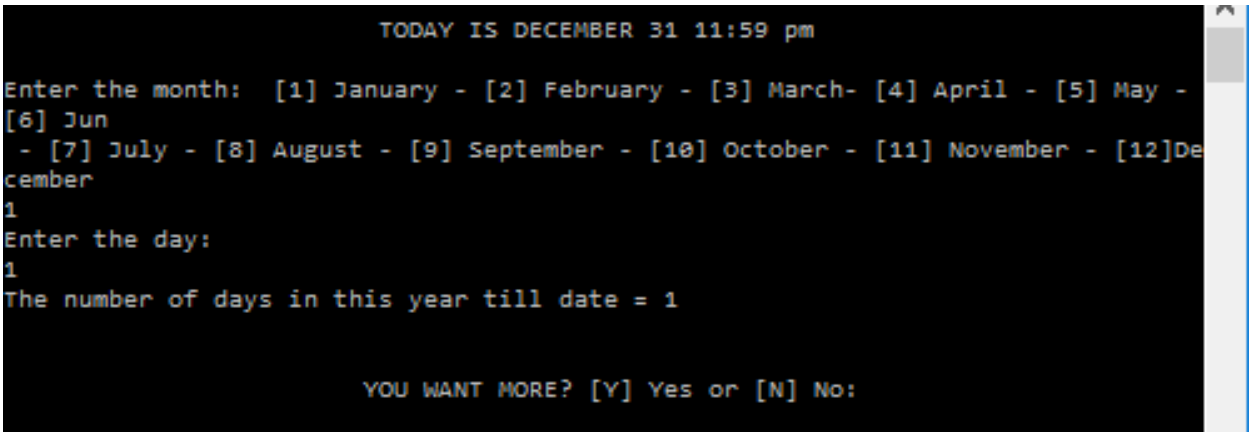
Days till the date within the year
Sample Input - Output:

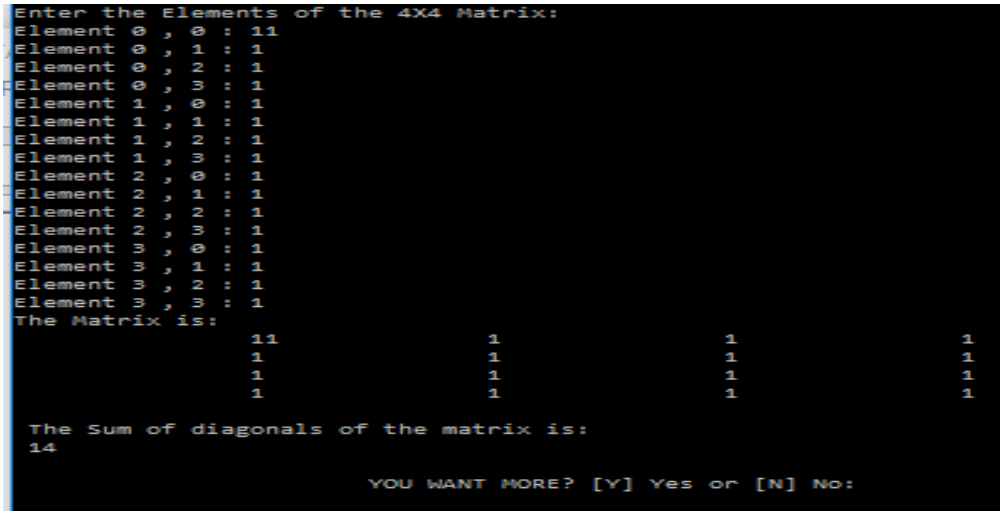


Source Code:

```
unit days_permonth[12] =
{31,28,31,30,31,30,31,31,30,31,30,31};
unit x;
PrimaryMission() {


unit day,month,total,var,choice;

go {
commence;
post("\n\t\t\t TODAY IS DECEMBER 31
11:59 pm \n");

post("\nEnter the month:  [1] January -
[2] February - [3] March- [4] April -
[5] May - [6] Jun \n - [7] July - [8]
August - [9] September - [10] October -
[11] November - [12]December\n");
capture(#month);
phase(month<0 || month>12 ) {
post("Please Enter A Number from [1 to
12]\n" );
capture(#month);
}
inorder(month = =2) {
post("Enter the day: \n");
capture(#day);
phase(day<=0 || day>28 ) {
post("Please Enter A Number for [1 to
31]
\n" );
capture(#day);
}
}
otherorder(month = =4) {

post("Enter the day: \n");
capture(#day);
phase(day<=0 || day>30 ) {
post("Please Enter A Number for [1 to
30]\n" );
capture(#day);
}
}
otherorder(month = =9) {
post("Enter the day: \n");
capture(#day);
phase(day<=0 || day>30 ) {
post("Please Enter A Number for [1 to
30] \n" );
capture(#day);
}
}
otherorder(month = = 11) {
post("Enter the day: \n");
capture(#day);
phase(day<=0 || day>30 ) {
post("Please Enter A Number for [1 to
30] \n");
capture(#day);
```

```
}
}
order {
post("Enter the day: \n");
capture(#day);
phase(day<=0 || day>32 ) {
post("Please Enter A Number for [1 to
31]\n" );
capture(#day);
}
}
total=   day - days_permonth[x] ;
var= month;
inquire(x=0;x<var;x++) {
total = total + days_permonth[x];
}
post("The number of days in this year
till date = " +total+ "\n");

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
day=0;
month=0;
var=0;
x=0;
total=0;

}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

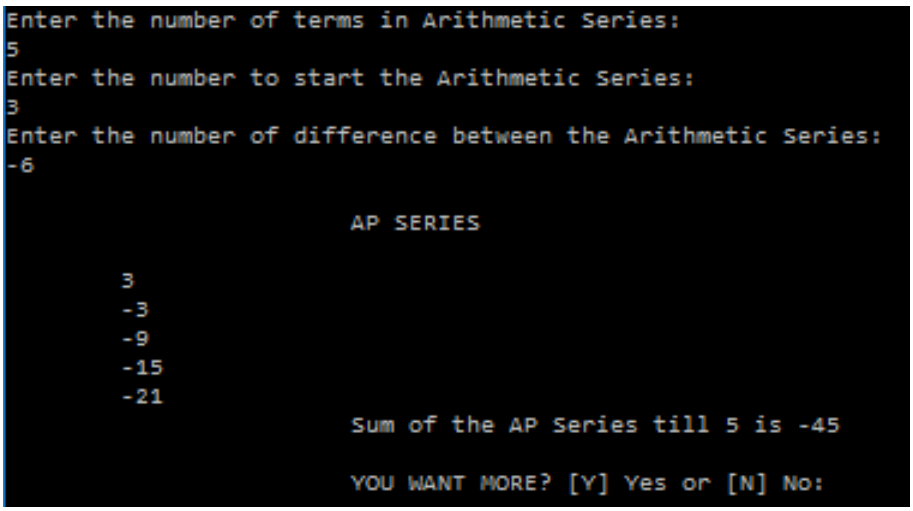
Matrix Diagonal SUM
Sample Input - Output:



Source Code:

```
unit num[4][4];
unit i=0;
unit j=0;
unit sum=0;

PrimaryMission() {
unit choice;
go {
commence;
post("Enter the Elements of the 4X4
Matrix: \n");
inquire(i=0;i<4;i++) {
inquire(j=0;j<4;j++) {
post("Element " + i + " , " + j + " :
");
capture(#num[i][j]);
}
}
post("The Matrix is: \n ");
inquire(i=0;i<4;i++) {
inquire(j=0;j<4;j++) {
post("\t\t" + num[i][j] + " ");
}
post("\n");
}
post("\n The Sum of diagonals of the
matrix is:\n ");
inorder(i = = j) {
inquire(i=0;i<4;i++) {
inquire(j=0;j<4;j++) {
inorder(i = = j) {
sum = sum + num[i][j];
}}}post(sum);
}order {
post("The Sum is not possible imbalance
Matrix : \n ");
}
go { company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: "); capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1; i=0;
j=0;sum=0;
} otherorder((ch = = "N") || (ch = =
"n")) { choice = 0;
} order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

Sum of Arithmetic Series
Sample Input - Output:



Source Code:

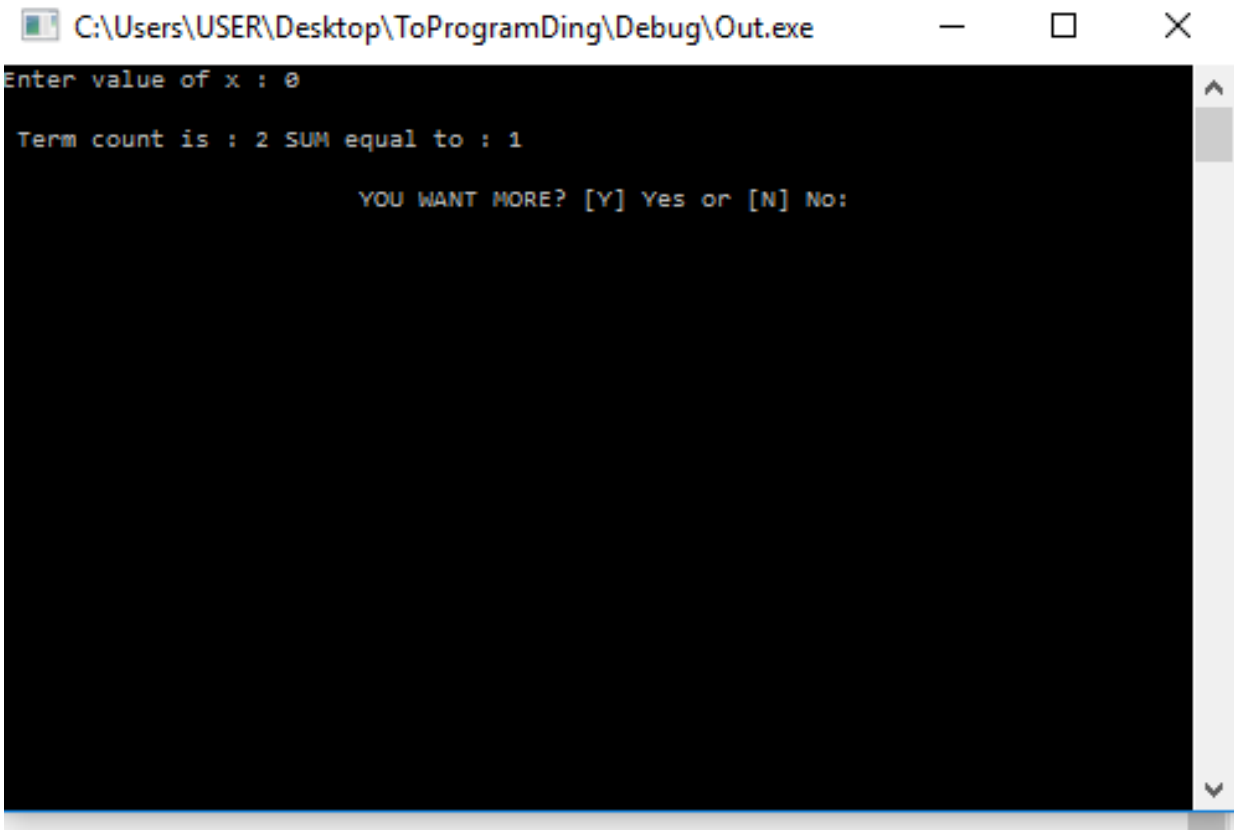
```
unit value;
PrimaryMission() {
unit first,diff,terms,sum=0, i,choice;
go {
commence;
post("Enter the number of terms in
Arithmetic Series: \n");
capture(#terms);
post("Enter the number to start the
Arithmetic Series: \n");
capture(#first);
post("Enter the number of difference
between the Arithmetic Series: \n");
capture(#diff);
value=first;
post("\n\t\t\tAP SERIES \n ");

inquire(i=0;i<terms;i++) {
post("\n\t" + value + " " );
sum= sum+value;
value = value + diff;
}
post("\n\t\t\tSum of the AP Series till
" +terms+ " is " +sum+ " ");

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");

capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
first=0;
diff=0;
terms=0;
sum=0;
i=0;
value=0;
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

Commision of Sales agent base on Sales
Sample Input - Output

```
Enter the total sale value of an agent : -90000
Please enter a positive amount of money: 90000
For a total sale value of : 90000
 The Agent's Commission is : 28800


                        YOU WANT MORE? [Y] Yes or [N] No:
```

Source Code:

```
digit svalue;
digit com,c;
PrimaryMission() {
unit choice;
go {
commence;
post("Enter the total sale value of an
agent : ");
capture(#svalue);
phase(svalue<0) {
post("Please enter a positive amount of
money: ");
capture(#svalue);
}
inorder(svalue <= 10000) {
c =5.0 / 100;
com = svalue * c;
post("For a total sale value of : "
+svalue+ " \n ");
post("The Agent's Commission is : "
+com+ " \n ");
}
otherorder(svalue <= 20000) {
c =10.0 / 100;
com = svalue * c;
post("For a total sale value of : "
+svalue+ " \n ");
post("The Agent's Commission is : "
+com+ " \n ");
}
otherorder(svalue <= 30000) {
c =15.0 / 100;
com = svalue * c;
post("For a total sale value of : "
+svalue+ " \n ");
post("The Agent's Commission is : "
+com+ " \n ");
}
otherorder(svalue <= 50000) {
c = 25.0 / 100;
com = svalue * c;
post("For a total sale value of : "
+svalue+ " \n ");
post("The Agent's Commission is : "
+com+ " \n ");
}
order {
c =32.0 / 100;
com = svalue * c;
post("For a total sale value of : "
+svalue+ " \n ");
post("The Agent's Commission is : "
+com+ " \n ");
}

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
svalue=0;
c=0;
com=0;

}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
```

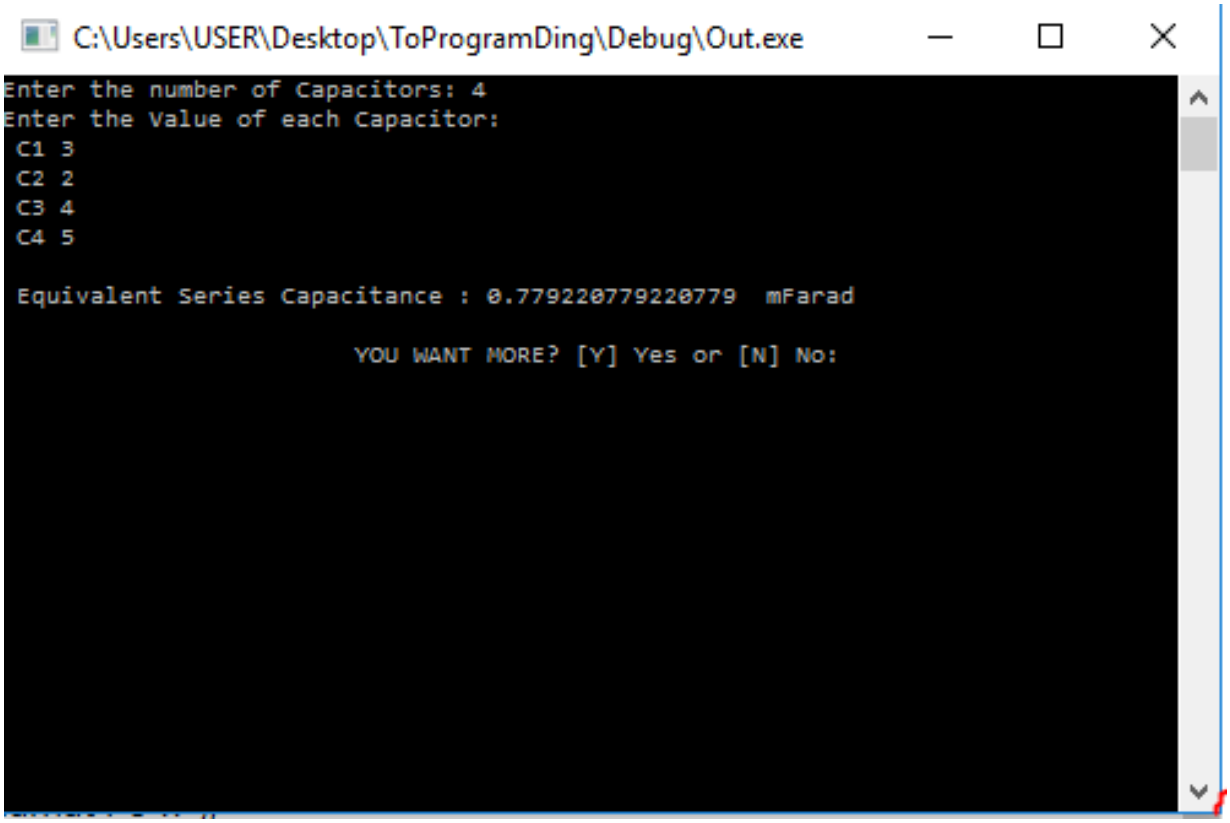



```
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

Source Code: Taylor Series or Power Of Exponents

```
digit accuracy = 0.0001;
PrimaryMission() {

unit n,count,choice;
digit x, term, sum=0.0;

go {
commence;

post("Enter value of x : ");
capture(#x);

n=1;
term=1;
sum=1;
count=1;

phase(n<=100) {
term = term*(x/n);
sum = sum + term;
count = count +1;
```

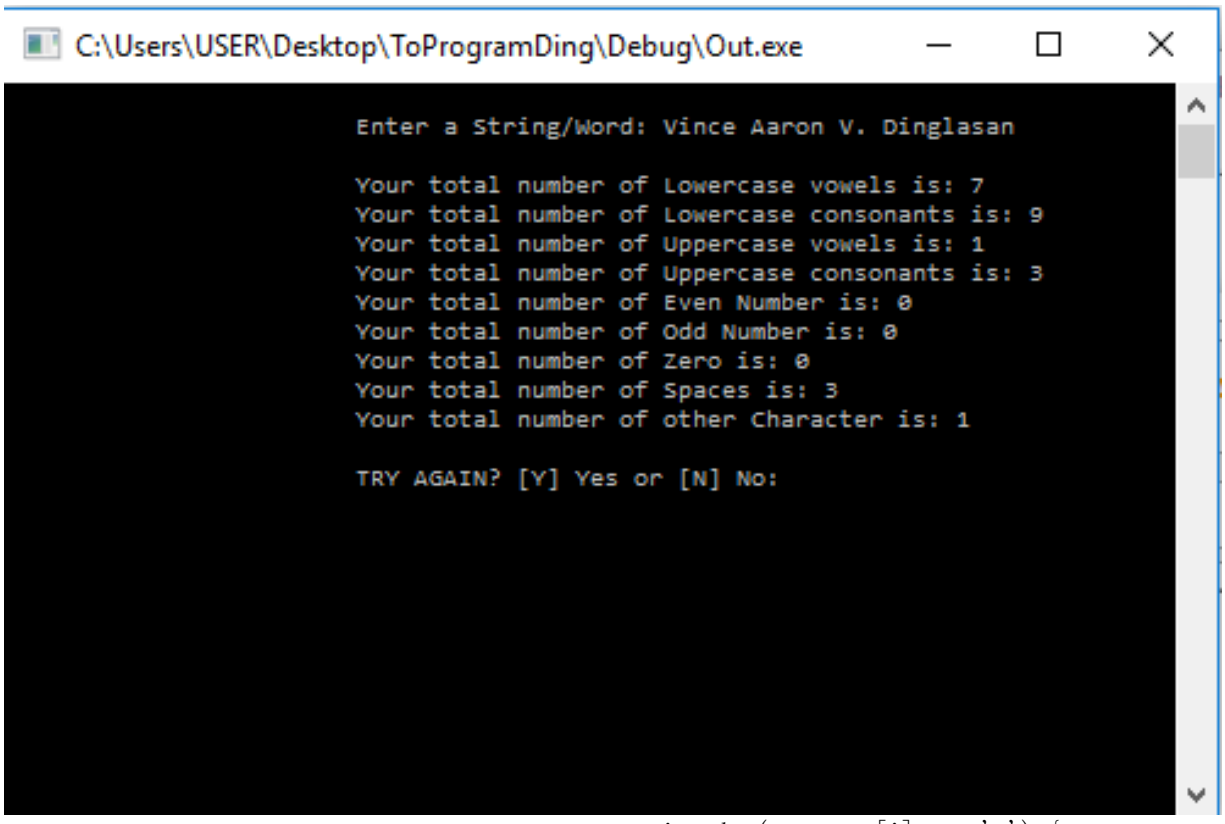
```
inorder(term<accuracy) {
n =999;
}
order {
n = n+1;
}
}
post("\n Term count is : " +count+ "
SUM equal to : " +sum+ " ");

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
n=0;
term=0;
sum=0;
x=0;
count=0;
accuracy = 0.0001;


}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```



Source Code: Equivalent Capacitance of Series Circuit

```
digit c[10];
digit num, Cs=0;
digit var, var2;
unit i;

PrimaryMission() {
unit choice;

go {
commence;
post("Enter the number of Capacitors:
");
capture(#num);

post("Enter the Value of each
Capacitor: \n");

inquire(i=0;i<num;i++) {
var = i + 1;
post(" C" + var  + " ");
capture(#c[i]);
}
inquire(i=0;i<num;i++) {
var2 = 1/c[i];
Cs = Cs + var2;
}
Cs = 1/Cs;

post(" \n Equivalent Series Capacitance
: " + Cs + "  mFarad ");

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
num=0;
Cs=0;
var=0;
var2=0;
i=0;
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;

}
```

```
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```



Source Code: Count Upper And LowerCase vowel and consonants, Even and Odd Number

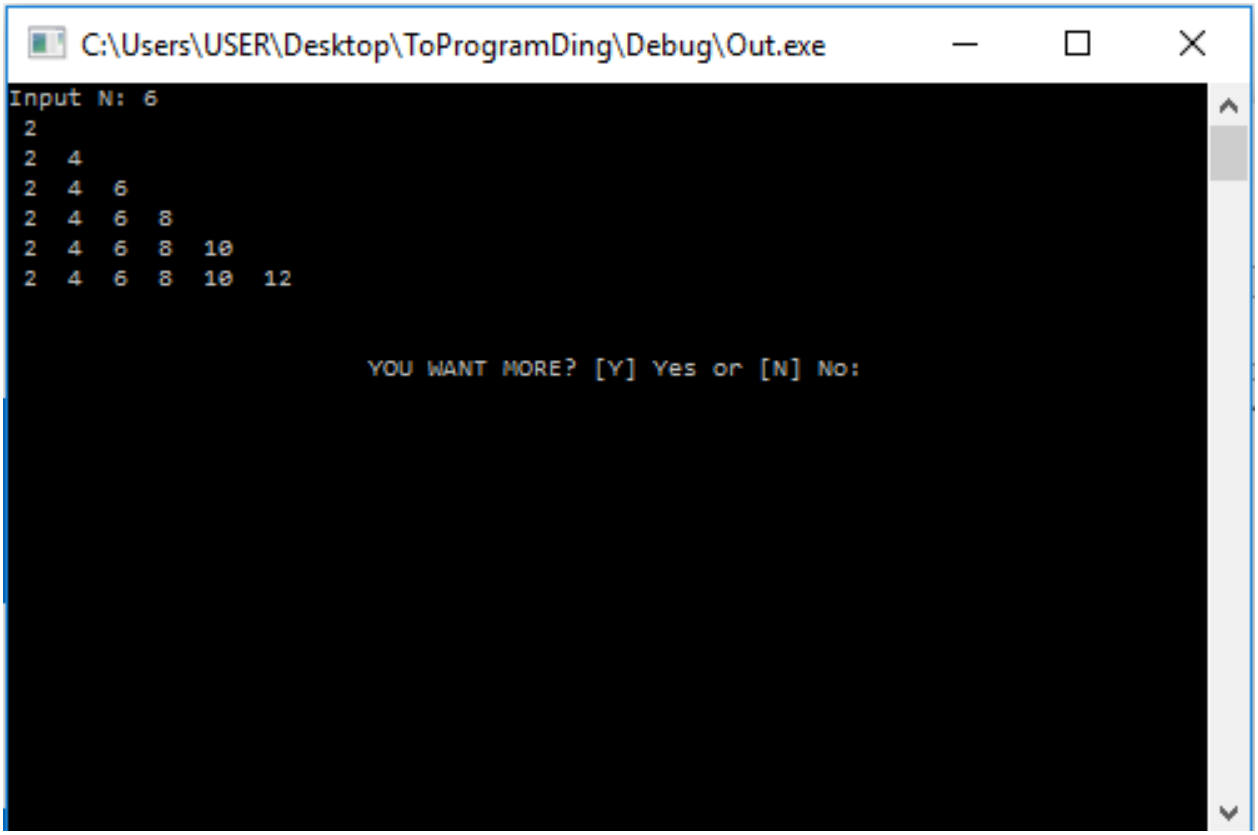
```
PrimaryMission() {
unit totalV = 0;
unit totalC = 0;
unit totalCC = 0;
unit totalVC = 0;
unit totalNumeven = 0;
unit totalNumodd = 0;
unit totalNumzero = 0;
unit totalSpace = 0;
unit totalCh = 0;
unit length, i;
company sentence;

unit choice=0;
go {
commence;
post("\n\t\t\tEnter a String/Word: ");
capture(#sentence);
length = sentence.Extent;

inquire(i = 0; i < length; i++) {
inorder(sentence[i] = = 'a') {
totalV++;
}
otherorder(sentence[i] = = 'e') {
totalV++;
}
otherorder(sentence[i] = = 'i') {
totalV++;
}
otherorder(sentence[i] = = 'o') {
totalV++;
}
otherorder(sentence[i] = = 'u') {
totalV++;
}
otherorder(sentence[i] = = 'A') {
totalVC++;
}
otherorder(sentence[i] = = 'I') {
totalVC++;
}
otherorder(sentence[i] = = 'E') {
totalVC++;
}
otherorder(sentence[i] = = 'O') {
totalVC++;
```

```
}
otherorder(sentence[i] = = 'U') {
totalVC++;
}
otherorder(sentence[i] = = 'b') {
totalC++;
}
otherorder(sentence[i] = = 'c') {
totalC++;
}
otherorder(sentence[i] = = 'd') {
totalC++;
}
otherorder(sentence[i] = = 'f') {
totalC++;
}
otherorder(sentence[i] = = 'g') {
totalC++;
}
otherorder(sentence[i] = = 'h') {
totalC++;
}
otherorder(sentence[i] = = 'j') {
totalC++;
}
otherorder(sentence[i] = = 'k') {
totalC++;
}
otherorder(sentence[i] = = 'l') {
totalC++;
}
otherorder(sentence[i] = = 'm') {
totalC++;
}
otherorder(sentence[i] = = 'n') {
totalC++;
}
otherorder(sentence[i] = = 'p') {
totalC++;
}
otherorder(sentence[i] = = 'q') {
totalC++;
}
otherorder(sentence[i] = = 'r') {
totalC++;
}
otherorder(sentence[i] = = 's') {
totalC++;
}
otherorder(sentence[i] = = 't') {
totalC++;
}
otherorder(sentence[i] = = 'v') {
totalC++;
}
otherorder(sentence[i] = = 'w') {
totalC++;
}
otherorder(sentence[i] = = 'x') {
totalC++;
}
otherorder(sentence[i] = = 'y') {
totalC++;
}
otherorder(sentence[i] = = 'z') {
totalC++;
}
otherorder(sentence[i] = = 'B') {
totalCC++;
}
otherorder(sentence[i] = = 'C') {
totalCC++;
}
otherorder(sentence[i] = = 'D') {
totalCC++;
}
otherorder(sentence[i] = = 'F') {
totalCC++;
}
otherorder(sentence[i] = = 'G') {
totalCC++;
}
otherorder(sentence[i] = = 'H') {
totalCC++;
}
otherorder(sentence[i] = = 'J') {
totalCC++;
}
otherorder(sentence[i] = = 'K') {
totalCC++;
}
otherorder(sentence[i] = = 'L') {
totalCC++;
}
otherorder(sentence[i] = = 'M') {
totalCC++;
}
otherorder(sentence[i] = = 'N') {
```

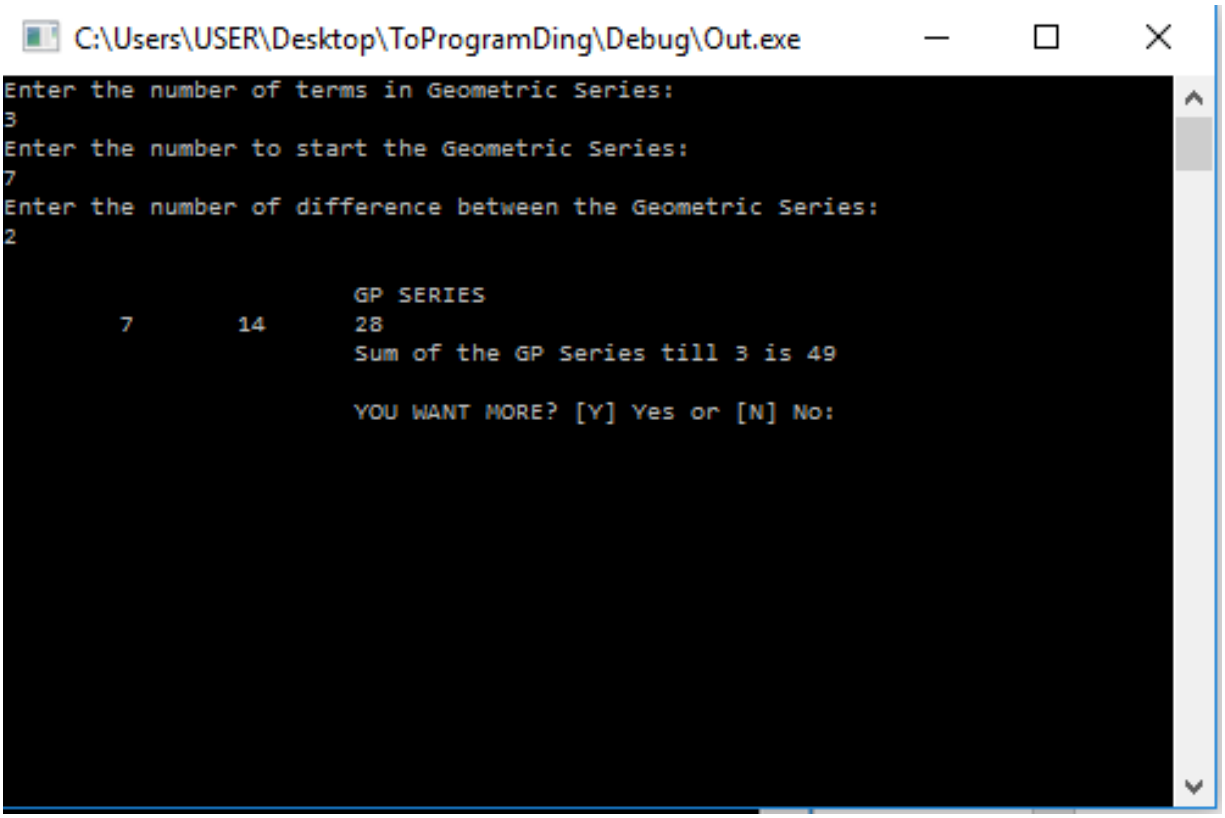
```
totalCC++;
}
otherorder(sentence[i] = = 'P') {
totalCC++;
}
otherorder(sentence[i] = = 'Q') {
totalCC++;
}
otherorder(sentence[i] = = 'R') {
totalCC++;
}
otherorder(sentence[i] = = 'S') {
totalCC++;
}
otherorder(sentence[i] = = 'T') {
totalCC++;
}
otherorder(sentence[i] = = 'V') {
totalCC++;
}
otherorder(sentence[i] = = 'W') {
totalCC++;
}
otherorder(sentence[i] = = 'X') {
totalCC++;
}
otherorder(sentence[i] = = 'Y') {
totalCC++;
}
otherorder(sentence[i] = = 'Z') {
totalCC++;
}
otherorder(sentence[i] = = ' ') {
totalSpace++;
}
otherorder(sentence[i] = = '1') {
totalNumodd++;
}
otherorder(sentence[i] = = '2') {
totalNumeven++;
}
otherorder(sentence[i] = = '3') {
totalNumodd++;
}
otherorder(sentence[i] = = '4') {
totalNumeven++;
}
otherorder(sentence[i] = = '5') {
totalNumodd++;
}
otherorder(sentence[i] = = '6') {
totalNumeven++;
}
otherorder(sentence[i] = = '7') {
totalNumodd++;
}
otherorder(sentence[i] = = '8') {
totalNumeven++;
}
otherorder(sentence[i] = = '9') {
totalNumodd++;
}
otherorder(sentence[i] = = '0') {
totalNumzero++;
}
order {
totalCh++;
}
}

post("\n\t\t\tYour total number of
Lowercase vowels is: " + totalV);
post("\n\t\t\tYour total number of
Lowercase consonants is: " + totalC);
post("\n\t\t\tYour total number of
Uppercase vowels is: " + totalVC);
post("\n\t\t\tYour total number of
Uppercase consonants is: " + totalCC);
post("\n\t\t\tYour total number of Even
Number is: " + totalNumeven);
post("\n\t\t\tYour total number of Odd
Number is: " + totalNumodd);
post("\n\t\t\tYour total number of Zero
is: " + totalNumzero);
post("\n\t\t\tYour total number of
Spaces is: " + totalSpace);
post("\n\t\t\tYour total number of
other Character is: " + totalCh);

go {
company ch;
post("\n\n\t\t\tTRY AGAIN? [Y] Yes or
[N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
totalV = 0;
totalVC = 0;
totalC = 0;
```

```
totalCC = 0;
totalNumeven =0;
totalNumodd = 0;
totalSpace = 0;
totalCh =0;
length=0;
i=0;
sentence= " ";
choice = 1;
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}

} phase(choice = = 3);

} phase(choice != 0);
post("\n\t\t\tGOODBYE!!");
} deploy();
```

Source Code: Even Triangle



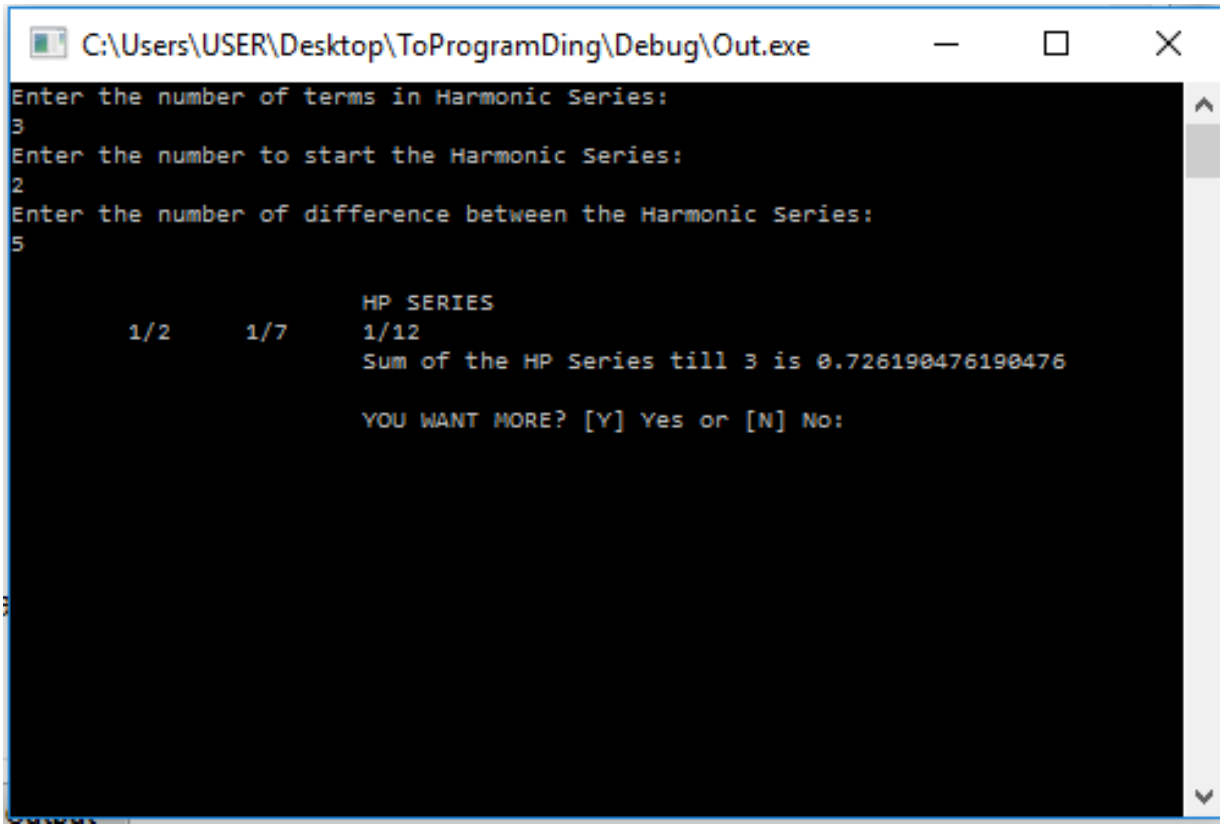

```
PrimaryMission() {
unit i, j,n, num= 2,choice;
go {
commence;
post("Input N: ");
capture(#n);
phase(n<0) {
post("Please Enter a Positive Integer:
\n");
capture(#n);
}
inquire(i=0;i<n;i++) {
num=2;
inquire(j=0;j<=i;j++) {
post(" " +num+ " ");
num = num + 2;
}
post("\n");
}

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
i=0;
j=0;
n=0;
num=2;



}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
```




```
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

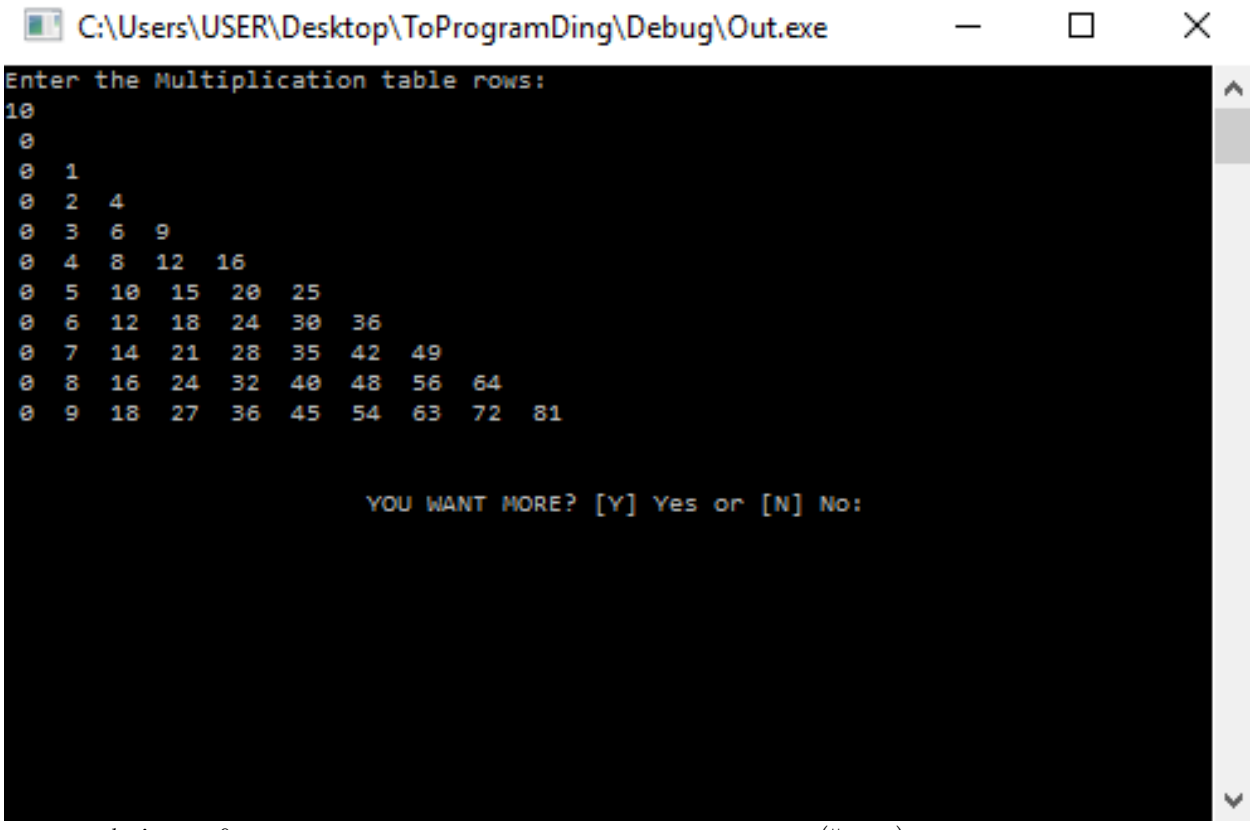
Source Code: Geometric Series

```
unit value;
PrimaryMission() {
unit first,ratio,terms,sum=0, i,choice;
go {
commence;
post("Enter the number of terms in
Geometric Series: \n");
capture(#terms);
post("Enter the number to start the
Geometric Series: \n");
capture(#first);
post("Enter the number of ratio between
the Geometric Series: \n");
capture(#ratio);
value=first;
post("\n\t\t\tGP SERIES \n ");

inquire(i=0;i<terms;i++) {
post("\t" + value + " " );
sum= sum+value;
value = value * ratio;
}
post("\n\t\t\tSum of the GP Series till
" +terms+ " is " +sum+ " ");

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
first =0;terms=0;ratio=0;value=0;sum=0;
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}order {
post("\n\t\t\tError Input!");
choice = 3;
}} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```



Source Code: Harmonic Series

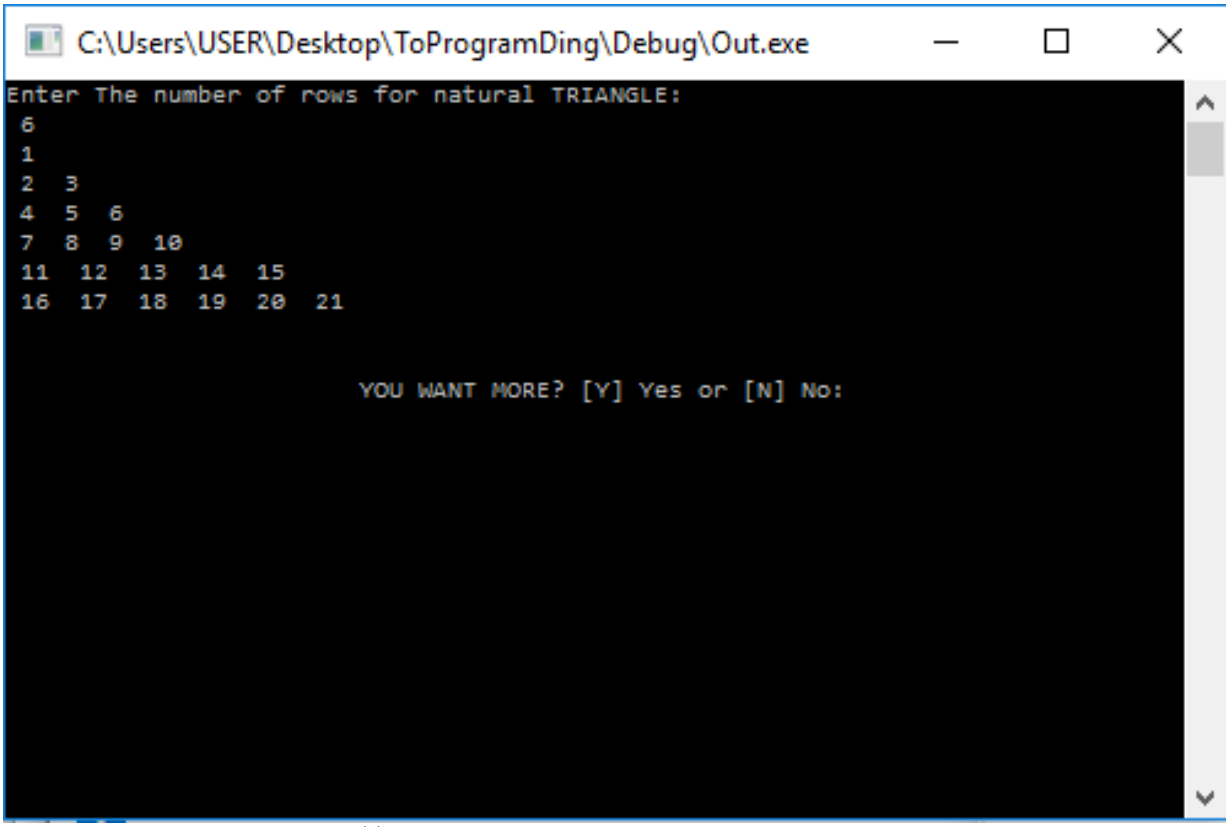
```
digit denominator;
digit sum= 0;
PrimaryMission() {
unit first,diff,terms, i,choice;
go {
commence;
post("Enter the number of terms in
Harmonic Series: \n");
capture(#terms);
phase(terms<0) {
post("Please Enter a Positive number of
terms :\n ");
capture(#terms);
}
post("Enter the number to start the
Harmonic Series: \n");
capture(#first);
post("Enter the number of difference
between the Harmonic Series: \n");
capture(#diff);
denominator=first;
post("\n\t\t\tHP SERIES \n ");

inquire(i=0;i<terms;i++) {
post("\t1/" + denominator + " " );
sum= sum + 1/denominator;
denominator = denominator + diff;
}
post("\n\t\t\tSum of the HP Series till
" +terms+ " is " +sum+ " ");

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
first =0;
terms=0;
diff=0;
denominator=0;
sum=0;



}
otherorder((ch = = "N") || (ch = =
"n")) {
```




```
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

Source Code: Multiplication Triangle

```
PrimaryMission() {
unit i,j,rows,var;
unit count=1;
unit choice;
go {
commence;
post("Enter the Multiplication table
rows: \n");
capture(#rows);
phase(rows<0) {
post("Please Enter a Positive number of
ROWS!! :\n");
capture(#rows);
}
inquire(i=0;i<rows;i++) {
inquire(j=0;j<=i;j++) {
var = i*j;
post(" " + var + " ");
}
post("\n");
}

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
i=0;
j=0;
rows=0;
var=0;
count=1;
}
```



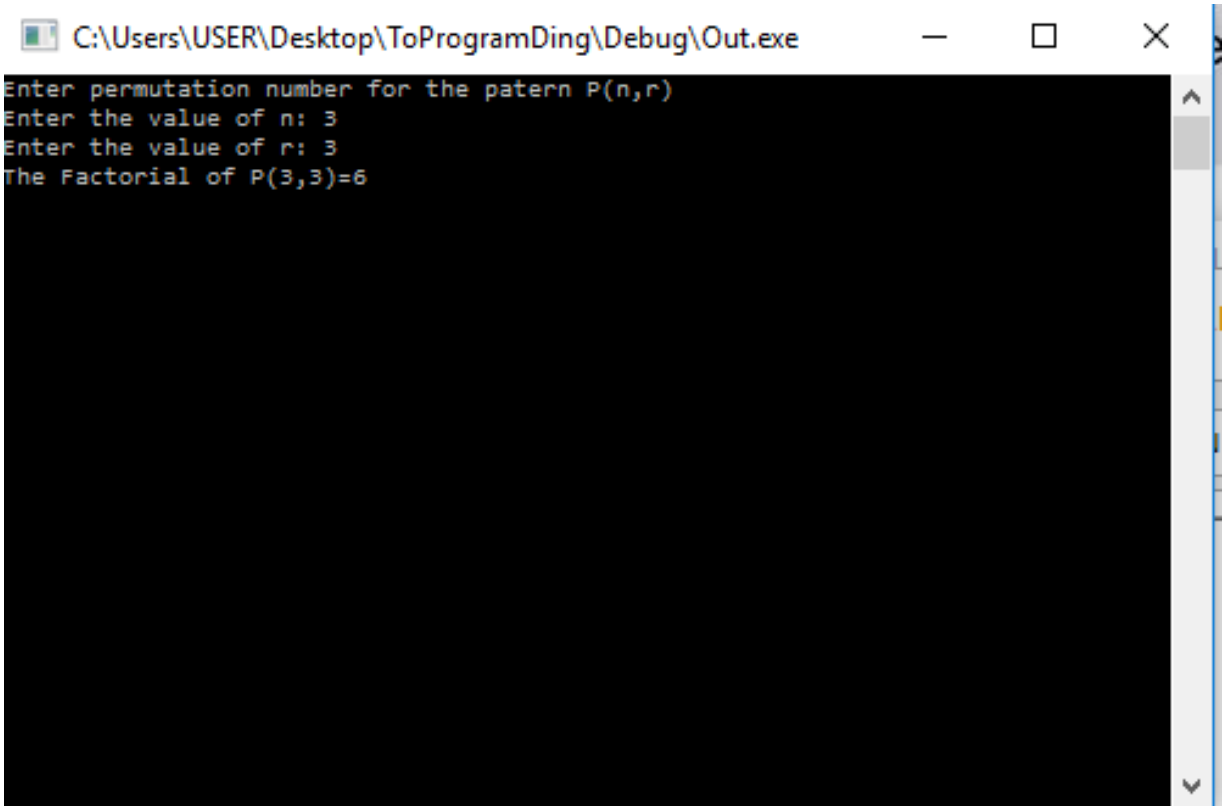

```
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

Source Code: Natural Number Triangle

```
PrimaryMission() {
unit i, j, rows,choice;
```

```
unit count =1;
```

```
inorder((ch = = "Y") || (ch = = "y")) {
```




```
go {
commence;

post("Enter The number of rows for
natural TRIANGLE: \n ");
capture(#rows);
phase(rows<0) {
post("Please Enter a Positive number of
ROWS!! :\n");
capture(#rows);
}

inquire(i=1;i<=rows;i++) {
inquire(j=1;j<=i;j++) {
post(" " +count+ " ");
count++;
}
post("\n");
}

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
choice = 1;
i=0;
j=0;
rows=0;
count=1;


}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

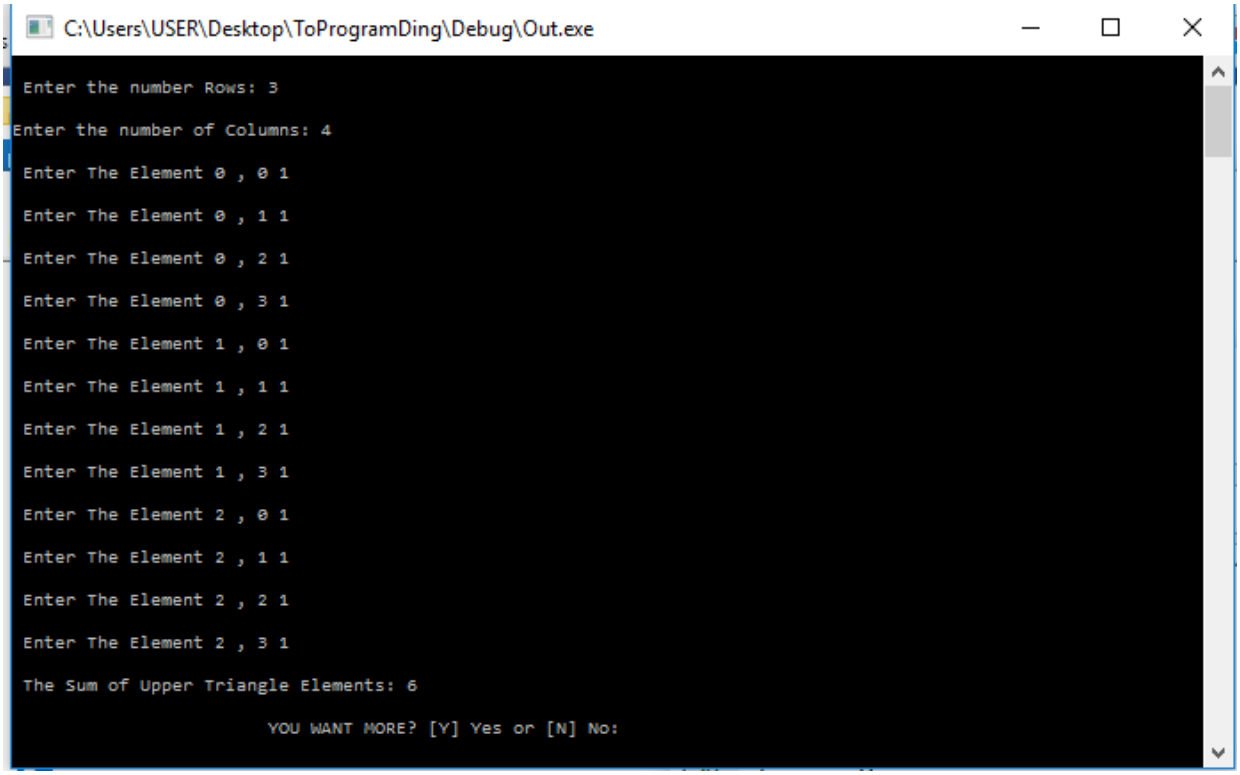
Source Code: Permutation

```
unit fact(unit c) {
```

```
unit f=1;
phase(c>0) {
f= f*c;
c--;
} backup(f);
}
PrimaryMission() {
unit p, r, n,v;
post("Enter permutation number for the
patern P(n,r) \n");
post("Enter the value of n: ");
capture(#n);
post("Enter the value of r: ");
capture(#r);
p= fact(n) / fact(n-r);
post("The Factorial of P(" +n+ "," +r+
")  =" +p+ " ");
} deploy();
```

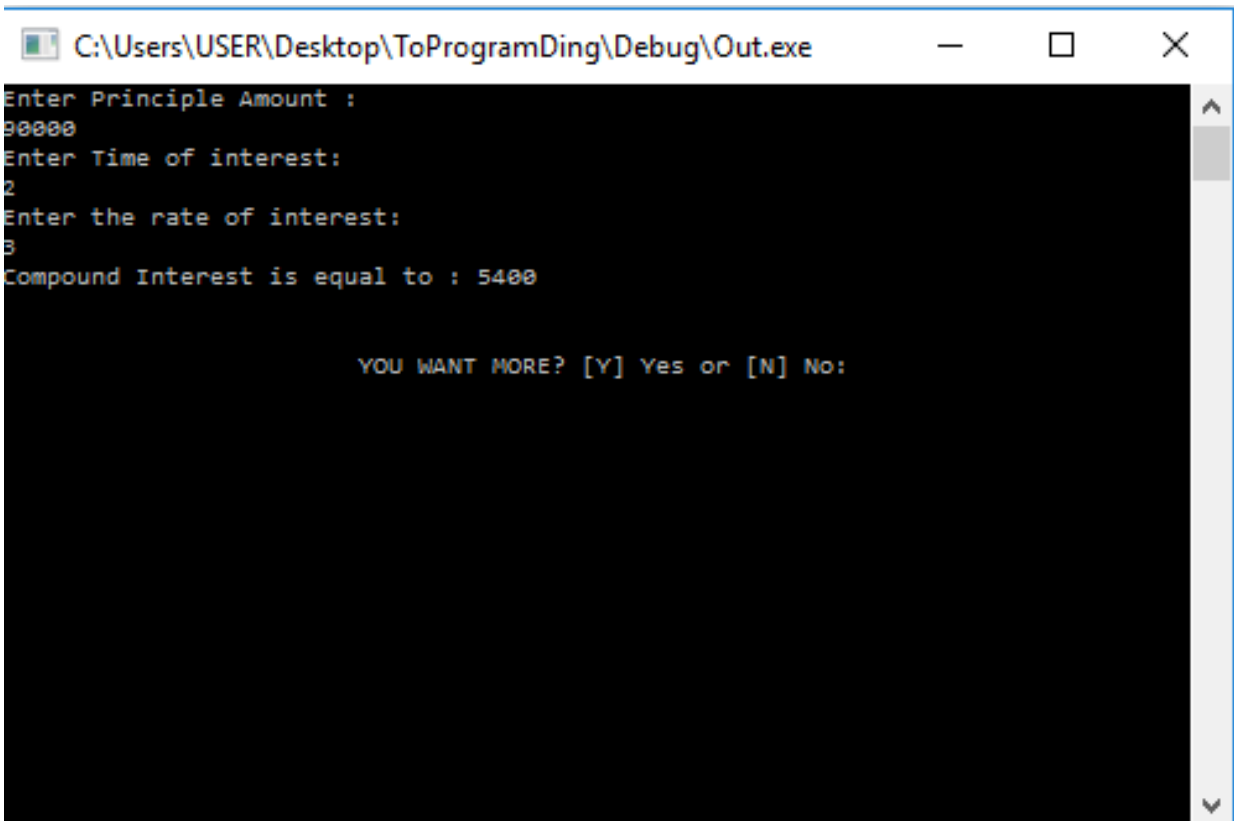
Source Code: Sum of Upper Triangle of a Matrix

```
unit a[10][10];
unit j;
PrimaryMission() {
unit i, sum, rows, columns,choice;

go {
commence;
post("\n Enter the number Rows: ");
capture(#rows);

post("\nEnter the number of Columns:
");
capture(#columns);

inquire(i=0;i<rows;i++) {
inquire(j=0;j<columns;j++) {
post("\n Enter The Element " + i + " ,
" +j+ " ");
capture(#a[i][j]);
}
}
sum =0;
inquire(i=0;i<rows;i++) {
inquire(j=0;j<columns;j++) {
```



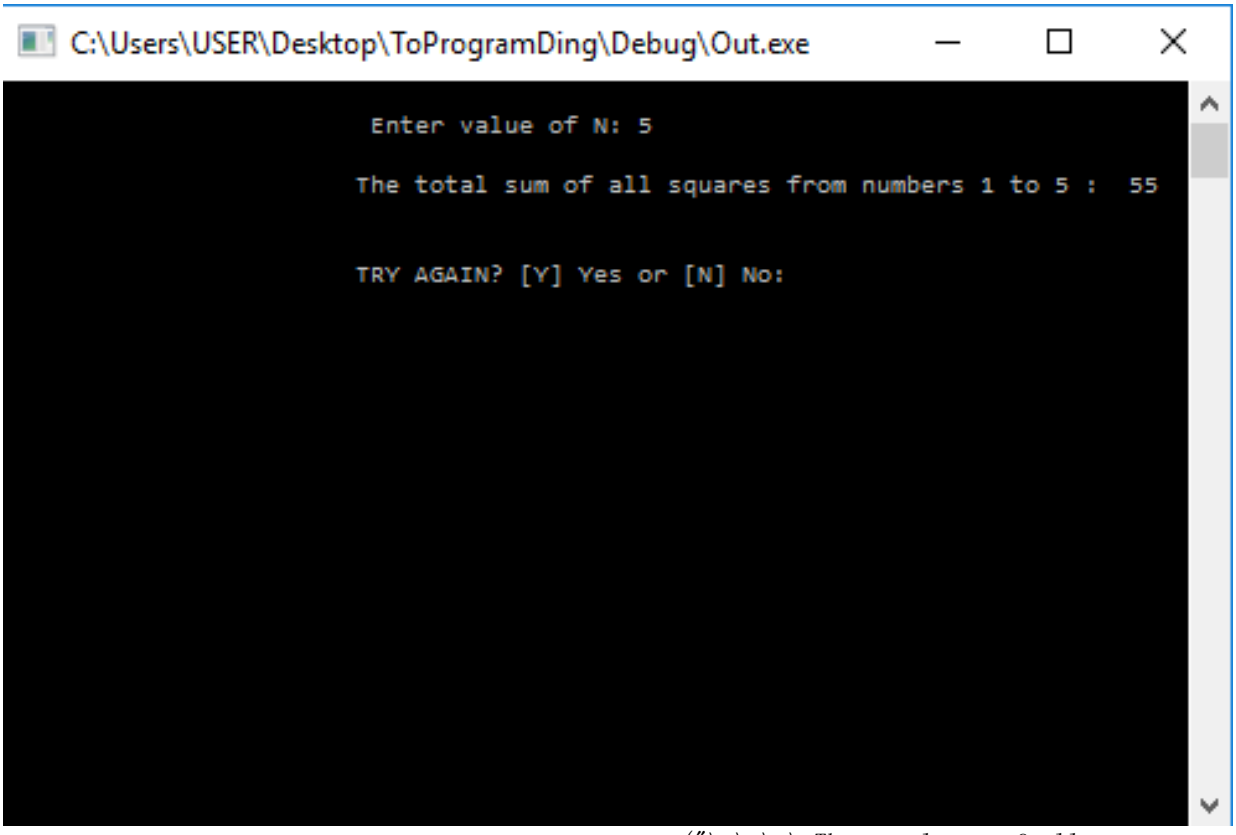

```
inorder(i<j) {
```

```
sum =
sum +
```




```
a[i][j];
}
}
}
post("\n The Sum of Upper Triangle
Elements: " +sum+ " ");

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
i=0;
j=0;
rows=0;
sum=0;
columns=0;



}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();

Source Code: Compound Interest
PrimaryMission() {
digit principle, rate,
C,C2,rated,timed;
unit time,choice;

go {
commence;
post("Enter Principle Amount : \n");
capture(#principle);

post("Enter Time of interest: \n");
capture(#time);
```

```
post("Enter the rate of interest: \n");
capture(#rate);
rated= 1 + (rate/100);
timed= rated*time;
C = principle*rated;
C2 = (C-principle) * time;
post("Compound Interest is equal to : "
+C2+ "\n");

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
principle =0;
rated=0;
C=0;
C2=0;
time=0;
timed=0;
rate=0;

}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

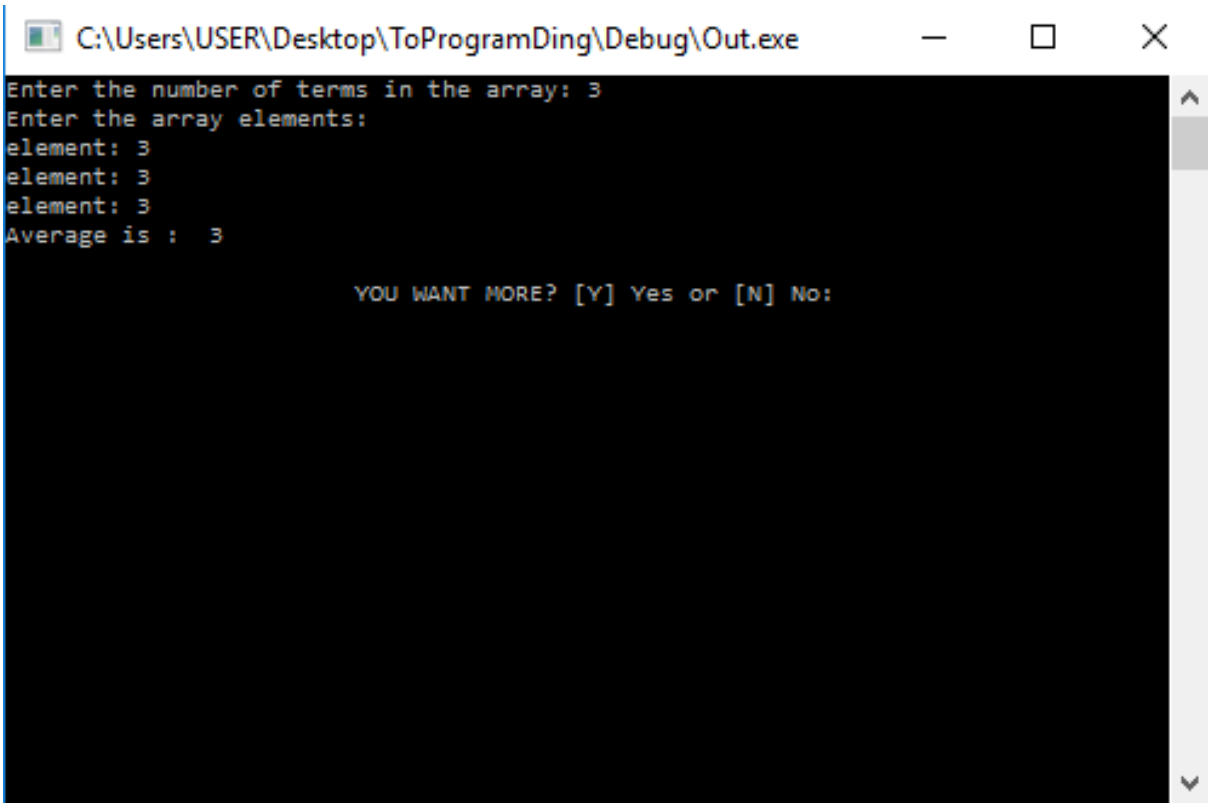


Source Code: Sum of Squares from 1-n

```
PrimaryMission() {
unit a, no, var,choice, square = 0,
sum=0;
go {
commence;
post("\n\t\t\t Enter value of N: " );
capture(#no);
inorder(no>0) {
inquire(a=1;a<=no;a++) {
square=a*a;
sum=sum+square;
}
post("\n\t\t\tThe total sum of all
squares from numbers 1 to " + no +  " :
" + sum + " \n" );
}
order {
inquire(a=1;a>=no;a--) {
square=a*a;
sum=sum+square;
}
post("\n\t\t\tThe total sum of all
squares from numbers 1 to " + no +  " :
" + sum + " \n" );

}

go {
company ch;
post("\n\n\t\t\tTRY AGAIN? [Y] Yes or
[N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
no=0;
var=0;
choice = 1;
sum=0;
square=0;
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
```

```
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\tPAALAM HANGGANG SA MULING
PAGIINPUT!!");
} deploy();
```

```
post("Enter the array elements: \n");
inquire(i=0;i<m;i++) {
post("element: ");
capture(#a[i]);
}
```

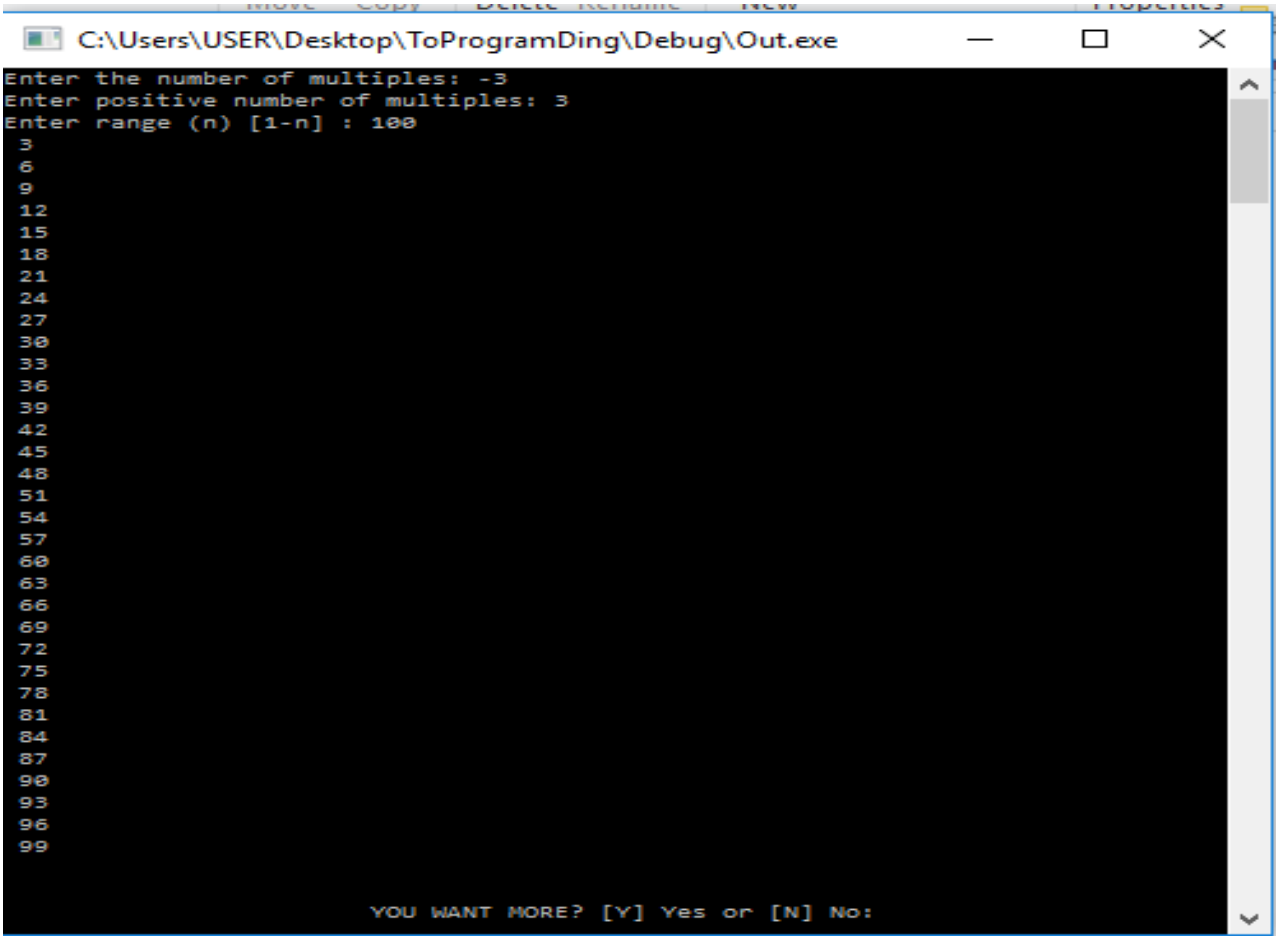


Souce Code: Average of set of elements

```
unit a[100];

PrimaryMission() {
unit m,i,choice;
unit sum= 0;
unit avg= 0;

go {
commence;
post("Enter the number of terms in the
array: ");
capture(#m);
phase(m<0) {
post("Enter positive number of  terms
in the array: ");
capture(#m);
}
a[m] = a[m];
```

```
inquire(i=0;i<m;i++) {
sum = sum + a[i];
}
avg= sum/m;
post("Average is :  " +avg + " ");

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
i=0;
m=0;
avg=0;
sum=0;
```

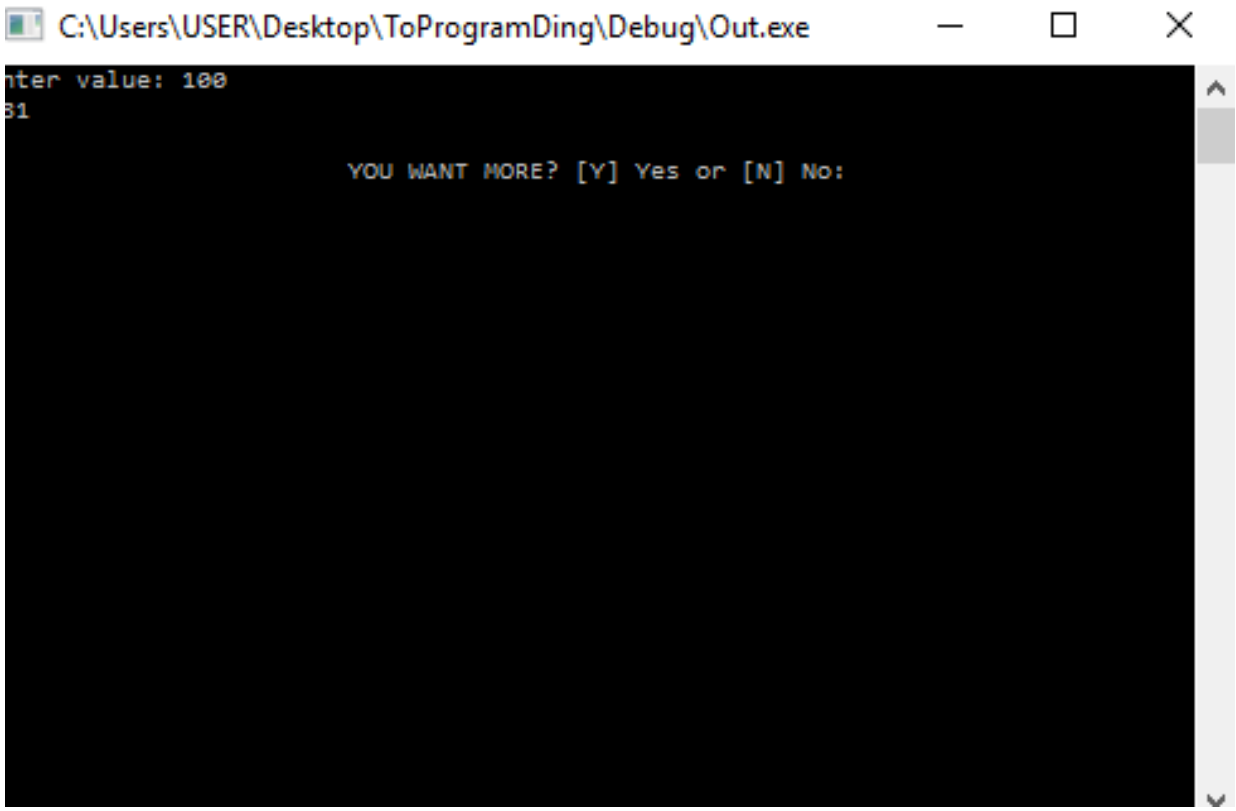
```
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```



Source Code: Multiples of y in given range

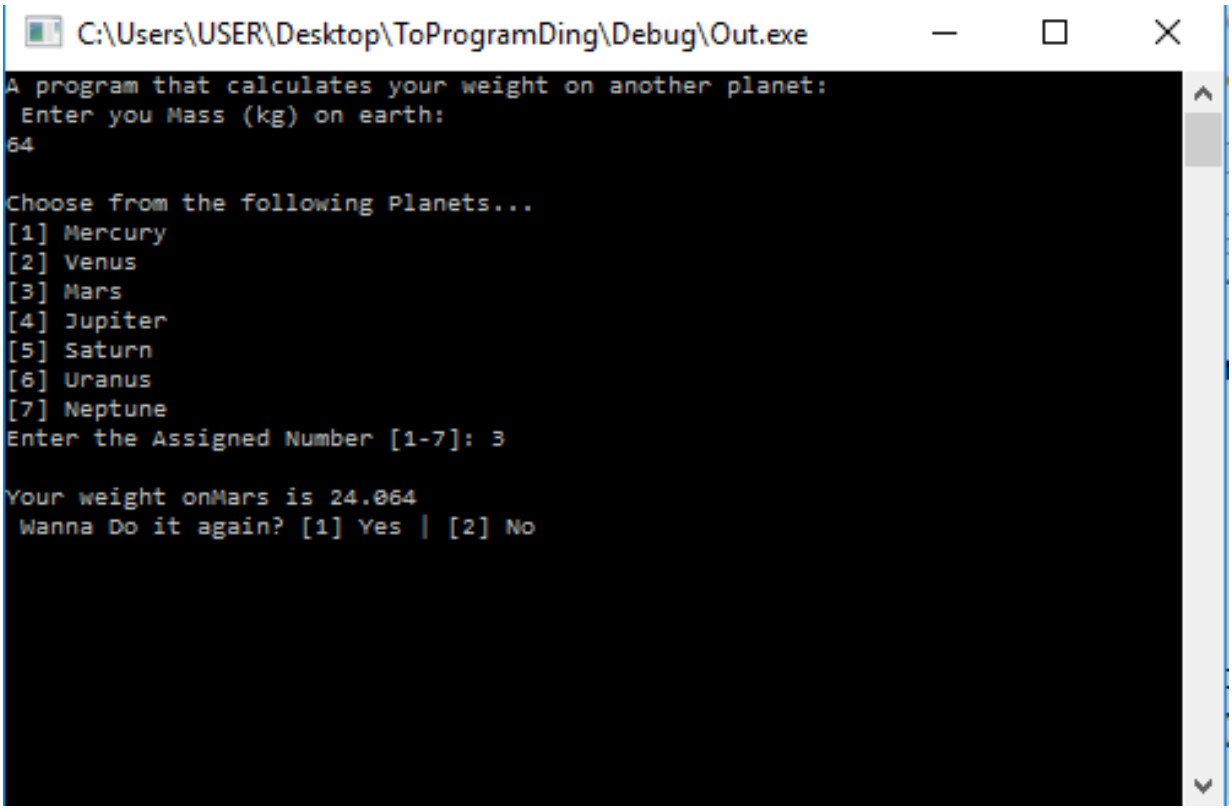
```
PrimaryMission() {
unit a,i,y,z,choice;

go {
commence;
post("Enter the number of multiples: ");
capture(#y);
phase(y<0) {
post("Enter positive number of multiples: ");
capture(#y);
}
post("Enter range (n) [1-n] : ");
capture(#z);
inorder(z>=1) {
inquire(i=1;i<z;i++) {
a = i % y;
inorder(a = = 0) {
post(" " +i+ "\n");
}
}
}
order {
post("Input value is too low..... : ");
}

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
i=0;
a=0;
y=0;
z=0;
```

```
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```



Source Code: Number that don't contains 3 within the range

```
unit count(unit n) {
unit m,msd,a,t,b,c,e,f,g,d;
unit po=1;
inorder(n<3) {
backup(n);
}
inorder((n>=3) & (n<10)) {
n= n-1;
backup(n);
}
m= n/po;
phase(m>9) {
po = po * 10;
m= n/po;
}

msd = n/po;
inorder(msd !=3) {
a = count(msd);
t= po-1;
b=count(t);
e= n%po;
f=count(e);
c= a * b + a+ f;
backup(c);
}
order {
d = msd* po -1;
g = count(d);
backup(g);
}
}

PrimaryMission() {
unit m,msd,a,t,b,c,e,f,g,d;
unit ans,val,choice;
unit po=1;
go {
commence;
post("Enter value: ");
capture(#val);
phase(val<0) {
post("Enter positive value: ");
capture(#val);
}
ans = count(val);
post(" " +ans+ " ");
```

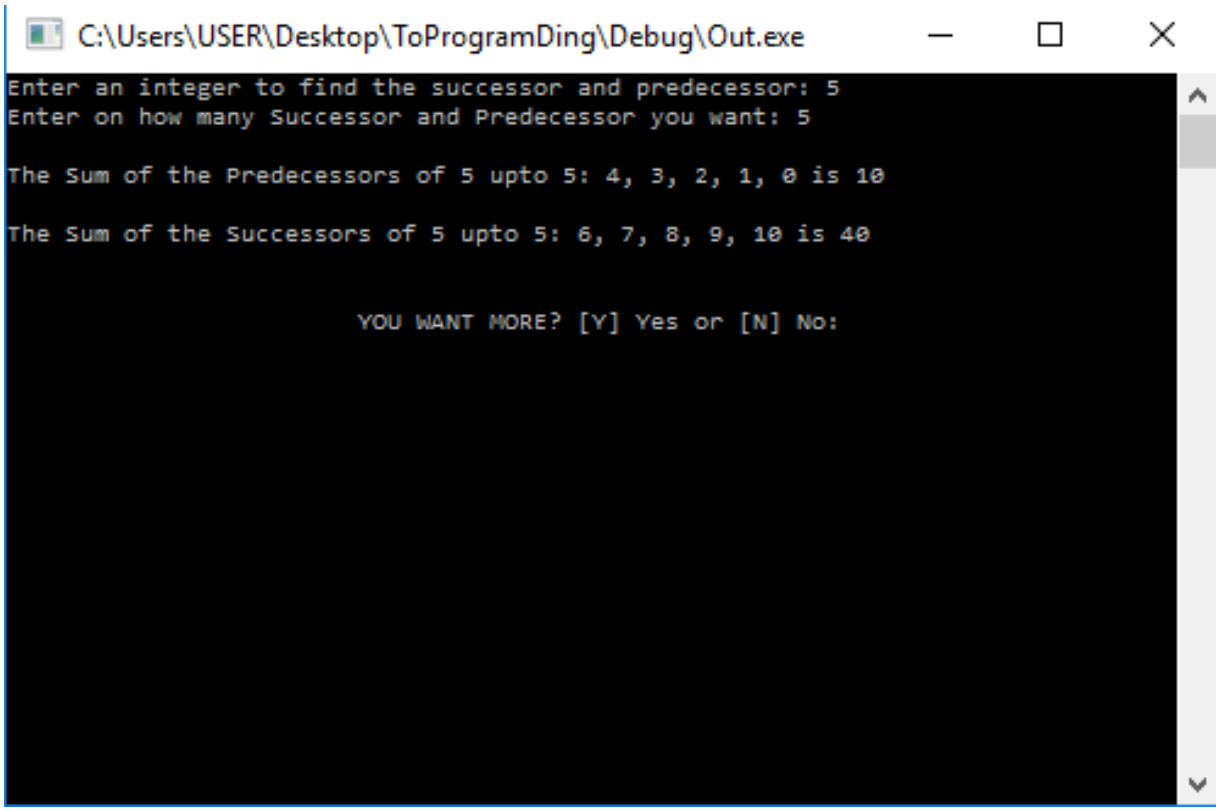



```
go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
m=0;
msd=0;
ans=0;
val=0;
a=0;
b=0;
c=0;
d=0;
g=0;
f=0;
t=0;
po=0;

}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

Source Code: Gravity converter

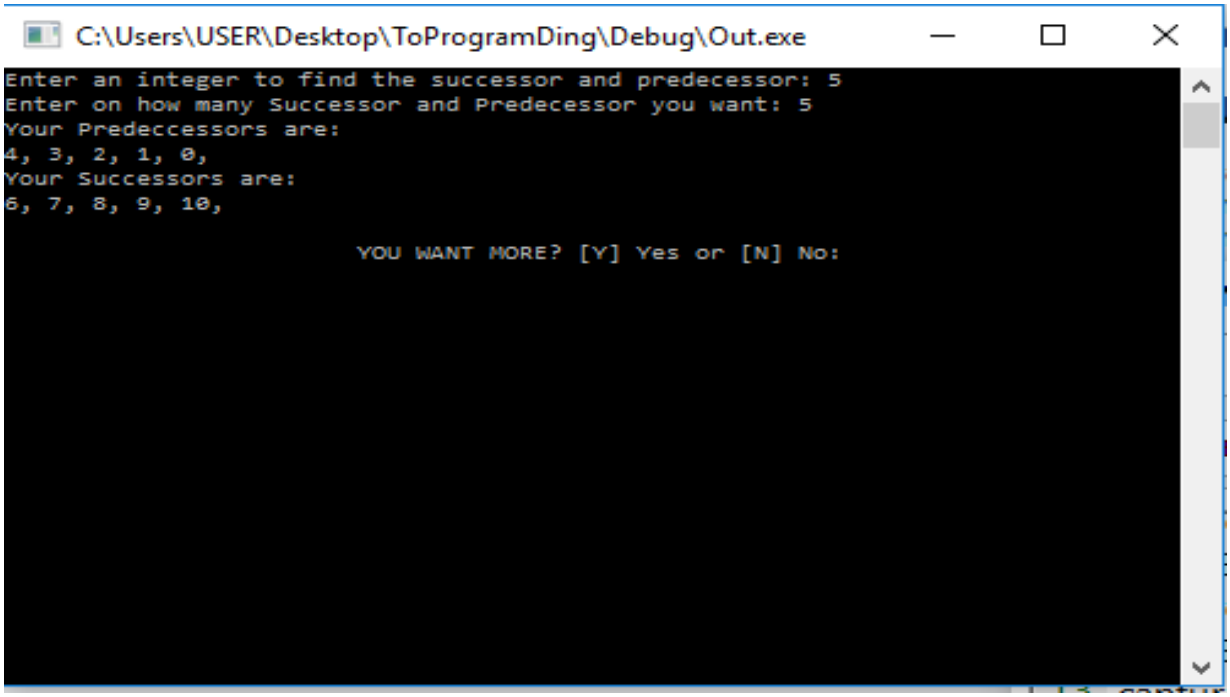
```
PrimaryMission() {
unit earth,nplanet,checker,ans;
unit again=0;
digit weight=0;
company splanet = " ";

go {
post("A program that calculates your
weight on another planet:\n ");
post("Enter you Mass (kg) on earth:
\n");
go {
capture(#earth);
```

```
inorder(earth<=0) {
post("Please dont input zero or
negative number, Input again: ");
checker=1;
}
order {
checker =0;
}
} phase(checker= = 1);
post("\nChoose from the following
Planets... \n[1] Mercury\n[2]
Venus\n[3] Mars\n[4] Jupiter\n[5]
Saturn\n[6] Uranus\n[7] Neptune\n");
post("Enter the Assigned Number [1-7]:
");
go {
capture(#nplanet);
inorder(nplanet = = 1) {
weight = earth * 0.38;
splanet = "Mercury";
checker =0;
}
otherorder(nplanet = = 2) {
weight = earth * 0.38;
splanet = "Mercury";
checker =0;
}
otherorder(nplanet = = 2) {
weight = earth * 0.904;
splanet = "Venus";
checker =0;
}
otherorder(nplanet = = 3) {
weight = earth * 0.376;
splanet = "Mars";
checker =0;
}
otherorder(nplanet = = 4) {
weight = earth * 2.53;
splanet = "Jupiter";
checker =0;
}
otherorder(nplanet = = 5) {
weight = earth * 1.07;
splanet = "Saturn";
checker =0;
}
otherorder(nplanet = = 6) {
weight = earth * 0.89;
splanet = "Uranus";
checker =0;
}
otherorder(nplanet = = 2) {
weight = earth * 1.14;
splanet = "Neptune";
checker =0;
}
order {
post("Invalid Input!! - Please Input
again: ");
checker=1;
}
} phase(checker==1);
post("\nYour weight on" +splanet+ " is
" +weight+ " " );
post("\n Wanna Do it again? [1] Yes |
[2] No ");
go {
capture(#ans);
inorder(ans= =1) {
commence;
checker = 0;
again=1;
}
otherorder(ans= =2) {
post("Thanks You ");
checker = 0;
again =0;
}
order {
post("Invalid Input !!! Please Input
again: ");
checker = 1;
}
} phase(checker= =1);
} phase(again= =1);
} deploy();
```



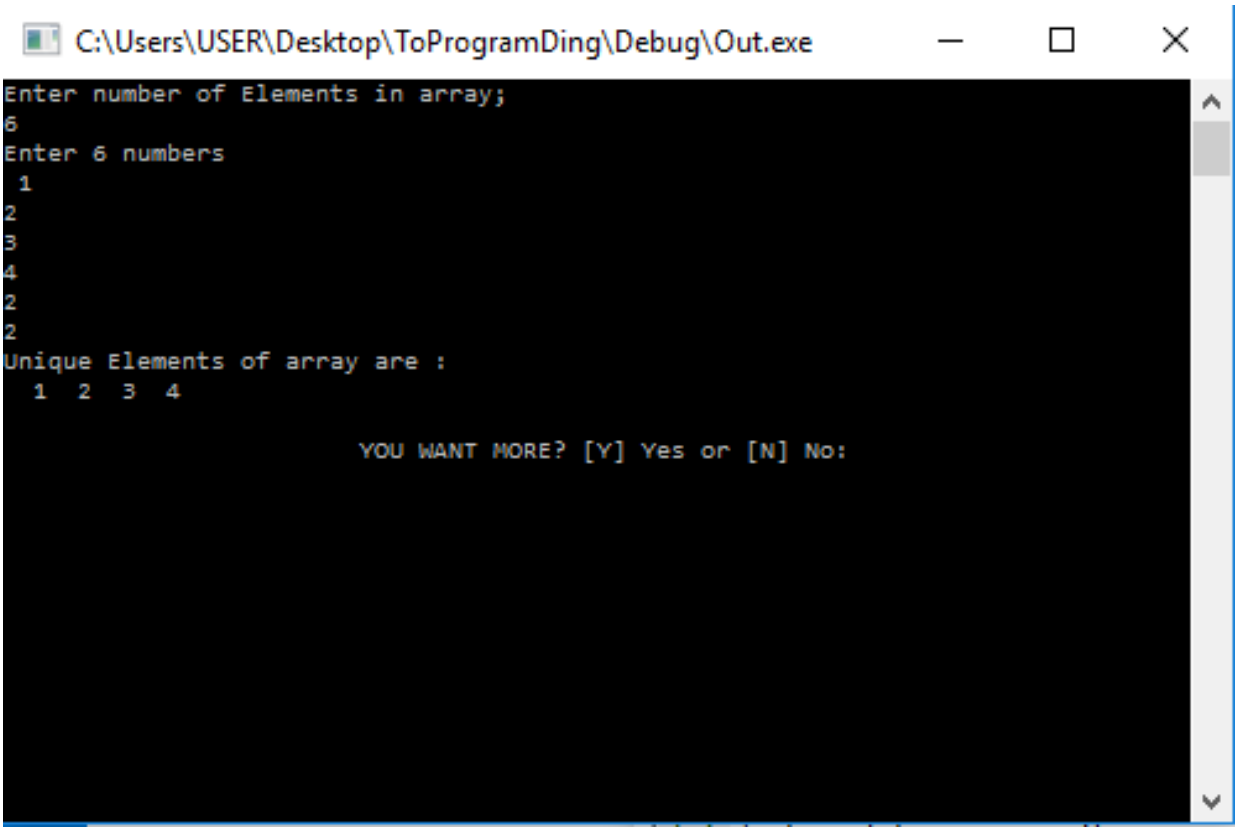
Source Code Predecessor and Successor SUM

```
unit x,i,n;
digit y,w;
PrimaryMission() {
digit sumy=0;
digit sumw=0;
unit choice;

go {
commence;
post("Enter an integer to find the
successor and predecessor: ");
capture(#x);
post("Enter on how many Successor and
Predecessor you want: ");
capture(#n);
inorder((x>0) || (x<0)) {
post("\nThe Sum of the Predecessors of
" +x+ " upto " + n  + ": ");
inquire(i=1;i<=n;i++) {
w = x-i;
post(w + ", " );
sumw = sumw + w;
}
post("\b\b is " + sumw + "\n");

post("\nThe Sum of the Successors of "
+x+ " upto " + n  + ": ");
inquire(i=1;i<=n;i++) {
y = x+i;
post(y + ", " );
sumy = sumy + y;
}
post("\b\b is " + sumy + "\n");

}

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
n=0;
sumy=0;
sumw=0;
x=0;
```



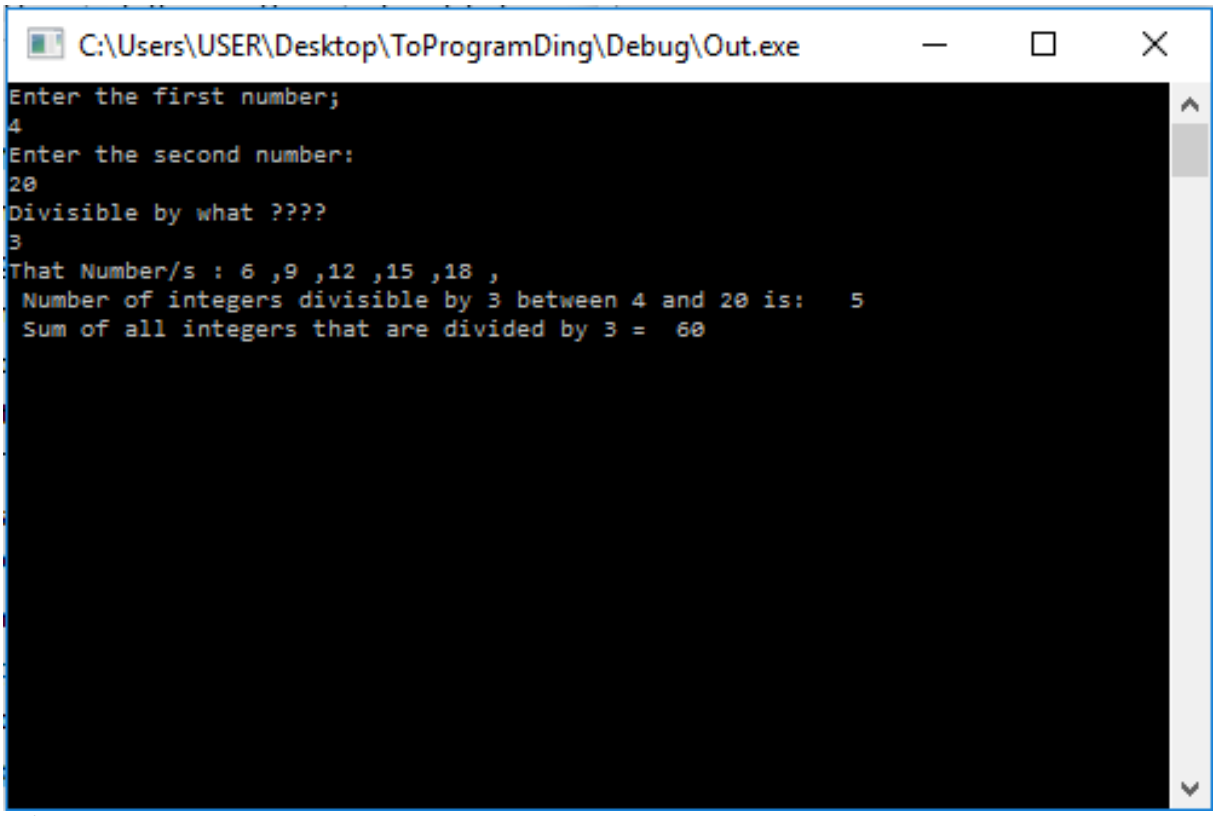
Source Code: Predecessors and Successors

```
unit x,i,n;
digit y,w;
PrimaryMission() {
digit sumy=0;
digit sumw=0;
unit choice;

go {
commence;
post("Enter an integer to find the successor and predecessor: ");
capture(#x);
post("Enter on how many Successor and Predecessor you want: ");
capture(#n);
inorder((x>0) || (x<0)) {
post("Your Predeccessors are: \n");
inquire(i=1;i<=n;i++) {
w = x-i;
post(w + ", " );
}
post("\nYour Successors are: \n");
inquire(i=1;i<=n;i++) {
y = x+i;
post(y + ", " );
}
}
go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
n=0;
x=0;
i=0;
w=0;
y=0;

}
otherorder((ch = = "N") || (ch = = "n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po !!");
} deploy();
```



Source Code: Unique Elements

```
unit array[100];

PrimaryMission() {
unit size, i, j, choice;
go {
commence;
post("Enter number of Elements in
array; \n");
capture(#size);
phase(size<0) {
post("Please ENTER A NUMBER Greater
than ZERO: ");
capture(#size);
}
post("Enter " +size+ " numbers \n ");

inquire(i=0;i<size;i++) {
capture(#array[i]);
}

post("Unique Elements of array are : \n
");
inquire(i=0;i<size;i++) {
inquire(j=0;j<i;j++) {
inorder(array[i] = = array[j]) {
inorder(i = = j) {
post(" " + array[i] + " ");
}
abort();
}
}
inorder(i = = j) {
post(" " + array[i] + " ");
}
}

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
i=0;
j=0;
size=0;
```

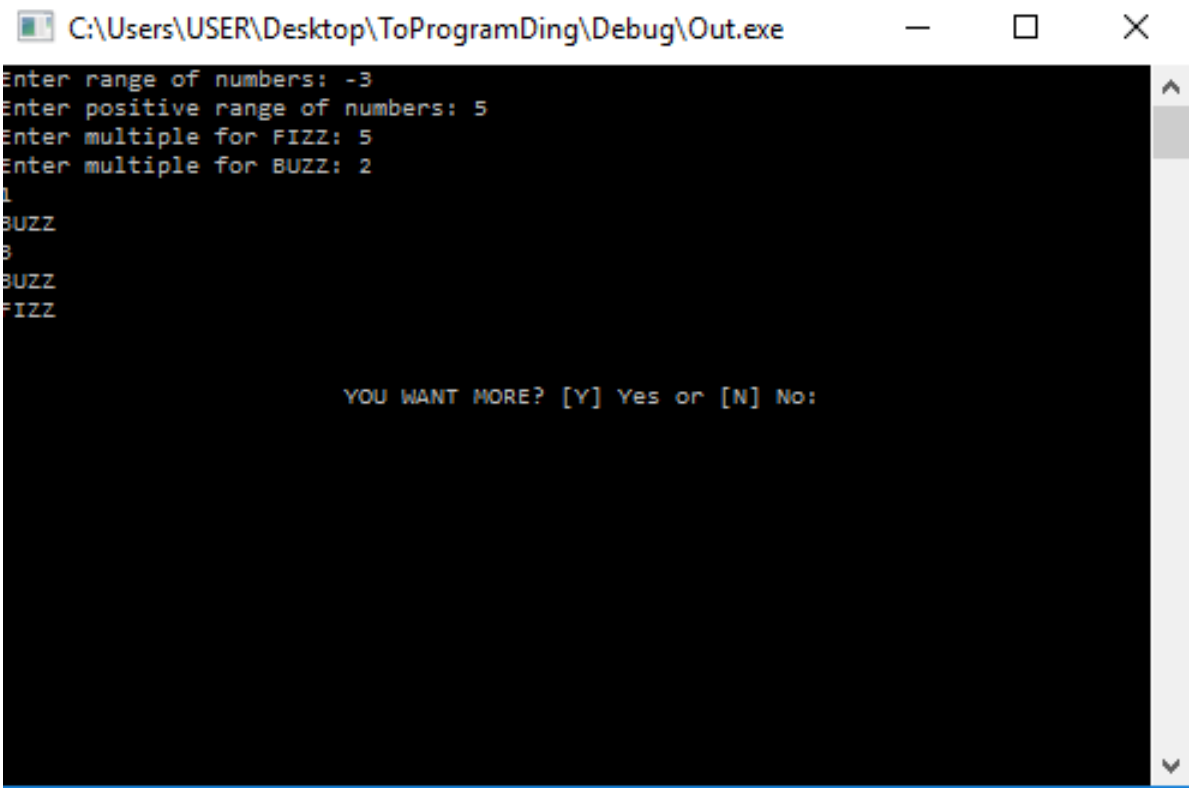



```
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

Source Code: Number divisible by n within the range and its sum

```
PrimaryMission() {
unit i, num1,num2, div, res,choice;
unit count=0;
unit sum=0;
```

```
go {
commence;
post("Enter the first number; \n");
capture(#num1);
post("Enter the second number: \n");
capture(#num2);
post("Divisible by what ???? \n");
capture(#div);

inorder((num1>0) & (num2>0)) {
post("That Number/s : ");
inquire(i=num1;i<num2;i++) {
res = i%div;
inorder(res = = 0) {
post(i + " ,");
count = count +1;
sum = sum+i;

}
}
post("\n Number of integers divisible
by " + div + " between " +num1+ " and "
+num2+ " is:   "  +count+ " ");
post("\n Sum of all integers that are
divided by " + div+ " =  " +sum+ " ");
}
order {
post("Invalid Value ");
}

go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
res=0;
div=0;
i=0;
num1=0;
num2=0;
count=0;
sum=0;
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```



Source Code: BUZZFIZZ

```
PrimaryMission() {
unit
num,i,buzz,fizz,bufsum,fisum,buss,busum
;
unit choice;
go {
commence;
post("Enter range of numbers: ");
capture(#num);
phase(num<0) {
post("Enter positive range of numbers:
");
capture(#num);
}
post("Enter multiple for FIZZ: ");
capture(#fizz);
phase(fizz<0) {
post("Enter positive range of numbers:
");
capture(#fizz);
}
post("Enter multiple for BUZZ: ");
capture(#buzz);
phase(buzz<0) {
post("Enter positive range of numbers:
");
capture(#buzz);
}
inquire(i=1;i<=num;i++) {
buss=fizz * buzz;
bufsum=i%buss;
fisum = i%fizz;
busum = i%buzz;
inorder(bufsum= = 0) {
post("BUZZFIZZ\n");
}
otherorder(fisum = =0) {
post("FIZZ\n");
}
otherorder(busum= = 0) {
post("BUZZ\n");
}
order {
post(i + "\n");
}
}

go {
```

```
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
i=0;
num=0;
buzz=0;
fizz=0;
fisum=0;
buss=0;
busum=0;
bufsum=0;
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```

Source code:
Circle without floating point arithmetic

```
unit r;

PrimaryMission() {
unit a,c,i,j,n,x,y;
unit choice;

go {
commence;
post("Enter value of N: ");
capture(#n);
inorder(n>0) {
n = 2 * r +1;
inquire(i=0;i<n;i++) {
inquire(j=0;j<n;j++) {
x =i -r;
y = j-r;
a = x * x + y * y;
c = r*r+1;
inorder(a<=c) {
post(".");
}
order {
post("* ");
}
post("");
}
post("\n");
}
}
order {
post("INVALID!!!! " );
} go {
company ch;
post("\n\n\t\t\tYOU WANT MORE? [Y] Yes
or [N] No: ");
capture(#ch);
inorder((ch = = "Y") || (ch = = "y")) {
choice = 1;
a=0;
c=0;
i=0;
j=0;
n=0;
x=0;
y=0;
r=0;
}
otherorder((ch = = "N") || (ch = =
"n")) {
choice = 0;
}
order {
post("\n\t\t\tError Input!");
choice = 3;
}
} phase(choice = = 3);
} phase(choice != 0);
post("\n\t\t\t Maraming Salamat Po
!!");
} deploy();
```