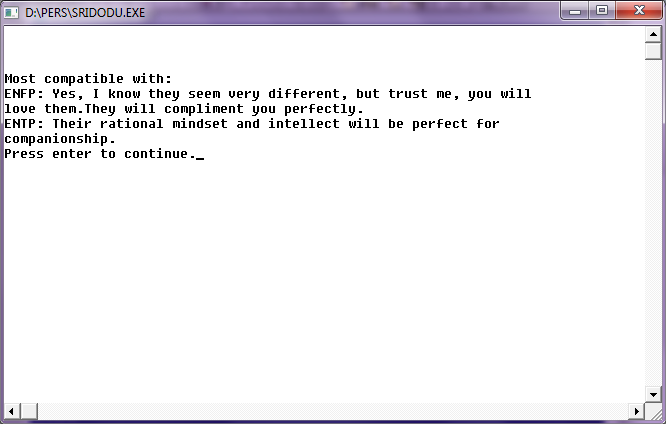
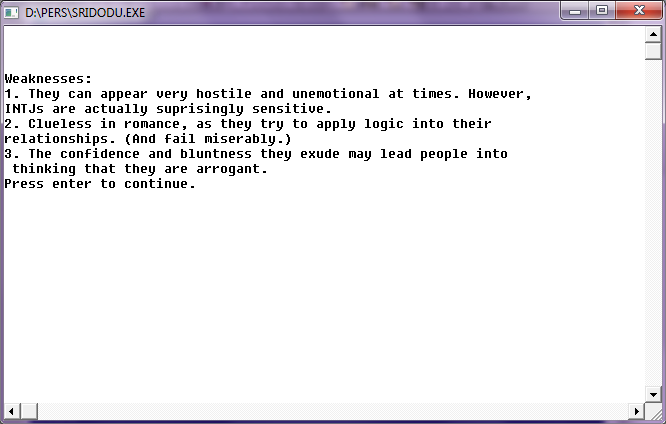
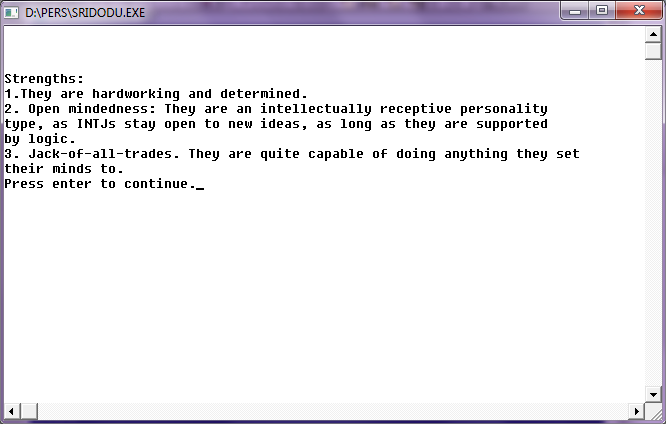


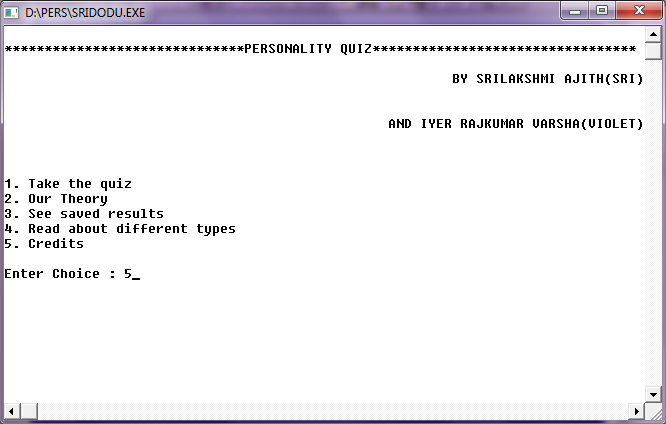
###########################################################################

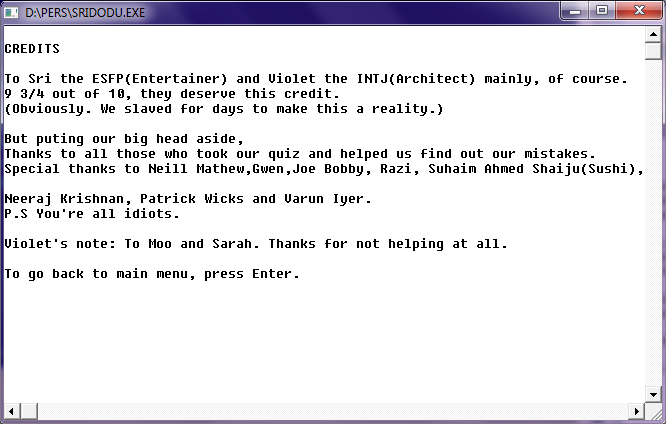
##############################################################################











**BIBLIOGRAPHY**

* “C++ programming” by Sumita Aurora
* www.16personalities.com
* “The Art of thinking clearly” by Rolf Dobelli
* [www.google.com](http://www.google.com)