Module – I

Number theory Problems

**Factors of an integer number**

**1. Given an Integer number M check whether it is even or odd.**

Sol: divide to 2 and find whether remainder is 0 or 1. If remainder is 0 given number is even otherwise odd.

#include<stdio.h>

int main()

{

int M;

scanf("%d",&M);

if(M%2==0) printf("%d is even",M);

else

printf("%d is odd",M);

}

*Note: If % operator is not allowed remainder can be found in the following way*

*Method 1:*

*Quotient=M/2. ( Here Quotient is a integer variable)*

*Remainder=M-Q\*2*

*Method 2:*

*Using bit manipulation.*

*We need to separate Least Significant Bit(LSB) of given number M. If M is ODD then LSB is 1 otherwise LSB is 0. Any bit in binary form of an integer can be separated by ANDing with a mask which has 1 in the required bit position and 0’s else where.*

*Here Mask= 0x01 ( Hexa decimal 01);*

*So if (M & 0x01) is 0 M is EVEN else M is ODD*

***2. Given two integers M and N check whether N is a divisor(factor) of M.***

Sol: Divide M by N and check whether remainder is 0. If so N is a factor of M.

#include<stdio.h>

int main()

{

int M,N;

scanf("%d%d",&M,&N);

if(M%N==0) printf("%d is a factor of %d\n",N,M);

else

printf("%d is a not a factor of %d\n",N,M);

}

***3. Given an integer M write a program to find and print all factors of M.***

Sol: Possible factors of a given an integer M are between 1 to M. So we generate N from 1 to M divide M with each of N and print N if remainder is 0.

#include<stdio.h>

int main()

{

int M,N;

scanf("%d",&M);

for(N=1;N<=M;N++)

if(M%N==0) printf("%d is a factor of %d\n",N,M);

else

printf("%d is a not a factor of %d\n",N,M);

}

*Note: Here we get a few important ‘is a factor’ messages and many unimportant ‘not a factor’ messages. We can remove the else part to avoid unimportant messages.*

***4. Extend the above problem to find number of factors(excluding the number itself), sum of factors and also check whether the given number M is a perfect number.***

Sol: Possible factors of a given an integer M(excluding itself) are between 1 to M/2. So we generate N from 1 to M/2 divide M with each of N and increment a counter C if remainder is 0. Also add N to S if remainder is 0. After finding sum S of all factors if S==M then M is called Perfect Number otherwise not.

#include<stdio.h>

int main()

{

int M,N,C=0,S=0;

scanf("%d",&M);

for(N=1;N<=M/2;N++)

if(M%N==0) {

C++;

S=S+N;

}

printf("No of factors=%d Sum of factors=%d\n",C,S);

if(M==S)printf("Given number %d is a perfect number\n",M);

else

printf("Given number %d is not a perfect number\n",M);

}

***5. Find a number between two positive integers A,B(A<B) with largest number of factors***

Sol: Generate **M** over the range from **A to B** and for each **M** and generate **N** over the range from **1 to M/2**, find number of factors **nf** for every number **M**. Then compare number of factors **nf** with maximum so far **maxfactor** and if is greater modify **maxfactor** and also assign the current **M** to **maxnumber** as the number with maximum number of factors.

At the end **maxnumber** tells us the number with maximum number of factors and **maxfactor** is the maximum number of factors.

#include<stdio.h>

void main()

{

int A,B,M,N,nf,maxfactor=0,maxnumber=0;

printf("enter values for A and B\n");

scanf("%d%d",&A,&B);

for(M=A;M<=B;M++)

{

nf=1;

for(N=1;N<=M/2;N++)

if(M%N==0)nf++;

if(nf>maxfactor){maxfactor=nf;

maxnumber=M;

}

}

printf("%d has maximum number of %d factors\n",maxnumber,maxfactor);

}

6. Given a range of integers from A to B(assume A=1) find top K integers with highest number of factors.

Sol: Indexing example. We used bubble sort to sort in factors array in descending order. Instead of swapping factors array values, we swap corresponding index values stored in index array. Note that we access elements of factors array via values of index array.

This allows us get values of factors array in descending order without really sorting factors array. This is the main advantage of INDEXING.

#include<stdio.h>

int main()

{

int M,N,A,B,factors[1000],index[1000],i,j,C,t;

scanf("%d%d",&A,&B);/\* assume A is 1 \*/

for(M=1;M<=B;M++)

{

C=0;

for(N=1;N<=M/2;N++)

if(M%N==0)C++;

factors[M]=C;

}

for(M=1;M<=B;M++)

index[M]=M;

for(j=1;j<=B-1;j++)

{

for(i=1;i<=B-1;i++)

if(factors[index[i]]<factors[index[i+1]])

{

t=index[i];

index[i]=index[i+1];

index[i+1]=t;

}

}

for(i=1;i<=B;i++)

printf("%6d%6d\n",index[i],factors[index[i]]);

}