

**RCC INSTITUTE OF INFORMATION TECHNOLOGY**

**CS- PROJECT**

**“DIABETES DISGNOSIS SOFTWARE”**

**SUBMITTED TO: MRS. MONIKA SINGH**

**SUBMITTED BY:**

1. **ANIRUDDHA SADHUKAN CSE2015/004**
2. **ANISHA ROY CSE2015/019**
3. **ARPITA BASAK CSE2015/023**
4. **ZENITH ROY CSE2015/025**

**PAPER: Basic Computation & Principle of Computer Programming [LAB]**

**PAPER CODE: CS291**

**ENGINEERING SESSION: 2015-19**

**INDEX**

|  |  |  |
| --- | --- | --- |
| SLNO | TOPIC |  |
| 1.  2.  3.  4.  5. | INTRODUCTION   * TOOLS USED   **DIABETES DIAGNOSIS SOFTWARE**   * OBJECTIVES * EXISTING SYSTEM   PROGRAM  OUTPUT  FILE SAVED |  |

INTRODUCTION

Diabetes is a disorder characterized by hyperglycemia or elevated blood glucose. If the amount of sugar in blood runs too high or too low, then we typically feel bad. Diabetes is the name of the condition where the blood sugar level consistently run too high. Diabetes is the most common (blood sugar). Our bodies function best at a certain level of sugar in the blood stream endocrine disorder. Diabetes has potential long term complications that can affect the kidneys, eyes, heart, blood vessels and nerves. There are two types of diabetes which are similar in their elevated blood sugar, but different in many other ways. Diabetes is correctly divided into two major subgroups. The division is based upon whether the blood sugar problem is caused by insulin deficiency or insulin resistance.

Hence here is our project on c language based programing on this Diabetes detection. The program can provide you with the following details:

1. This program can tell if you are diabetic or not and if you are diabetic what kind of diabetes you are suffering from.
2. It can store the details of the patients along with the report.
3. It shows all the details stored.
4. One have the option to search the reports based on names or numbers.

TOOLS USED:

* **HEADER FILES:**

**1) stdio.h // for standard i/o operations**

**2)stdlib.h // for standard library functions**

**3)string.h // for string handling**

* **STRUCTURE: - 1) patients**
* **FUNCTIONS: -**

**1) details()**

**2) diagnostic1 ()**

**3) diagnostic2 ()**

**4) diagnostic3 ()**

* **LOOP:- 1)for**
* **STATEMENTS:- 1)if-else statement 2)switch statement**

DIABETES DIAGNOSIS SOFTWARE: OBJECTIVES

**Objective:**

If you are facing with any kind of disease or infection and knowing about these disease on time will help you to get cure and take precautions on time. This process will also help to get over from these suffering in few days or in months. Among several diseases and sufferings, many peoples are suffering from diabetes. With this computer based diabetes detection software, user will able to do their self-checkup without taking help of a doctor. This system will keep records of particular patient or of any person and by using their general data collected during analysis process it will able to tell you, whether you are suffering from diabetes or not. This system having the ability to detect diabetes and categories as per the data which has been collected during questions answer section. Its user friendly graphical mode will make its user to use this system in interactive manner. You will have to just answer few questions and these questions has been divided into sections. Upon completion of these questions you will able to get your final result and your health status regarding diabetes. To keep records, you will required to ask your basic information and using these information, your final report will be prepared by keeping view of answer which has been provided by you. Number of options will be given in answer section and you have to select whatever suits you the best.

Thereafter it can also store details and records of number of patients

And hence search them by name or number from all the details recorded.

**Existing System:**

As per the existing system, users have to invest money before knowing their diagnosis report. They have to visit diagnostic center, consult their doctor and wait for a day to get their result. If someone have only doubt and want to have a checkup, then it will have their wastage of money and time for them. Each time you have to provide your basic information and go through the same diagnosis process to get your diagnosis result which can be dangerous in some serious condition. Patients should go through the checkup process immediately before the diagnosis process, if such condition arises. So keeping customer records and searching their records manually sometimes not possible and sometime it’s a time taking process.

PROGRAM

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

char d1[10];

struct patients

{

char name[20];

int age,no;

char sex;

float ht,wt;

char report[30];

}pts[100];

void cls() //clear screen(clear for linux,cls for windows)

{

system("cls||clear");

}

void details(int i) //take input of personal details of patient

{

printf("\nPatient number %d\n\n\n",i+1);

pts[i].no=i+1;

printf("P E R S O N A L I N F O R M A T I O N\n\n\n");

printf("N A M E :");

scanf(" ");

gets(pts[i].name);

printf("A G E :");

scanf("%d",&pts[i].age);

printf("W E I G H T :");

scanf("%f",&pts[i].wt);

printf("H E I G H T :");

scanf("%f",&pts[i].ht);

printf("S E X (M/F) :");

scanf(" %c",&pts[i].sex);

pts[i].sex=toupper(pts[i].sex);

cls();

}

void welcome() //print welcome screen

{

char a;

printf("\t\*\*\*\*\*\*\*\*\* W E L C O M E \*\*\*\*\*\*\*\*\*\n\n\n");

printf("M E D I C A L D I A G N O S I S S O F T W A R E\n");

printf ("---------------------------------------------------\n\n");

printf("Created by Aniruddha Sadhukhan CSE/2015/004\n");

printf(" Anisha Roy CSE/2015/019\n");

printf(" Arpita Basak CSE/2015/023\n");

printf(" Zenith Roy CSE/2015/025\n");

printf("\n\n \*\*\*\*\*\*\*\* PRESS ENTER TO CONTINUE \*\*\*\*\*\*\*\*\*");

scanf("%c",&a);

return;

}

int diagnostics1() //Level 1 diagnostic

{

int i=0,result=0,j,count=0;

printf("Please enter the form page\n");

//Getting level 1 symptoms :----

printf("APPETITE (H(HIGH),/L(LOW),/N(NORMAL) :");

scanf(" %c",&d1[i]); //space before %c is necessary

d1[i]=toupper(d1[i]);

++i;

printf("FREQUENCY OF THIRST(H(HIGH),/L(LOW)/N(NORMAL) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

++i;

printf("FREQUENCY OF URINATION(H(HIGH),/L(LOW),/N(NORMAL) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

++i;

printf("VISION (I(IMPAIRMENT),/N(NORMAL) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

printf("URINE SUGAR(P(PASSIVE)/A(ACTIVE) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

printf("KETONUREA(P(PASSIVE)/A(ACTIVE) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

printf("FASTING BLOOD SUGAR(H(HIGH)/L(LOW)/N(NOMAL) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

printf("R B S (H(HIGH)/L(LOW)/N(NORMAL) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

printf("FAMILY HISTORY OF DIABETES(P(PASSIVE)/A(ACTIVE) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

printf("OGTT(D/N) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

//diagnosis of level 1 symptoms:----

if(d1[9]=='D' )

result=1;

else if(d1[5]=='P'&& d1[6]=='H' && d1[7]=='H')

result=1;

else

{

for(j=0;j<10;j++)

{

if(d1[j]=='H'||d1[j]=='P'||d1[j]=='D'||d1[j]=='I')

count++;

}

if(count>5)

result=1;

}

cls();

if (result==1)

printf("\n\n\n\tYou are DIABETIC\n\n");

else printf("\n\n\n\tYou are NOT DIABETIC\n\n");

return(result);

}

int diagnostics2() //Level 2 diagnostic

{

int i=0,result=0,j,count=0;

printf("Test for primary and secondary dibetics\n\n");

//Getting level 2 symptoms :----

printf("PANCREATITIS(P/A) :");

scanf(" %c",&d1[i]);//space before %c is necessary

d1[i]=toupper(d1[i]);

++i;

printf("CARCINOMA(P/A) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

++i;

printf("CIRHHOSIS(P/A) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

++i;

printf(" HCTS (H/L/N) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

printf("HEPATITIS(P/A) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

printf("HORMONAL DISORDER(P/A) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

printf("PANCREATECTOMY(P/A) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

//diagnosis of level 2 symptoms:----

for(j=0;j<7;j++)

{

if(d1[j]=='H'||d1[j]=='P')

count++;

}

if(count==7)

result=1;

cls();

if (result==1)

printf("\n\n\nIT IS SECONDARY DIABETES\n\n");

else printf("\n\n\n\tIT IS PRIMARY DIABETES\n\n");

return(result);

}

int diagnostics3() //Level 3 diagnostic

{

int i=0,result=0,j,count=0;

printf("Test for NON INSULIN DEPENDENT or INSULIN DEPENDENT DIABETES\n");

//Getting level 3 symptoms :----

printf("AGE(young(Y)/Middle aged(M)/Elderly(E)) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

++i;

printf("BODY WEIGHT(normal(N)/Above normal(A)/Below normal(B)) :");//space before %c is necessary

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

++i;

printf("DURATION (weeks(W)/Months(M)/Years(Y)) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

++i;

printf("KETONUREA(P/A) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

printf("AUTO ANTIBODIES(P/A) :");

scanf(" %c",&d1[i]);

d1[i]=toupper(d1[i]);

i++;

//diagnosis of level 3 symptoms:----

if((d1[0]=='Y')&&(d1[1]=='N')&&(d1[2]=='W')&&(d1[3]=='P')&&(d1[4]=='P')||

(d1[0]=='Y')&&(d1[1]=='B')&&(d1[2]=='W')&&(d1[3]=='P')&&(d1[4]=='P')||

(d1[0]=='Y')&&(d1[1]=='N')&&(d1[2]=='M')&&(d1[3]=='P')&&(d1[4]=='P')||

(d1[0]=='Y')&&(d1[1]=='N')&&(d1[2]=='Y')&&(d1[3]=='P')&&(d1[4]=='P'))

result=0;

else result=1;

cls();

if (result==1)

printf("\n\n\nIT IS NON INSULIN DEPENDENT DIABETES\n\n");

else printf("\n\n\n\tIT IS INSULIN DEPENDENT DIABETES\n\n");

return(result);

}

void display(int i) //display details of patients

{

printf("\n\n\nPatient number %d \nN A M E :",pts[i].no puts(pts[i].name);

printf("A G E :%d\n",pts[i].age);

printf("W E I G H T :%.2f\n",pts[i].wt);

printf("H E I G H T :%.1f\n",pts[i].ht);

printf("S E X :%c\nReport :",pts[i].sex);

puts(pts[i].report);

}

int main()

{

int m,n,o,p,j,z,num,i;

char a,b,namee[20],ch;

welcome();

cls();

printf("Enter no. of patients[1-100] : ");

scanf("%d",&p);

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

{

details(j);

m=diagnostics1();

printf("\*\*\*\*\*\*\*\* PRESS ENTER TO CONTINUE \*\*\*\*\*\*\*\*\*");

scanf("%c%c",&a,&b);//first %c takes input of ENTER pressed earlier

cls();

if (m==0)

strcpy(pts[j].report,"Not Diabetic");

else

{

n=diagnostics2();

printf("\*\*\*\*\*\*\*\* PRESS ENTER TO CONTINUE \*\*\*\*\*\*\*\*\*");

scanf("%c%c",&a,&b);//first %c takes input of ENTER pressed earlier

cls();

if (n==1)

strcpy(pts[j].report,"Secondary Diabetic");

else

{

o=diagnostics3();

if (o==0)

strcpy(pts[j].report,"INSULIN DEPENDENT DIABETES");

else

strcpy(pts[j].report,"NON INSULIN DEPENDENT DIABETES");

printf("\*\*\*\*\*\*\*\* PRESS ENTER TO CONTINUE \*\*\*\*\*\*\*\*\*");

scanf("%c%c",&a,&b);//first %c takes input of ENTER pressed earlier

cls();

}

}

}

for(j=0;j<p;j++) //display all records

{

display(j);

}

printf ("\n\nDo you to save the entry details to file?(y/n) ");

scanf("%c",&ch);

if(ch=='y'||ch=='Y')

{

FILE \*fp;

fp=fopen("pat.txt","w");

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

{

fprintf(fp,"\n\n\nPatient number%d\n",pts[i].no);

fprintf(fp,"N A M E :");

fputs(pts[i].name,fp);

fprintf(fp,"\nA G E :%d",pts[i].age);

fprintf(fp,"\nW E I G H T :%.2f",pts[i].wt);

fprintf(fp,"\nH E I G H T :%.1f",pts[i].ht);

fprintf(fp,"\nS E X :%c\nReport:",pts[i].sex);

fputs(pts[i].report,fp);

}

fclose(fp);

}

lebel:printf ("\n\nEnter 0 to exit,1 for search by name,2 for search by number: ");

scanf("%d",&z);

switch(z)

{

case 0: exit(0); //exit

case 1: printf ("Enter name you want to search: "); //search by name

scanf(" ");

gets(namee);

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

{

if(strcmp(namee,pts[j].name)==0)

{

display(j);

break;

}

}

if (j==p)

printf ("\nnot found\n");

break;

case 2: printf ("Enter number you want to search: "); //search by number

scanf("%d",&num);

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

{

if(num==pts[j].no)

{

display(j);

break;

}

}

if (j==p)

printf ("not found");

break;

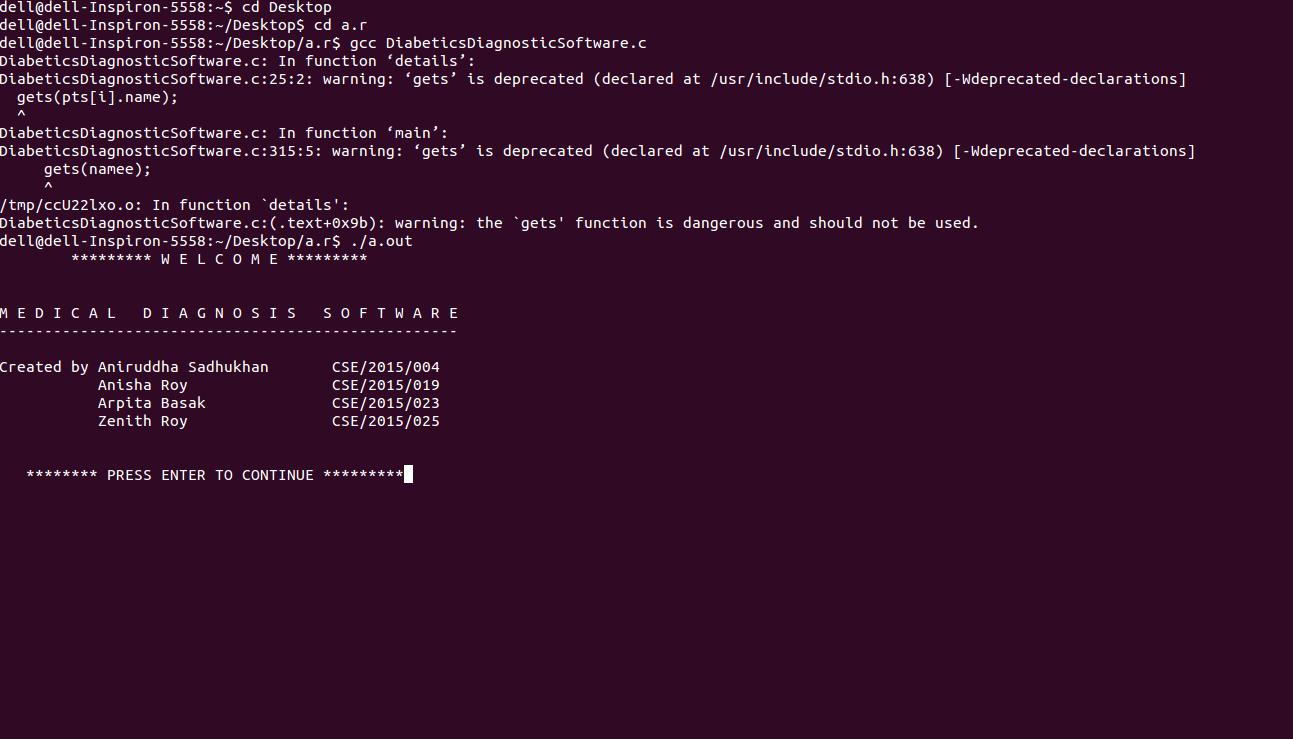
default:printf ("Wrong option");

}

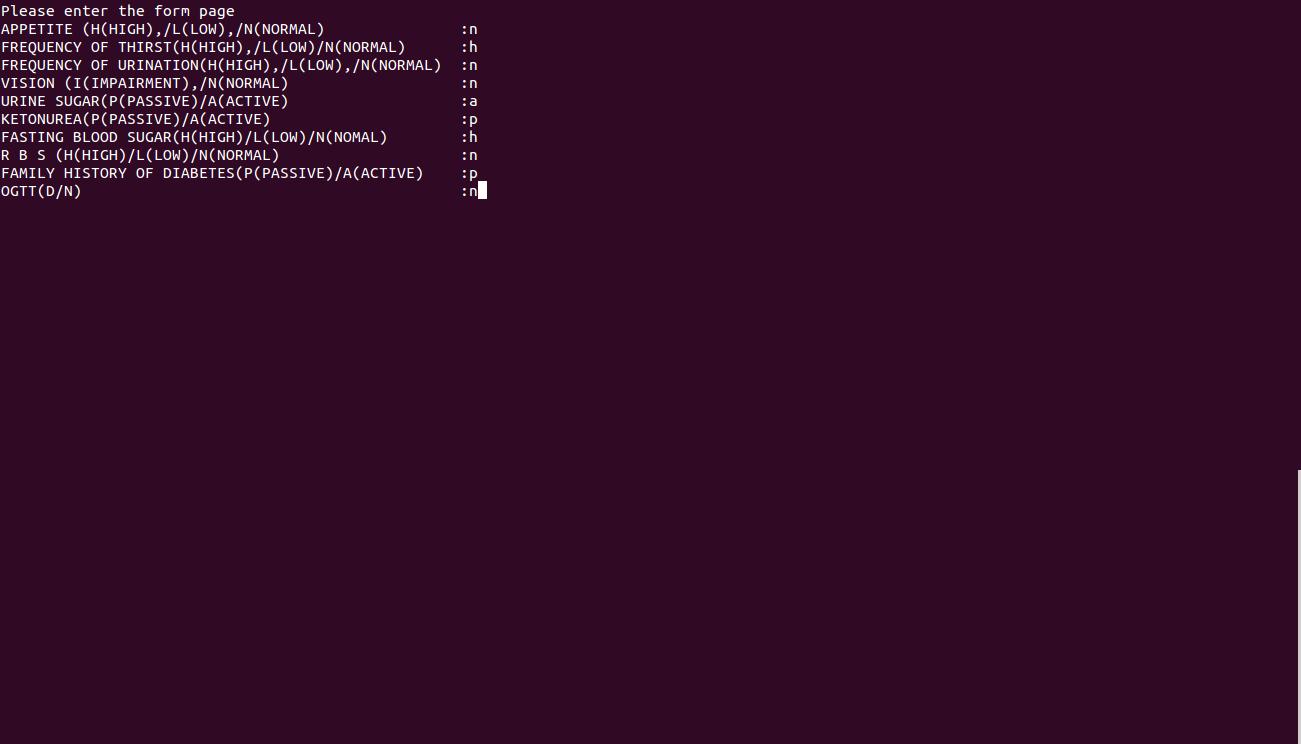
goto lebel;

}

OUTPUT

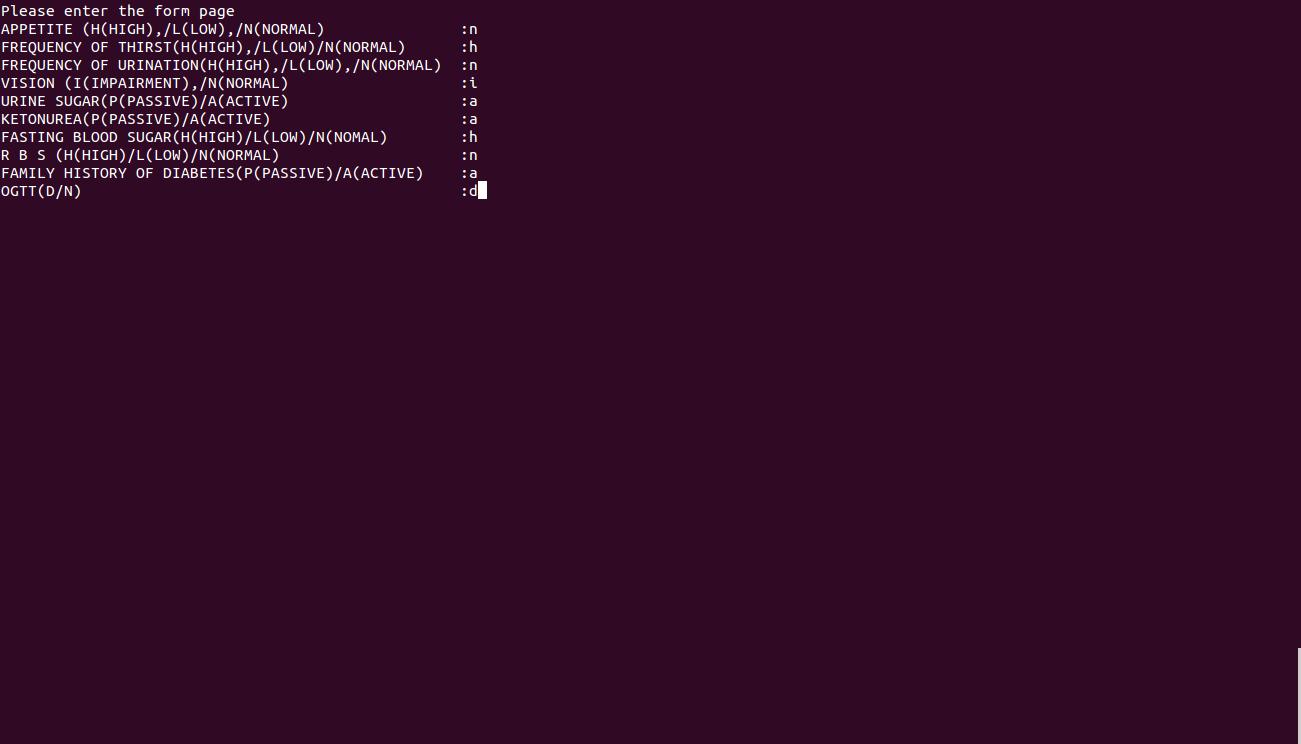








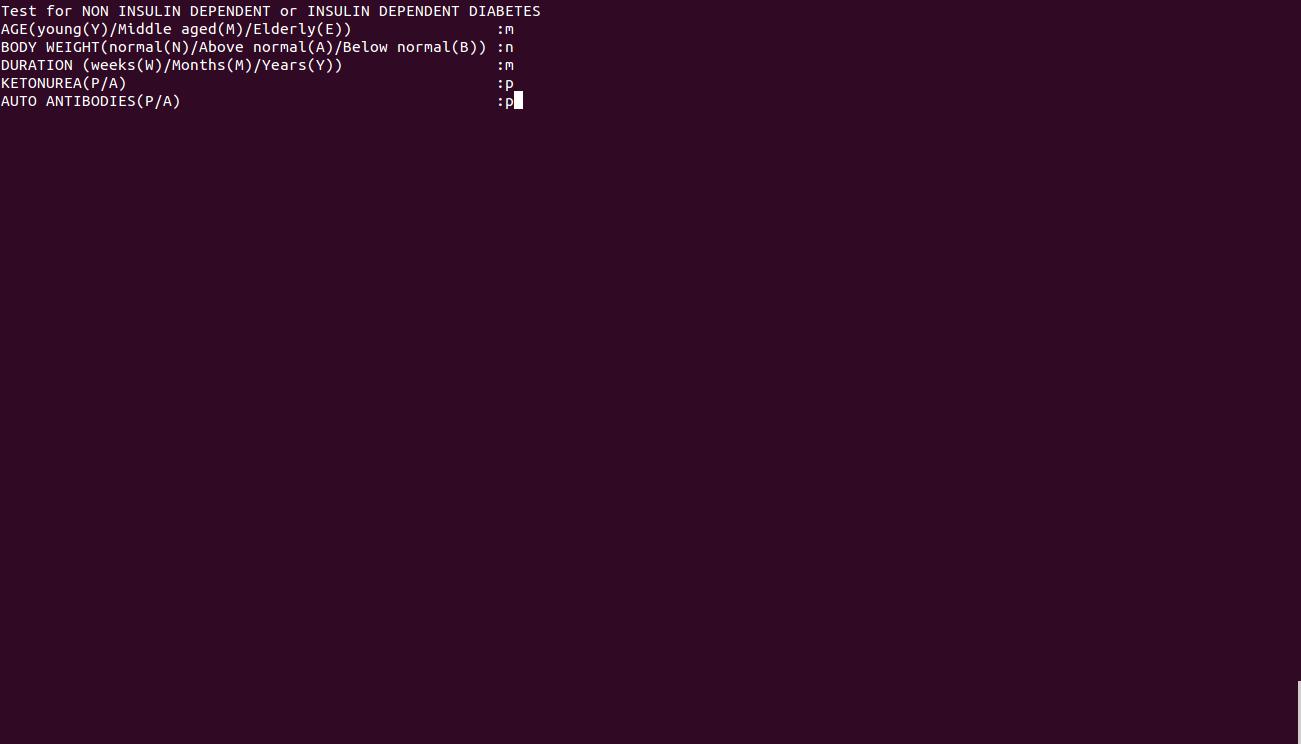






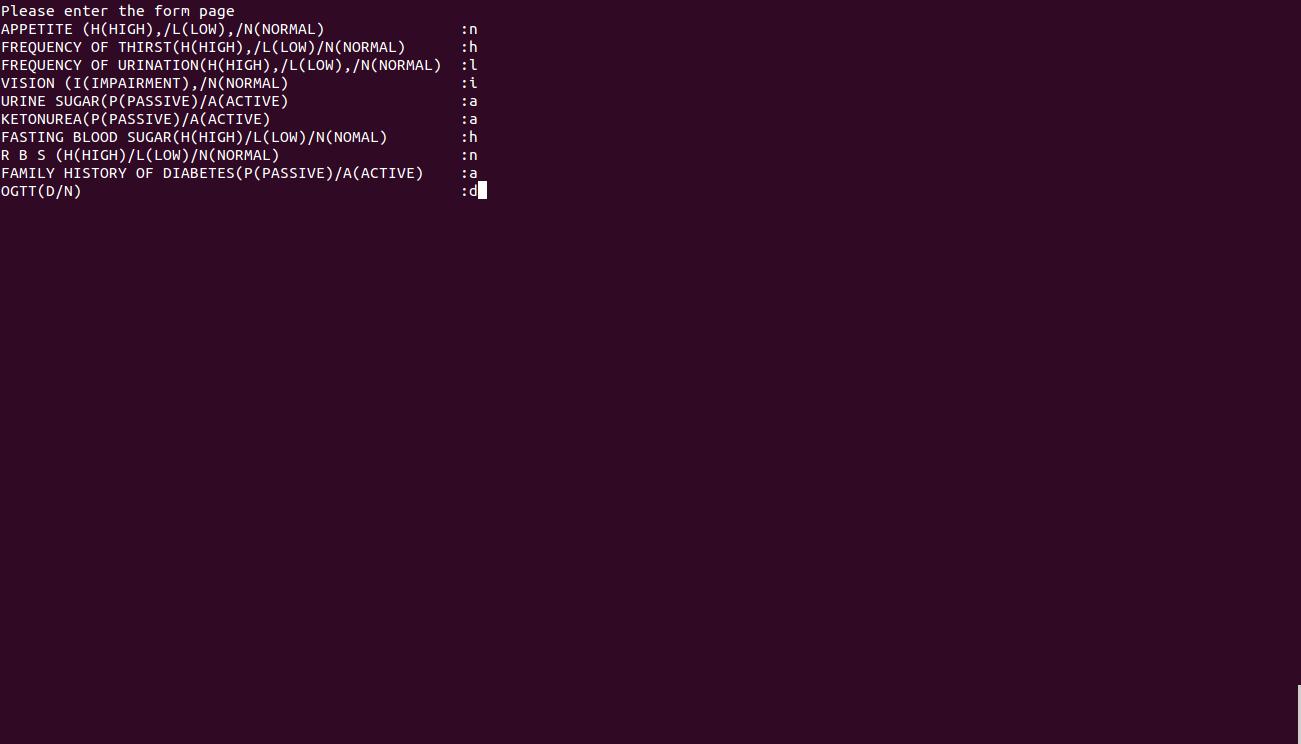








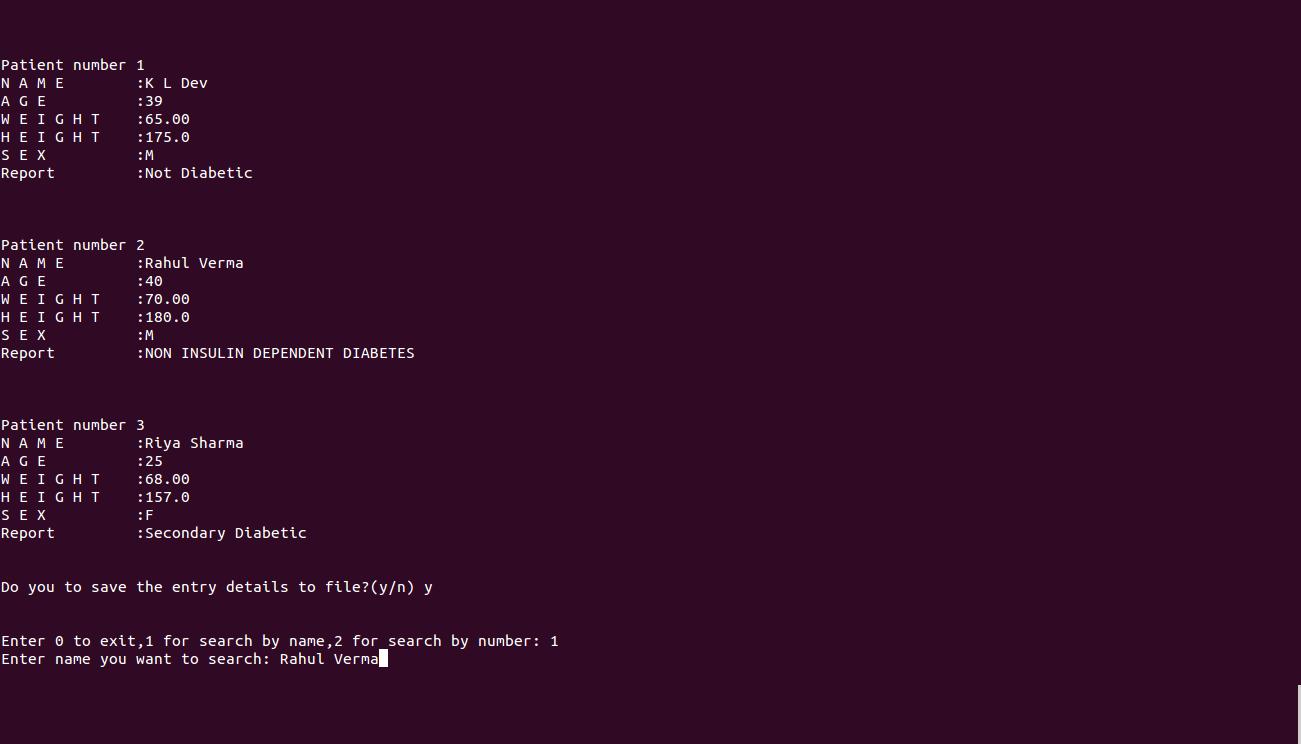


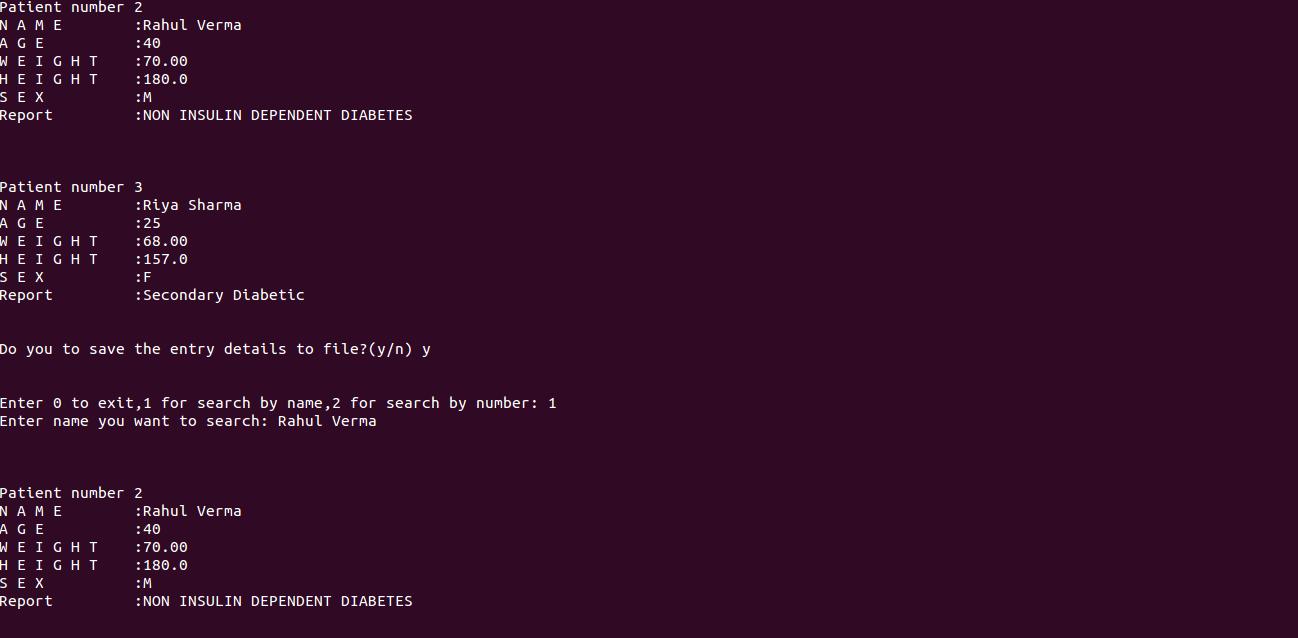


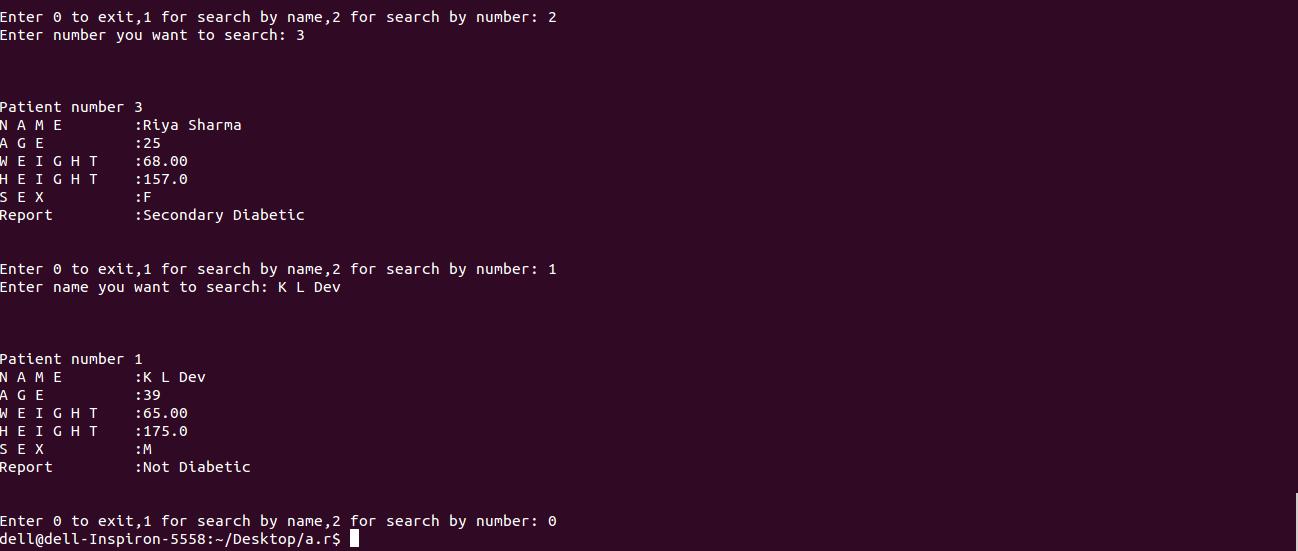




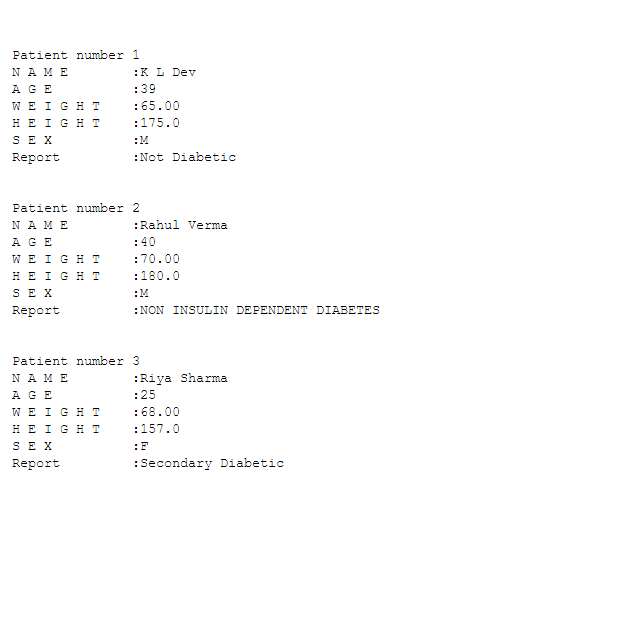








FILE SAVED



**\_\_\_\_THE END\_\_\_\_**