octToBinary

Write a program that reads in an octal number, converts the octal number into the equivalent binary number (i.e. converts the number with base value 8 to base value 2) and prints the converted binary number to the screen. You do not need to check user input errors in the program.

A sample program template is given below:

```
#include <stdio.h>
#include <math.h>
int main()
{
   /* Write your code here */
   return 0;
}
```

Some test input and output sessions are given below:

(1) Test Case 1

```
Enter an octal number:
5
The equivalent binary number: 101
```

(2) Test Case 2

```
Enter an octal number:
30
The equivalent binary number: 11000
```

(3) Test Case 3

```
Enter an octal number:
60
The equivalent binary number: 110000
```

(4) Test Case 4

```
Enter an octal number:
100
The equivalent binary number: 1000000
```

```
#include <stdio.h>
#include <math.h>
int main()
{
   int octal,octalq;
   int dec = 0;
   int temp = 0;
   int ult,remainder;

   int binary;
   int temp2 = 1;
   int rembin;
   printf("Enter an octal number:\n");
   scanf("%d",&octal);

   while(octal!=0)
   {
      remainder = octal%10;

      ult=remainder*pow(8,temp);
      dec = dec+ ult;
      temp++;
      octal = octal/10;
   }
   //binary value of octal number

   while (dec!=0)
   {
      rembin = dec%2;
      binary = binary + rembin *temp2;
      temp2 = temp2*10;
      dec = dec/2;
   }

   printf("The equivalent binary number: %d\n",binary);

   return 0;
}
```