Computation of Simple Lens Design Parameters Using C++ A Lens Design Tool for System Engineer

Kaustubh Borole [1] & Aarti Borole [2]
[1] Alard College of Engg & Management, Marunje, Pune 411057
kborole7@gmail.com
[2] Precision Optics & Scientific Instruments, Pune 411016, optroindia@gmail.com
aartiborole@gmail.com

ABSTRACT: Optics form part of the Instrumentation Design of almost all branches of Science and Technology. System Engineers (non-optics personal) manage with little or no theoretical knowledge. On several occasions, Engineers are required to compute the exact values of associated lens design parameters, say, focal length of a lens to verify the design data before finalization and submitting the same for production. To compute focal length to verify the design data, and refractive index of glass material of a lens while reverse engineering, requires precise input lens design parameters along with the proper sign conventions. The sign conventions vary with the type and configuration of a lens. Correct results are achieved only with the proper signs conventions as input. This requires some basic understanding of optics. The “LENS”, lens design tool, has been developed by us for System Engineers to determine the exact values of associated lens design parameters, wherein Sign Conventions as input are totally eliminated. Only positive values of lens parameters are entered as input. The System Engineer, on the other hand, should able to give input precisely as - one of the eight configurations of a lens - much simpler than inputting the sign conventions. A guide line has been given in using the design tool “LENS” with proper illustrations. The program has been found useful at various stages of System Design and Development. The “LENS” is also useful for Experienced Optical Engineers to get the reliable results quickly and precisely.

Key Word: Lens Design; Optical System Design & Engineering; Computer Programming in Optical Engineering.

I INTRODUCTION

What is the Lens? The lens is the most basic optical component and form part of the design of nearly all Optical Systems and Instruments. It is an image forming device, collects light from a source and refracts that light to form a usable image of the source. The pencils of light diverging from points of the object are, after refraction by the lens, made to converge toward or diverge from the corresponding point of the image.

All compound and complex lenses are made up of simple lenses. The simple lens is a single element made up of transparent material having two optical surfaces. The compound lens consists of a single group of two or more simple lenses and the complex lens is the one made up of multiple groups of lens elements. All high quality lenses are compound and complex lenses, that is, they consist of number of simple lenses have a common axis. Thus the term lens is ‘generic’ and simple lenses are building blocks of all compound and complex lenses.

II SIMPLE LENS CLASSIFICATION

The simple lenses may be classified by the manner in which they bend light; there are two types of lenses:

1] Convex, Converging, or Positive Lenses [VX or vx] and

2] Concave, Diverging, or Negative Lenses [CV or cv]

Each type can be placed into three groups:

Group 1: Asymmetrical Lenses [AS]

In this group, where R1≠ R2, we can further classify into two configurations as:

a) Bi-Convex Lenses & Bi-Concave Lenses (R1≠R2)

b) **Positive Meniscus Lenses & Negative Meniscus Lenses (R1≠R2).**

Set I equations, given below, are useful to determine the basic lens design parameters of both asymmetrical lenses.

Group 2: Symmetrical [SY] Lenses

In this group we have Bi-Convex Lenses & Bi-Concave Lenses (where R1=R2).

To differentiate from Group 1, we write: Equi-Convex Lenses or Equi-Concave Lenses, since R1=R2.

Set II equations, given below, are useful to determine the basic lens design parameters for symmetrical lenses.

Group 3: Plano [PL] Lenses

The lenses have one curved surface and another plane surface.

In our work we consider second surface as a plane, for both plano-convex lenses and plano-concave lenses.

Set III equations, given below, are useful to determine the lens design parameters for plano lenses.

To summarize, we have two types of lenses: convex and concave, each type can be placed into three groups: asymmetrical, symmetrical and plano. The asymmetrical can be further divided into two, giving in total four different lens configurations, of each type, all these classifications are depicted in Figure 1.0 (A & B) and in Flow chart Figure 2.0.

III LENS MATHEMATICS

Lens maker's equation:

We are familiar with the Lens Maker's Equation as

$$\frac{1}{f} = (n-1)\left[\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n-1)d}{nR_1R_2}\right],$$

This equation can be re-written for our convenience as:

$$\frac{1}{\mathrm{FL}} = (\mathrm{RN}-1)\left[\frac{1}{\mathrm{R1}} - \frac{1}{\mathrm{R2}} + \frac{(\mathrm{RN}-1)\,\mathrm{CT}}{\mathrm{RN}*\mathrm{R1}*\mathrm{R2}}\right] \qquad (1.0)$$

Where:

FL - is the focal length or effective focal length (efl) of the lens,
RN - is the refractive index of the lens material,
R1 - is the radius of curvature of first lens surface (Left side),
R2 - is the radius of curvature of second lens surface (Right side)
CT - is the central (axial) thickness of the lens

SET – I

Re-arranging and re-writing

$$\mathrm{FL} = \frac{\mathrm{R1R2}}{(\mathrm{RN}-1)[(\mathrm{R2}-\mathrm{R1}) + \frac{\mathrm{CT}(\mathrm{RN}-1)}{\mathrm{RN}}} \qquad (1.1)$$

$$\mathrm{R1} = \frac{\mathrm{R2} + \frac{\mathrm{CT}(\mathrm{RN}-1)}{\mathrm{RN}}}{1 + \frac{\mathrm{R2}}{\mathrm{FL}(\mathrm{RN}-1)}} \qquad (1.2)$$

$$\mathrm{R2} = \frac{\mathrm{R1} - \frac{\mathrm{CT}(\mathrm{RN}-1)}{\mathrm{RN}}}{1 - \frac{\mathrm{R1}}{\mathrm{FL}(\mathrm{RN}-1)}} \qquad (1.3)$$

$$\mathrm{CT} = \frac{\mathrm{RN}}{(\mathrm{RN}-1)}\left[\frac{\mathrm{R1R2}}{\mathrm{FL}(\mathrm{RN}-1)} - (\mathrm{R2}-\mathrm{R1})\right] \qquad (1.4)$$

$$\mathrm{RN} = \frac{(\mathrm{R2}-\mathrm{R1}+\mathrm{CT}) + \mathrm{CT} + \frac{\mathrm{R1R2}}{\mathrm{FL}}}{2(\mathrm{R2}-\mathrm{R1}+\mathrm{CT})}$$

$$+\sqrt{\left(\frac{(\mathrm{R2}-\mathrm{R1}+\mathrm{CT}) + \mathrm{CT} + \frac{\mathrm{R1R2}}{\mathrm{FL}}}{2(\mathrm{R2}-\mathrm{R1}+\mathrm{CT})}\right)^2 - \frac{\mathrm{CT}}{(\mathrm{R2}-\mathrm{R1}+\mathrm{CT})}}$$

$$(1.5)$$

Additional equations:

Front Focal Length

$$FFL = FL + \frac{R1 * CT}{RN(R2-R1)+CT(RN-1)} \qquad (1.6)$$

Back Focal Length

$$BFL = FL - \frac{R2D}{N(R2-R1)+D(N-1)} \qquad (1.7)$$

First Principal Point

$$FPP = \frac{R1D}{N(R2-R1)+D(N-1)} = FL - FFL \qquad (1.8)$$

Second Principal Point

$$SPP = \frac{R2D}{N(R2-R1)+D(N-1)} \qquad (1.9)$$

SET II : FOR SYMMETRICAL LENS (where R1 = R2)

$$FL = \frac{R1^2}{(RN-1)\left[2R1 - \frac{CT(RN-1)}{RN}\right]} \qquad (2.1)$$

$$R1 = FL(RN-1)\left(1 + \sqrt{1 - \frac{CT}{RN * FL}}\right) \qquad (2.2)$$

$$CT = \frac{RN}{(RN-1)}\left[2R1 - \frac{R1 * R1}{FL(RN-1)}\right] \qquad (2.3)$$

$$RN = \frac{(2R1-CT)-CT+\frac{R1*R1}{FL}}{2(2R1-CT)} + \sqrt{\left(\frac{(2R1-CT)-CT+\frac{R1*R1}{FL}}{2(2R1-CT)}\right)^2 + \frac{CT}{(2R1-CT)}} \qquad (2.4)$$

SET III: FOR PLANO (PL) LENS, (R2= plane):

Equations for Plano (PL) Convex & Concave Lenses, we have R2 = PLANE

$$FL = \frac{R1}{(RN-1)} \qquad (3.1)$$

$$R1 = FL(RN-1) \qquad (3.2)$$

$$RN = \frac{R1}{FL} + 1 \qquad (3.3)$$

4.0 SIGN CONVENTIONS

The above equations are useful along with the following sign conventions.

1) Light travels from left to right [→]
2) The Radius R1 or R2 is positive if curvature touches first the incident light, and negative if the centre of the curvature of a surface touches first the light.
3) Focal Length FL of converging lens is always positive
4) Focal Length FL of diverging lens is always negative
5) Lens Thickness CT and Refractive Index RN are always positive.

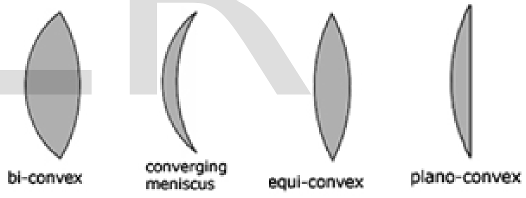
The Basic Lens Configurations of A Simple Lens are:

A] Convex, Converging or Positive Lenses [VX]

i] Bi-convex [BI or bi] ii] Converging Meniscus [MN or mn]

iii]Equi-convex [EQ or eq] iv] Plano-convex [PL or pl]

R1 ≠ R2 R1<R2 R1=R2 R2 = plane



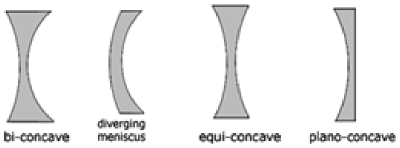
Asymmetrical [AS] Symmetrical[SY] Plano[PL]

B] Concave, Diverging or Negative Lenses [CV]

i] Bi-concave [BI or bi] ii] Negative Meniscus [MN or mn]

iii] Equi-concave [EQ or eq] iv] Plano-concave [PL or pl]

R1 ≠ R2 R1>R2 R1= R2 R2 = plane



Asymmetrical [AS] Symmetrical[SY] Plano[PL]

FIGURE 1.0: BASIC EIGHT LENS CONFIGURATIONS

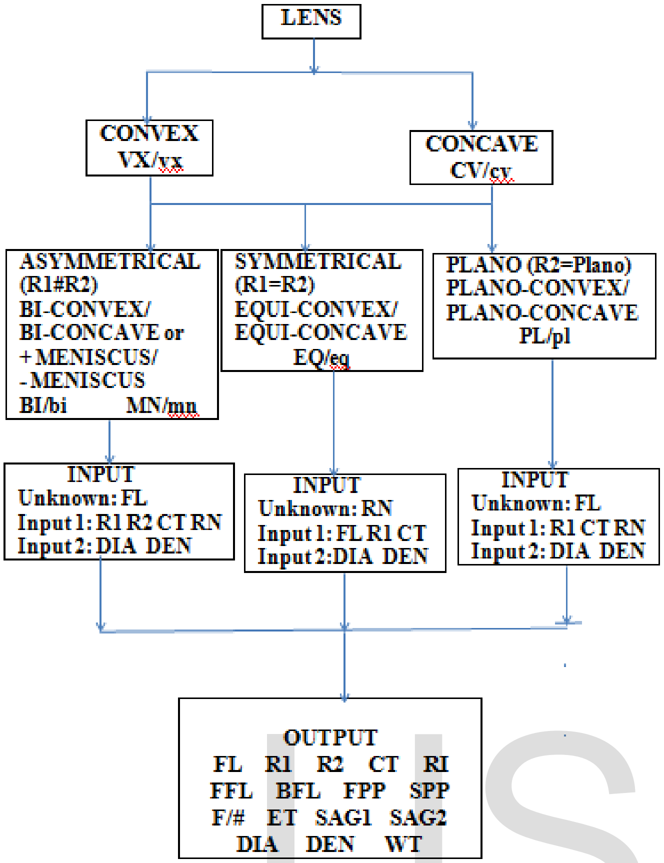


FIGURE 2.0: 'LENS' FLOW CHART

5.0 'LENS' APPLICATIONS

To completely understand the tool "LENS", consider the following applications & illustrations:

1] Design Data Verification

It is essential for any system engineer to verify the design data R1, R2, CT & RN before finalizing a design and going into the production. Remember, the manufacture of a lens is time-taking and uneconomical process. FL determination using "LENS" and matching its value with the original verify the design data.

Illustration 1.0: Given R1 = 11.86; R2=11.86; CT=6.68; RN=1.5167; FL?

Solution: Lens Type: Converging Lens

OUTPUT:

FL=12.7;R1=11.86; R2= -11.86 ; CT=6.68; RI=1.5167

FFL=-10.26 BFL=10.26;FPP=2.43; SPP=-2.43;ET=3.00

2] Reverse Engineering

Optical lens design parameters FL, R1, R2 & CT can be easily measured using optical and mechanical methods, except refractive index RN. To determine RN the program "LENS" is found useful.

Illustration 2.0: You have a lens with the following measured lens design data:

FL=27mm; R1 =19.7mm; R2=-50.27mm & CT = 5 mm. wish to make a similar one, what is the refractive index of the glass material of the lens?

Solution:

Given data: FL = 27mm; R1=19.7mm; R2=50.27; CT=5mm
To compute RN=?

Asymmetrical Converging Lens: Bi-convex Lens (R1≠ R2)

OUTPUT:

FL = 27; R1=19.7; R2=-50.27; CT=5; RN=1.5376

FFL= -26.06; BFL= 24.60; FPP=-0.94 ; SPP=-2.4 ;

3] Computation of unknown Individual Lens Design Parameter

Any unknown lens parameter, FL, R1, R2, CT, or RI can be easily determined using "LENS" having known the other lens design parameters.

Illustration 3.0: You wish to design a lens with a focal length of 125 mm. The first radius must be 128.16 mm to be compatible with an existing system. What is the second radius when the thickness at the center is 4.25 mm and the refractive index is 1.5167?

Solution:

FL = 125; R1=128.16mm;   R2=-128.16; CT=4.25; RI=1.5167

6.0 PROGRAM ' LENS' FEATURES

The program "LENS" developed has the following features:

1] The lens data entered as input, for computation of lens parameter, have always positive values, regardless of the shape and type of a simple lens. No sign is given as input. The sign conventions are taken care by the program itself.

2] The Configurations: Convex [VX] or Concave [CV]; Asymmetrical [AS], Symmetrical [SY] or Plano [PL]; Shapes BI, MN, EQ, PL should be know precisely by the user as input.

2] Program accepts Capital as well as Small Letters as input.

3] The input parameters required are displayed in prior.

4] The associated lens parameters: F/#, FFL, BFL, FPP, SPP, WT (weight) is often determined given the input diameter (DIA) and density (DEN) of a lens material.

CONCLUSION

"LENS" is an useful tool for system engineer using optics and for experienced optical engineer. The proliferation of tool has minimized the need for calculation by hand and to eliminate sign conventions as input, which varies with the type and configuration of the lens. The "LENS" developed is user-friendly and requires limited knowledge of Optics. The input lens parameters are always in positive quantity, thereby eliminating and avoiding the risk of confusion of sign conventions given at input. The "LENS" output gives all the parameters with proper sign conventions. The "LENS" enables us to compute in addition, F/#, FFL, BFL, FPP, SPP & WT, these are associated parameters found useful at various stages of system design and development. The programming language used is C++, however any computer language can be used. The program has been tested for all eight configurations using standard illustrations given in texts [1][2]and catalog[3] The "LENS" is in use as a quick and reliable design tool at various stages of system development by us. The best way to learn LENS is to use it. The program listing in C++ is given in Appendix.

REFERENCES

[1] Warren J Smith,

MODERN OPTICAL ENGINEERING,

The McGraw Hill, 2008

[2] Rudolf Kingslake

LENS DESIGN FUNDAMENTALS,

Academic Press, Inc, 1978

[3] Catalog: SPINDLER & HOYER, USA

ACKNOWLEDGEMENT

The authors are thankful to Dr Prakash O Borole for his valuable guidance at all stages. Thanks are also due to Management of Alard College of Engg & Management, Marunje, University of Pune, Pune for valuable support and permitting us to publish the research work.

```
/*Program Listing*/
/*PROGRAM TO COMPUTE LENS DESIGN
PARAMETERS OF A SIMPLE LENS */
/*DECLARATION OF STANDARD FILES */
#include<stdio.h>
#include<string.h>
#include<conio.h>
#include<dos.h>
#include<iostream.h>
#include<math.h>
#include<stdlib.h>
/*DEFINING THE CONSTANT  */
#define pi 3.14159265
/* MAIN PROGRAM  */
void main() {  int N;
float   DIA,FNO,ET,P1,P2,WT,SR,VC,V1,V2,VOL,DEN;
float FL,R1,R2,CT,RN;  float fl,r1,r2,ct,rn;
float
ASFL,ASR1,ASR2,ASCT,ASRN,SYFL,SYR1,SYCT,SYRN,
PLFL,PLR1,PLRN;
float RNA,RNB,FPP,SPP,FFL,BFL,SAG1,SAG2;
```

```
char ch[3];  clrscr();  printf("\n");
printf("\n\tPROGRAM TO COMPUTE INDIVIDUAL
LENS DESIGN PARAMETER OF A SIMPLE LENS\n");
printf("\n");
printf("\n\t\t\t  ........................................press enter \n");
getch(); clrscr();
{   M:  l:
printf("\n\t ENTER LENS: [TWO TYPES] CONVEX [VX]
or CONCAVE [CV]\n");  printf("\n");
printf("\n\t FOR CONVEX LENS ... enter... [VX or vx]\n");
printf("\n\t FOR CONCAVE LENS... enter... [CV or cv]\n");
printf("\n");
printf("\t   TO SKIP THIS LOOP...enter... [QT or qt]\n");
printf("\t   TO SKIP PROGRAM  ...enter... [EX or ex]\n");
printf("\n");  scanf("%s", &ch);
if(strcmp(ch,"VX")==0||strcmp(ch,"vx")==0){N=1; goto NS;}
if(strcmp(ch,"CV")==0||strcmp(ch,"cv")==0){N=2; goto NS;}
if(strcmp(ch,"QT")==0||strcmp(ch,"qt")==0){goto M ;}
if(strcmp(ch,"EX")==0||strcmp(ch,"ex")==0){exit(0);} else {
cout<<"\nERROR: Enter correctly: VX/vx/CV/cv  or
Q/q/EX/ex\n"; goto l; }}
NS: {  ll:
printf("\n\tENTER LENS:[FOUR CONFIGURATIONS
BI;MN;SY;PL]\n\n");
printf("\n");
printf("\t  FOR ASY BI-VEX /BI-CAVE LENS..enter..[BI or
bi]\n");
printf("\t  FOR ASY +VE MN /-VE MN  LENS..enter..[MN
or mn]\n");
printf("\t  FOR  SY LENS              ..enter..[SY or sy]\n");
printf("\t  FOR  PL LENS              ..enter..[PL or pl]\n");
printf("\n");
printf("\t TO SKIP THIS LOOP ... enter.......... [QT or
qt]\n");
printf("\t TO SKIP PROGRAM   ... enter....... ...[EX or
ex]\n"); printf("\n");   scanf("%s",&ch);
if(strcmp(ch,"BI") ==0||strcmp(ch,"bi")==0)  {goto BI;}
if(strcmp(ch,"MN") ==0||strcmp(ch,"mn")==0)  {goto MN;}
if(strcmp(ch,"SY")==0||strcmp(ch,"sy")==0) {goto SY;}
if(strcmp(ch,"PL")==0||strcmp(ch,"pl")==0) {goto PL;}
if(strcmp(ch,"QT")==0||strcmp(ch,"qt")==0){goto M ;}
if(strcmp(ch,"EX")==0||strcmp(ch,"ex")==0){exit(0);} else   {
cout<<"\nERROR: Enter correctly:
BI/bi/MN/mn/SY/sy/PL/pl or Q/q/EX/ex\n";
goto ll; }}
NS1: {
FPP=-CT*FL*(RN-1)/(R2*RN); FFL=  FL-FPP;
SPP=-CT*FL*(RN-1)/(R1*RN); BFL=  FL+SPP;
printf("\n------------------------------------------------------------------
--------\n");
printf("\nFL=%f\tR1=%0.4f\tR2=%0.4f\tCT=%0.2f\tRN=%f\
n",FL,R1,R2,CT,RN);
printf("\n------------------------------------------------------------------
--------\n");
printf("\nFFL=%f\tBFL=%0.4f\tFPP=%0.4f\tSPP=%0.2f\n",F
FL,BFL,FPP,SPP);
printf("\n------------------------------------------------------------------
--------\n");
cout<<"\nENTER: Diameter of Lens\n"; cin>>DIA;
cout<<"\nENTER: Density of Glass material of a lens \n";
cin>>DEN; FNO=FL/DIA;
P1=sqrt(R1*R1-DIA*DIA/4); P2=sqrt(R2*R2-DIA*DIA/4);
R1=sqrt(R1*R1); R2=sqrt(R2*R2); SAG1=R1-P1; SAG2=R2-
P2;
if(N==1){ET=CT-SAG1-SAG2;}
if(N==2){ET=CT+SAG1+SAG2;}
if(N==3||N==4){ET=CT-SAG1+SAG2;}
VC=pi*DIA*DIA*ET/4; V1=pi*SAG1*SAG1*(3*R1-SAG1);
V2=pi*SAG2*SAG2*(3*R2-SAG2);
if(N==1){VOL=(VC+V1+V2)/1000;} if(N==2){VOL=(VC-V1-
V2)/1000;}
if(N==3){VOL=(VC+V1-V2)/1000;} if(N==4){VOL=(VC+V1-
V2)/1000;}
printf("\nVOL=%f\tVC=%f\tV1=%f\tV2=%f\n",VOL,VC,V
1,V2);
printf("\n------------------------------------------------------------------
-------\n");
printf("\nDIA=%f\tF/#=%f\tET=%f\tSAG1=%f\tSAG2=%0.
4f\n",DIA,FNO,ET,SAG1,SAG2);
printf("\n------------------------------------------------------------------
-------\n"); WT=DEN*VOL;
printf("\nLENS WEIGHT  WT=%f              DENSITY OF
GLASS   DEN=%f\n",WT,DEN);
printf("\n------------------------------------------------------------------
-------\n");
printf("\n\t\t\t...........END..................to continue press
enter"); getch(); clrscr();
FL=0;R1=0;R2=0;CT=0;RN=0;DIA=0;ET=0;
FFL=0;BFL=0;FPP=0;SPP=0;DIA=0;FNO=0;ET=0;SAG1=0;S
AG2=0;WT=0;DEN=0; goto M; }
MN:  { if(N==1){N=3;} if(N==2){N=4;}}
BI:  { lll:
printf("\tFOR AS BI-/MN-LENS [R1#R2]:DETERMINE
PARAMETER FL;R1;R2;CT;RN]\n"); printf("\n");
printf("\tBI-LENS/MENISCUS FL ... enter... [ASFL or
asfl]\n");
printf("\tBI-LENS/MENISCUS R1 ... enter... [ASR1 or
asr1]\n");
printf("\tBI-LENS/MENISCUS R2 ... enter... [ASR2 or
asr2]\n");
```

```
printf("\tBI-LENS/MENISCUS CT ... enter... [ASCT or
asct]\n");
printf("\tBI-LENS/MENISCUS RN ... enter... [ASRN or
asrn]\n"); printf("\n");
printf("\tTO SKIP THIS LOOP ... enter .....[QT or qt]\n");
printf("\t TO SKIP PROGRAM   ... enter.....[EX or ex]\n");
printf("\n"); scanf("%s",&ch);
if(strcmp(ch,"ASFL")==0||strcmp(ch,"asfl")==0){goto
ASFL;}
if(strcmp(ch,"ASR1")==0||strcmp(ch,"asr1")==0){goto
ASR1;}
if(strcmp(ch,"ASR2")==0||strcmp(ch,"asr2")==0){goto
ASR2;}
if(strcmp(ch,"ASCT")==0||strcmp(ch,"asct")==0){goto
ASCT;}
if(strcmp(ch,"ASRN")==0||strcmp(ch,"asrn")==0){goto
ASRN;}
if(strcmp(ch,"QT")==0||strcmp(ch,"qt")==0){goto M ;}
if(strcmp(ch,"EX")==0||strcmp(ch,"ex")==0){exit(0);} else {
cout<<"\nERROR: Enter correctly:
ASFL/asfl/ASR1/asr1/ASR2/asr2/ASCT/asct/ASRN/asrn or
Q/q/EX/ex\n"; goto lll; }}
ASFL: {
printf("\n\tENTER R1, R2 CT, RN  ...+ve VALUES
ONLY\n");  printf("\n");
scanf("%f%f%f%f", &R1, &R2, &CT, &RN);
printf("\n");
if(N==1){R2=-R2;} if(N==2){R1=-R1;}
if(N==3){R1=R1; R2=-R2;} if(N==4){R1=R1;R2=R2;}
FL=R1*R2/((RN-1)*((R2-R1)+(CT*(RN-1)/RN)));
goto NS1;  }
ASR1:  {
printf("\n\tENTER FL,R2, CT, RN  ...+ve VALUES
ONLY\n"); printf("\n");
scanf("%f%f%f%f", &FL, &R2, &CT, &RN);
printf("\n");
if(N==1){R2=-R2;}  if(N==2){FL=-FL;R2=R2;}
if(N==3){R2=-R2;}  if(N==4){FL=-FL;R2=R2;}
R1=(R2+(CT*(RN-1)/RN))/(1+R2/(FL*(RN-1)));
goto NS1;  }
ASR2:  {
printf("\n\tENTER FL, R1, CT, RN  ...+ve VALUES
ONLY\n"); printf("\n");
scanf("%f%f%f%f", &FL, &R1, &CT, &RN);
printf("\n");
if(N==1){R1=R1;} if(N==2){FL=-FL;R1=-R1;}
if(N==3){R1=R1;}  if(N==4){FL=-FL;R1=R1;}
R2=(R1-(CT*(RN-1)/RN))/(1-R1/(FL*(RN-1)));
goto NS1;  }
ASCT: {
printf("\n\tENTER FL, R1, R2, RN  ...+ve VALUES
ONLY\n");  printf("\n");
scanf("%f%f%f%f", &FL, &R1, &R2, &RN);
printf("\n");
if(N==1){R2=-R2;} if(N==2){FL=-FL;R1=-R1;}
if(N==3){R2=-R2;}  if(N==4){FL=-FL;R1=R1;}
CT=(RN/(RN-1))*((R1*R2/(FL*(RN-1)))-(R2-R1));
goto NS1;  }
ASRN: {
printf("\n\tENTER FL, R1, R2, CT, ...+ve VALUES
ONLY\n"); printf("\n");
scanf("%f%f%f%f", &FL, &R1, &R2, &CT);
printf("\n");
if(N==1){R2=-R2;} if(N==2){FL=-FL;R1=-R1;}
if(N==3){R2=-R2;}  if(N==4){FL=-FL;R1=R1;}
RNA=(R2-R1+CT+CT+R1*R2/FL)/(2*(R2-R1+CT));
RNB=pow(RNA*RNA-(CT/(R2-R1+CT)),0.5);
RN=RNA+RNB; goto NS1;  }
NS2: {
R2=-R1; FPP=CT*FL*(RN-1)/(RN*R1); FFL=FL-FPP;
SPP=-FPP;  BFL=FL+SPP;
printf("\n------------------------------------------------------------------
-----\n");
printf("\nFL=%f\tR1=%0.4f\tR2=%0.4f\tCT=%0.2f\tRN=%f\
n",FL,R1,R2,CT,RN);
printf("\n------------------------------------------------------------------
-----\n");
printf("\nFFL=%f\tBFL=%0.4f\tFPP=%0.4f\tSPP=%0.2f\n",F
FL,BFL,FPP,SPP);
printf("\n------------------------------------------------------------------
-----\n");
cout<<"\nENTER:Diameter of Lens\n"; cin>>DIA;
cout<<"\nENTER:Density of Glass material of lens\n";
cin>>DEN; FNO=FL/DIA;
P1=sqrt(R1*R1-DIA*DIA/4); R1=sqrt(R1*R1); SAG1=R1-
P1; SAG2=SAG1;
if(N==1){ET=CT-SAG1-SAG2;}
if(N==2){ET=CT+SAG1+SAG2;}
VC=pi*DIA*DIA*ET/4; V1=pi*SAG1*SAG1*(3*R1-
SAG1);V2=V1;
if(N==1){VOL=(VC+V1+V2)/1000;} if(N==2){VOL=(VC-V1-
V2)/1000;}
printf("\nVOL=%f\tVC=%f\tV1=%f\tV2=%f\n",VOL,VC,V
1,V2);
printf("\n------------------------------------------------------------------
---------\n");
printf("\nDIA=%f\tF/#=%f\tET=%f\tSAG1=%f\tSAG2=%0.
4f\n",DIA,FNO,ET,SAG1,SAG2);
printf("\n------------------------------------------------------------------
---------\n");  WT=DEN*VOL;
```

```
printf("\nLENS WEIGHT  WT=%f          DENSITY OF
GLASS   DEN=%f\n",WT,DEN);
printf("\n-------------------------------------------------------------------
---------\n");
printf("\n\t\t\t............END................to continue press
enter"); getch();  clrscr();
FL=0;R1=0;R2=0;CT=0;RN=0;DIA=0;ET=0;
FFL=0;BFL=0;FPP=0;SPP=0;DIA=0;FNO=0;ET=0;SAG1=0;S
AG2=0;WT=0;DEN=0;  goto M;  }
SY: { llll:
printf("\tFOR SY LENS [R1=R2]: DETERMINE
PARAMETER FL:R1:CT:RN\n"); printf("\n");
printf("\t SYMM-LENS FL ... enter... [SYFL or syfl]\n");
printf("\t SYMM-LENS R1 ... enter... [SYR1 or syr1]\n");
printf("\t SYMM-LENS CT ... enter... [SYCT or syct]\n");
printf("\t SYMM-LENS RN ... enter... [SYRN or syrn]\n");
printf("\n"); printf("\t TO SKIP THIS LOOP... enter...[QT
or qt]\n");
printf("\t TO SKIP PROGRAM  ... enter...[EX or ex]\n");
printf("\n");  scanf("%s",&ch);
if(strcmp(ch,"SYFL")==0||strcmp(ch,"syfl")==0){goto
SYFL;}
if(strcmp(ch,"SYR1")==0||strcmp(ch,"syr1")==0){goto
SYR1;}
if(strcmp(ch,"SYCT")==0||strcmp(ch,"syct")==0){goto
SYCT;}
if(strcmp(ch,"SYRN")==0||strcmp(ch,"syrn")==0){goto
SYRN;}
if(strcmp(ch,"QT")==0||strcmp(ch,"qt")==0){goto M ;}
if(strcmp(ch,"EX")==0||strcmp(ch,"ex")==0){exit(0);} else {
cout<<"\nError: Enter
correctly:SYFL/syfl/SYR1/syr1/SYCT/syct/SYRN/syrn or
Q/q/EX/ex\n";   goto llll; }}
SYFL: {
printf("\n\tENTER R1, CT, RN ...+ve VALUES ONLY\n");
printf("\n");
scanf("%f%f%f", &R1, &CT, &RN); printf("\n");
if(N==2){R1=-R1;}
FL=(R1*R1)/((RN-1)*((2*R1)-(CT*(RN-1)/RN)));
goto NS2; }
SYR1: {
printf("\n\tENTER FL, CT, RN ...+ve VALUES ONLY\n");
printf("\n");
scanf("%f%f%f", &FL, &CT, &RN);   printf("\n");
if(N==2){FL=-FL;}
SR=pow(1-(CT/(RN*FL)),0.5);
R1=FL*(RN-1)*(1+SR);  goto NS2;  }
SYCT:  {
printf("\n\tENTER FL, R1, RN ...+ve VALUES ONLY\n");
printf("\n");
scanf("%f%f%f", &FL, &R1, &RN);
printf("\n"); if(N==2){FL=-FL;R1=-R1;}
CT=(RN/(RN-1))*(2*R1-(R1*R1/(FL*(RN-1)))); goto NS2;  }
SYRN: {
printf("\n\tENTER FL, R1, CT ...+ve VALUES ONLY\n");
printf("\n");
scanf("%f%f%f", &FL, &R1, &CT); printf("\n");
if(N==2){FL=-FL;R1=-R1;}
RNA=(2*R1-2*CT+R1*R1/FL)/(2*(2*R1-CT));
RNB=pow(RNA*RNA+(CT/(2*R1-CT)),0.5);
RN=RNA+RNB; goto NS2;  }
NS3:{ R2=0; FPP=0; FFL=FL-FPP; SPP=-CT/RN;
BFL=FL+SPP;
printf("\n-------------------------------------------------------------------
----\n");
printf("\nFL=%f\tR1=%0.4f\tR2=%0.4f\tCT=%0.2f\tRN=%f\
n",FL,R1,R2,CT,RN);
printf("\n-------------------------------------------------------------------
----\n");
printf("\nFFL=%f\tBFL=%0.4f\tFPP=%0.4f\tSPP=%0.2f\n",F
FL,BFL,FPP,SPP);
printf("\n-------------------------------------------------------------------
----\n");
cout<<"\nENTER:Diameter of Lens\n";  cin>>DIA;
cout<<"\nENTER: Density of Glass material of lens \n";
cin>>DEN; FNO=FL/DIA;
P1=sqrt(R1*R1-DIA*DIA/4); R1=sqrt(R1*R1); SAG1=R1-
P1; SAG2=0;
if(N==1){ET=CT-SAG1;} if(N==2){ET=CT+SAG1;}
VC=pi*DIA*DIA*ET/4; V1=pi*SAG1*SAG1*(3*R1-SAG1);
V2=0;
if(N==1){VOL=(VC+V1)/1000;} if(N==2){VOL=(VC-
V1)/1000;}
printf("\nVOL=%f\tVC=%f\tV1=%f\tV2=%f\n",VOL,VC,V
1,V2);
printf("\n-------------------------------------------------------------------
----\n");
printf("\nDIA=%f\tF/#=%f\tET=%f\tSAG1=%f\tSAG2=%0.
4f\n",DIA,FNO,ET,SAG1,SAG2);
printf("\n-------------------------------------------------------------------
----\n");  WT=DEN*VOL;
printf("\nLENS WEIGHT  WT=%f             DENSITY OF
GLASS   DEN=%f\n" ,WT,DEN);
printf("\n-------------------------------------------------------------------
----\n");
printf("\n\t\t\t...........END.................to continue press
enter"); getch(); clrscr();
FL=0;R1=0;R2=0;CT=0;RN=0;DIA=0;ET=0;
FFL=0;BFL=0;FPP=0;SPP=0;DIA=0;FNO=0;ET=0;SAG1=0;S
AG2=0;WT=0;DEN=0; goto M;
```

```
PL: { lllll:
printf("\t PL-LENS [R2=0] :DETERMINE PARAMETER
FL;R1;RN\n"); printf("\n");
printf("\t PLANO-LENS FL ... enter...[PLFL or plfl]\n");
printf("\t PLANO-LENS R1 ... enter...[PLR1 or plr1]\n");
printf("\t PLANO-LENS RN ... enter...[PLRN or plrn]\n");
printf("\n");
printf("\t TO SKIP THIS LOOP ... enter...[QT or qt]\n");
printf("\t TO SKIP PROGRAM   ... enter...[EX or ex]\n");
printf("\n");  scanf("%s",&ch);
if(strcmp(ch,"PLFL")==0||strcmp(ch,"plfl")==0){goto
PLFL;}
if(strcmp(ch,"PLR1")==0||strcmp(ch,"plr1")==0){goto
PLR1;}
if(strcmp(ch,"PLRN")==0||strcmp(ch,"plrn")==0){goto
PLRN;}
if(strcmp(ch,"QT")==0||strcmp(ch,"qt")==0){goto M ;}
if(strcmp(ch,"EX")==0||strcmp(ch,"ex")==0){exit(0);} else  {
cout<<"\nError: Enter
correctly:PLFL/plfl/PLR1/plr1/PLRN/plrn or Q/q/EX/ex\n";
goto lllll; }}
PLFL: {
printf("\n\tENTER R1, CT, RN ...+ve VALUES ONLY\n");
printf("\n");
scanf("%f%f%f", &R1, &CT, &RN); printf("\n");
if(N==2){R1=-R1;} FL=R1/(RN-1); goto NS3;   }
PLR1: {
printf("\n\tENTER FL, CT, RN ...+ve VALUES ONLY\n");
printf("\n");
scanf("%f%f%f", &FL, &CT, &RN); printf("\n");
if(N==2){FL=-FL;} R1=FL*(RN-1); goto NS3;   }
PLRN: {
printf("\n\tENTER FL, R1, CT ...+ve VALUES ONLY\n");
printf("\n");
scanf("%f%f%f", &FL, &R1, &CT);
printf("\n"); if(N==2){FL=-FL;R1=-R1;}
RN=1+R1/FL; goto NS3; } }}
 /*  END   */
```