H.W 1- decimal to binary

LAB-ASSIGNMENT 2A (Upto Control Structure)

SUBJECT: Computer Lab.

1. Write a C program to check a number is palindrome or not
2. Write a C program to find the size of int, float, double and char.
3. Write a C program to check whether a character is a Vowel or Consonant.
4. Write a C program to find GCD of two numbers.
5. Write a C program to find LCM of two numbers.
6. Write a C program to display the factors of a number.
7. Write a C program to convert Decimal number to binary and vice-versa.
8. Write a C program to convert Decimal number to octal and vice-versa.
9. Write a C program to find the second largest from 5 numbers.
10. Write a C program to calculate nPr and nCr.
11. Write a C program to calculate simple and compound interest of a number where %of interest and year is also given as input.

H.W 2

H.W 3

size

10 20 5 2 17

o/p → 17

printf("%d", sizeof(int)); 4B

Byte = 8 bits

4×8 = 32 bits

(-2^{n-1}) to $(+2^{n-1}-1)$ signed

-2^{32-1} to $+2^{32-1}-1$

-32768 to +32767

short int

char → 1B → 8

-2^{n-1} to $+(2^{n-1}-1)$

-2^{8-1} to $+2^{8-1}-1$

-128 to 127

palindrome or not

mom

767

madam

```
#include <stdio.h>
int main() {
   int n, rev = 0, rem, ori;
   printf("Enter an integer: ");
   scanf("%d", &n);
   ori = n;

   // reversed integer is stored in rev
   while (n != 0) {
      rem = n % 10;
      rev = rev * 10 + rem;
      n /= 10;
   }

   // palindrome if ori and rev are equal
   if (ori == rev)
      printf("%d is a palindrome.", ori);
   else
      printf("%d is not a palindrome.", ori);

   return 0;
}
```

n = 767

rem = 767 % 10 = 7

rev = 0*10 + 7 = 7

n = n/10 = 767/10 = 76

while(76!=0)
{
rem=6
rev
= 7*10+6
= 76
n = 7

7!=0
{rem = 7
rev = 76*10+7
= 767
n=0

while(n!=0) false

if (767 == 767)
767 is a palindrome

```
#include <stdio.h>
#include<math.h>
#include<float.h>


int main()
{

  int n;
long unsigned int m;


//end of main()*/

/*  In general if the data-type takes up n bits in the memory
   then for the signed situation the
   range is from -2^(n-1) to 2^(n-1)-1
   and for the unsigned situation
   the range is from 0 to 2^(n)-1

*/
n=8*sizeof(int);

   //the name of the data type is passed as a string and
   //the size is n which is in bits
   long long unsigned int limit=pow(2,n-1);//storing a
      //frequently required number in the limit

   //printing the data for the signed case
   printf("Size of integer is %d bits and its range is:\n",n);
   printf(" -%llu to %llu\n\n\n",limit,limit-1);

   //printing the data for the unsigned situation
   printf("Size of unsigned integer is %d bits and its range is:\n",n);
   printf(" %d to %llu\n\n",0,2*limit-1);
}//end of range(int,char*)
```

```c
#include <stdio.h>
int main() {
   char c;
   int lowercase_vowel, uppercase_vowel;
   printf("Enter an alphabet: ");
   scanf("%c", &c);

   // evaluates to 1 if variable c is a lowercase vowel
   lowercase_vowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

   // evaluates to 1 if variable c is a uppercase vowel
   uppercase_vowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');

   // evaluates to 1 (true) if c is a vowel
   if (lowercase_vowel || uppercase_vowel)
      printf("%c is a vowel.", c);
   else
      printf("%c is a consonant.", c);
   return 0;
}
```

ok

try it-

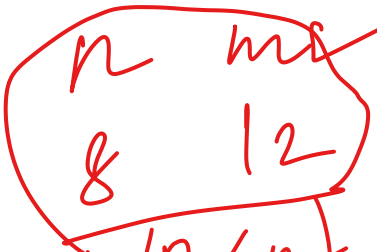
$$ {}^{n}P_{r} = \frac{n!}{(n-r)!} $$

$$ {}^{n}C_{r} = \frac{n!}{r!\,(n-r)!} $$

```
#include<stdio.h>
int main()
{
int n,m,temp;
printf("give first number");
scanf("%d",&n);
printf("give second number");
scanf("%d",&m);
if(n<m)
{
n=n^m;
m=n^m;
n=n^m;
}
while(n%m !=0){

temp=m;
m=n%m;
n=temp;
}

printf("\n GCD= %d",m);
return 0;
}
```



if (n<m) swapping

n m
12, 8

(n>m)

l = 12
s = 8

((12%8)!=0)

temp = 8
m = 4
n = temp

8 | 12 (1
8
4 | 8 (2
8
0

GCD

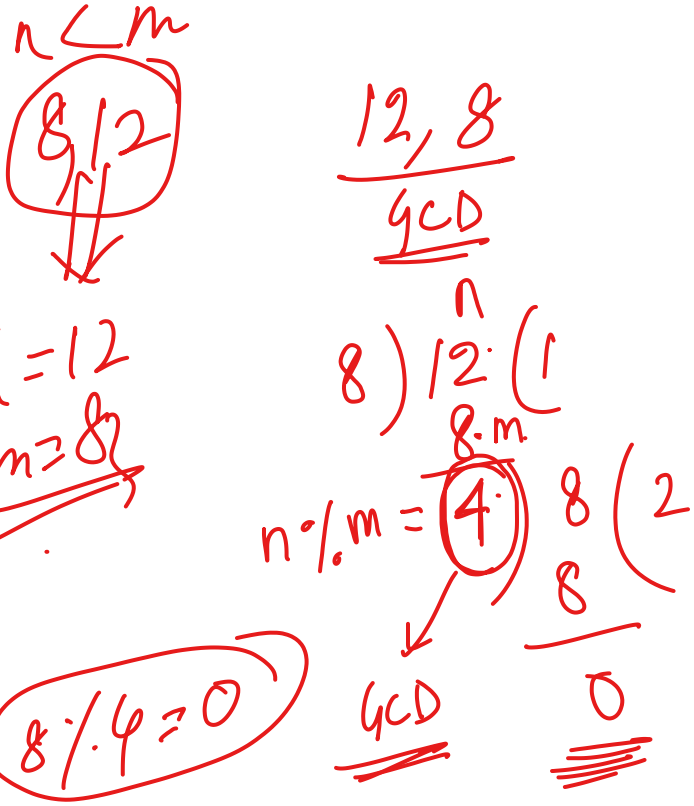
LCM = (num1 * num2) / GCD;

```
#include<stdio.h>
int main()
{
int n,m,temp,num1,num2,LCM;
printf("give first number");
scanf("%d",&n);
printf("give second number");
scanf("%d",&m);
num1=n;
num2=m;
if(n<m)
{
n=n^m;
m=n^m;
n=n^m;
}
while(n%m !=0){

temp=m;
m=n%m;
n=temp;
}

printf("\n GCD= %d",m);

LCM=(num1*num2)/m;
printf("\n LCM= %d",LCM);
return 0;
}
```



```c
#include <stdio.h>
int main() {
   int num, i;
   printf("Enter a positive integer: ");
   scanf("%d", &num);
   printf("Factors of %d are: ", num);
   for (i = 1; i <= num; ++i) {
      if (num % i == 0) {
         printf("%d ", i);
      }
   }
   return 0;
}
```

Factors

12

1 2 3 4 6 12

factors

for (i=1; i<=num; ++i)
if (num % i == 0)

1 2 3 4 6 12

12 - 1 == 0

binary

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$(10101)_2$

$1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0$

$16+0+4+0+1=(21)_{10}$

$10101/10 = 1$

$10101/10 = 1010\ 1*2^0$

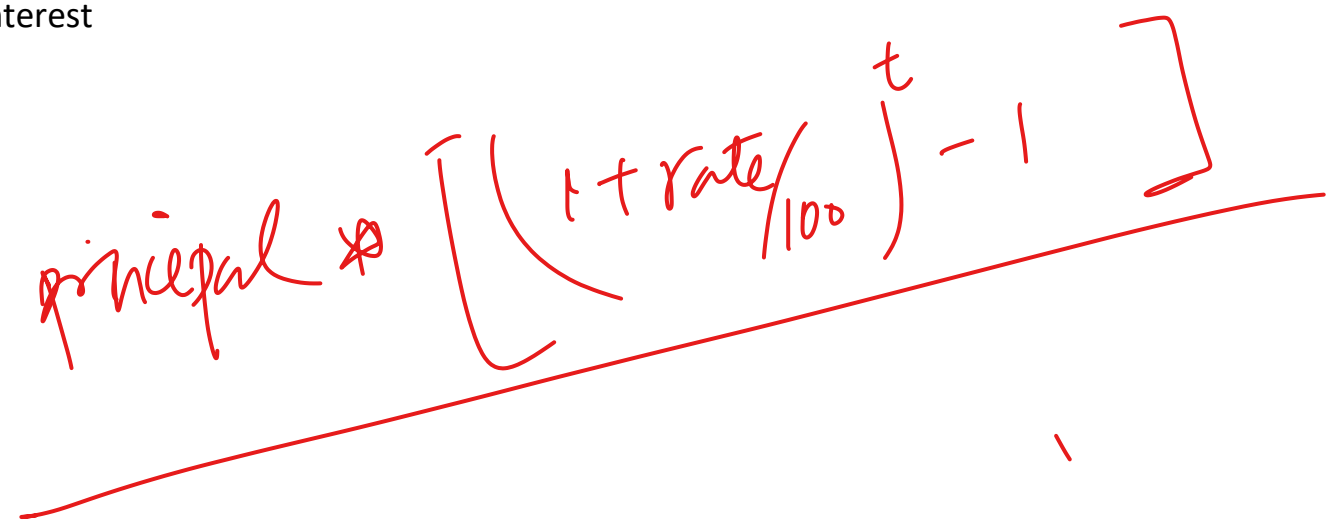
$1010 \% 10 = 0$

$0*2^1$

$1010/10 = 101$

```
#include<stdio.h>
#include<math.h>
int main()
{
	printf("ENTER PRINCIPAL,INTEREST RATE PER ANNUM,TIME IN YEARS \n");
	double principal,rate,time;
	scanf("%lf %lf %lf",&principal,&rate,&time);
	//SI=p*r*t/100
	double si=(principal*rate*time)/100;//the simple interest
	printf("SIMPLE INTEREST =>%lf ",si);//printing the simple interest
	double ci=principal*(pow((1+(rate/100)),time)-1);
	//the compound interest
	printf("\nCOMPOUND INTEREST =>%lf \n",ci);//printing the compound
interest

}
```



```
#include <math.h>
#include <stdio.h>
int convert(long long n);
int main() {
   long long n;
   printf("Enter a binary number: ");
   scanf("%lld", &n);
   printf("%lld in binary = %d in decimal", n, convert(n));
   return 0;
}

int convert(long long n) {
   int dec = 0, i = 0, rem;
   while (n != 0) {
      rem = n % 10;
      n /= 10;
      dec += rem * pow(2, i);
      ++i;
   }
   return dec;
}
```