Number Theory

Anowar Hossain Anu

Computer Science And Engineering

Islamic University,Bangladesh

Mail:anowar1112@gmail.com

Mobil:01738378684

Dist:Chapai Nawabganj Sodor

1.Single Prime Number efficient algorithm : this algorithm is 10^14 number .

```
int single_prime(long long n)

{

   if(n<2)return 0;

   if(n==2) return 1;

   if(n%2==0) return 0;

   for(int i=3;i<=sqrt(n);i+=2)

   if(n%i==0)

   return 0;

   return 1;

}
```

Example: n=1000 ;

If 1000 is prime number than yes . Others No .

18.anagrammatic_prime number total 22 ti . ai sob prime number gulo jevabe lekhi sob somoy prime hobe .

```
anagrammatic_prime[]={2,3,5,7,11,13,17,31,37,71,73,79,97,113,131,

199,311,337,373,733,919,991};
```

. **this algorithm** used to **10^18** number .this number used to **single prime** | This **algorithm** called by **miller_rabin theory**|

```
const int S=8;
long long mult_mod(long long a,long long b,long long c)
{
   a=a%c;
   b=b%c;
   long long ret=0,tmp=a;
   while(b)
   {
      if(b&1)
      {
         ret+=tmp;
         if(ret>c)
         ret-=c;
      }
      tmp<<=1;
      if(tmp>c)
      tmp-=c;
      b>>=1;
   }
   return ret;
}
```

```
long long pow_mod(long long a,long long n,long long mod)
{
    long long ret=1;
    long long temp=a%mod;
    while(n)
    {
        if(n&1)
        ret=mult_mod(ret,temp,mod);
        temp=mult_mod(temp,temp,mod);
        n>>=1;
    }
    return ret;
}
bool check(long long a,long long n,long long x,long long t)
{
    long long ret=pow_mod(a,x,n);
    long long lll=ret;
    for(int i=1;i<=t;i++)
    {
        ret=mult_mod(ret,ret,n);
        if(ret==1 and lll!=1 and lll!= n-1)
        return true;
        lll=ret;
    }
    if(ret!=1)
    return true;
    else
    return false;
}
```

```
bool miller_rabin(long long n)
{
   if(n<2) return false;
   if(n==2)return true;
   if((n&1)==0) return false;
   long long x=n-1;
   long long t=0;
   while((x&1)==0)
   {
      x>>=1;
      t++;
   }
   srand(time(NULL));
   for(int i=0;i<S;i++)
   {
      long long a=rand()%(n-1)+1;
      if(check(a,n,x,t))
      return false;
   }
   return true;
}
```

here , n number single number . if this number is prime than YES others NO .

3. **Seive prime algorithm : this algorithm used 1 to N prime number .**

```
#define MAX 1002   //it is N

#define LMT 19     // it is squre(MAX)  and LMT is odd number

#define i64 long long

long int flag[MAX+1], primes[MAX], total;

#define ifc(x) (flag[ x >> 6 ] & (1 << ((x >> 1) & 31)))

#define isc(x) (flag[ x >> 6]|=(1 << ((x >> 1) & 31)))

void sieve()

{

   int i, j, k;

   primes[0] = 2;

   total = 1;

   for(i = 3; i <=LMT; i += 2)

   {

      if(!ifc(i))

      {

         primes[total++] = i;

         for(j = i*i; j <=MAX; j += 2*i)

         isc(j);

      }

   }

   for(i = LMT+2; i <=MAX; i += 2)    //LMT value must be odd number

   {

      if(!ifc(i))

      {

        primes[total++] = i;

      }

   }

}
```

This algorithm used array primes[] , store to prime number 1 to N.

4.Range Primes Number :this algorithm previous store , max range difference.how many A from B difference prime number? And A and B 10^18 projanto hoy . tahole mone rakhte hobe . A and B difference 10^7 er besi hobe na . tai age 10^7 projnato prime store kore rakhte hobe .

```
#define N 1000006  // max difference number A and B

char x[N];

long long range_prime(long long a,long long b)

{

   long long t=0,n=0,d,i,k,che=0,j=0;

   if(a>b)    return 0;

   if(a==b and b==1) return 0;

   CLR(x);      //x array er sob position a 0 rakhte hobe

   if(a<3)

   {

      cout<<"2";

      che=1;

      t=1;

      a=3;

   }

   if(a%2==0)

   a=a+1;

   if(b%2==0)

   b=b-1;

   d=sqrt(b);

   i=0;

   while(prime[i]<=d)                         //prime[] array te age max difference prime store kore rakhte hobe .

   {

      n=ceil((a*1.0)/prime[i]);

      if(n==1)
```

```
        n=2*prime[i];
      else
        n=n*prime[i];
      for(k=n;k<=b;k=k+prime[i])
      x[k-a]=1;
      i=i+1;
   }
   for(i=a;i<=b;i=i+2)
   {
      if(x[i-a]!=1)
      {
        if(che==1)
        cout<<" "<<i;
        else
        cout<<i;
        che=1;
         t=t+1;
      }
   }
   cout<<endl;
   return t;  // total prime number range er modhe
}
```

If A =3 and B =10 hoy .tahole

Print korbe 3 , 5,7

And total 3 ta prime ache range er modhe .

5. a single number how many divisor and use to 10^14 ?

```
int divi[100000];

int number_of_divisor(int n)

{

   int i,s,p;

   s=0;

   for(i=1;i<=sqrt(n);i++)

   {

      if(n%i==0)

      {

         divi[s]=i;

         s++;

         p=n/i;

         if(p!=i)

         {

            divi[s]=i;

            s++;

         }

      }

   }

   return s;

}
```

Example: n=10;

n=1,2,5,10 // this number is stor divi array.

S is count of divisor .

6.Seive divisor algorithm .this algorithm use to number of divisor , prime divisor, prime factor , sum of all divisor .range to 10^14 projnato ber kora jai . number of divisor , sum of all divisor .

```
long long str[1000];

i64 divisor(i64 n)

{

   i64 ret = 1,k,ssss=1,lll=1, i,j=0,cnt, rt = (LL)sqrt((double)n);

   for(i = 0; i < total && primes[i] <= rt; i++)

   {

      if(n % primes[i] == 0)  // before stor primes[] number  sqrt(n) projanto

      {

         n /= primes[i], cnt = 1;

         str[j]=primes[i];

         j++;

         while(n % primes[i] == 0)

         {

            n/= primes[i], cnt++;

            str[j]=primes[i];

            j++;

         }

         rt = (int)sqrt((double)n);

         ret*=(cnt+1);

        ll*=(pow(primes[i],(cnt+1))-1);

         lll*=(primes[i]-1);

         if(ll%lll==0)

         {

            ll=ll/lll;

            lll=1;

         }

      }
```

```
   }
   if(n > 1)
   {
      str[j]=n;
      j++;     ret <<= 1;
     ll=ll*(poww(n,2) -1);
     lll*=(n-1);
     if(ll%lll==0)
     {
        ll=ll/lll;
     }
   }
}
```

When n=20;   // j is number of prime divisor

Prime divisor =2,5 //str[]

All divisor of sum =((2^(2+1))-1)*((5^(1+1))-1)/(2-1)*(5-1)

=(7*24)/4

= 168/4=42   // ll is sum of divisor

Number of divisor=(2+1)*(1+1)=3*2=6   //cnt is number of divisor

7. 1 from n projanto every number , number of divisor . 10^15 projanto ninnoy kora jai .

Example : n=5 ;

1= 1;   2= 1,2 ; 3= 1,3 ; 4=1,2,4 ; 5=1,5 ;

All divisor number =1+2+2+3+2=10;

```
  x=sqrt(n);
  for(k=1;k<=x;k++)
    s=s+n/k;
  r=2*s-x*x;
```

this code n=5 , hole r=10;

8. 1 from n projanto every number , sum of divisor .10^15 projanto ninnoy kora jabe .

Example : n=5 ;

1= 1; 2= 1,2 ; 3= 1,3 ; 4=1,2,4 ; 5=1,5 ;

Every number sum of divisor =1+(1+2)+(1+3)+(1+2+4)+(1+5)

=1+3+4+7+6

=21;

```
 rt = (int)sqrt((double)n);

i64 sum = 0,s=0;

for(i = 2; i <= rt; i++)

 {

     j = n / i;

    sum+= (i64)i*(j-i+1) + (i64)j*(j+1)/2 - (i64)i*(i+1)/2;

}

sum=sum+((n*(n+1))/2.0)+(n-1);
```

this code n=5 hole , sum = 21 // sum is variable

9.number of Divisor , number of prime factor,sum of all divisor .10^18 projanto ninnoy kora jai.

```
typedef unsigned long long uint64;

typedef long double float80;

typedef long long ll;

typedef long long int64;

typedef unsigned uint;

typedef unsigned char uint8;


static const uint64 PRIME_MAX = 1024;

static const uint64 THRESHOLD = PRIME_MAX * PRIME_MAX;

static const uint64 POLLARD_RHO_M = 250;


static const uint64 MOD = 1000000000000000000ull;

static const float80 MOD_INV = float80(1) / MOD;


static const int primes[] = {

        2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97,101,103,107,109,113,127,131,137,139,149,151,157,163,16
7,173,179,181,191,193,197,199,211,223,227,229,233,239,241,251,257,263,269,271,277,281,283,293,307,311,313,317,331,337,347,349,353,3
59,367,373,379,383,389,397,401,409,419,421,431,433,439,443,449,457,461,463,467,479,487,491,499,503,509,521,523,541,547,557,563,569,
571,577,587,593,599,601,607,613,617,619,631,641,643,647,653,659,661,673,677,683,691,701,709,719,727,733,739,743,751,757,761,769,773
,787,797,809,811,821,823,827,829,839,853,857,859,863,877,881,883,887,907,911,919,929,937,941,947,953,967,971,977,983,991,997,1009,1
013,1019,1021,1031

};


inline uint xrand(void)

{

        static uint x = 123456789, y = 362436069, z = 521288629, w = 88675123;

        uint t = x ^ (x << 11); x = y; y = z; z = w;

        return w = (w ^ (w >> 19)) ^ (t ^ (t >> 8));

}
```

```
inline uint randrange(uint64)
{
        return (uint64(xrand()) * 0xFFFFFFFF) >> 32;
}

template <typename T>
T gcd(T a, T b)
{
        if(b > a)
  {
                T tmp = a;
                a = b;
                b = tmp;
        }
        while(1)
  {
                if(!b) return a;
                a-=b;if(a>=b){a-=b;if(a>=b){a-=b;if(a>=b){a-=b;if(a>=b){a%=b;}}}}
                if(!a) return b;
                b-=a;if(b>=a){b-=a;if(b>=a){b-=a;if(b>=a){b-=a;if(b>=a){b%=a;}}}}
        }
}
inline uint square_add_mod(uint a, uint c, uint mod, float80)
{
        return (uint64(a) * a + c) % mod;
}
inline uint mul_mod(uint a, uint b, uint mod, float80)
{
        return uint64(a) * b % mod;
}
```

```
template <typename T>

T pow_mod(uint base, T exp, T mod, float80 modi)

{

        T ret = 1;

        T q = base;

        while(exp) {

                if(exp & 1) {

                        ret = mul_mod(ret, q, mod, modi);

                }

                exp >>= 1;

                q = mul_mod(q, q, mod, modi);

        }

        return ret;

}


template <typename T>

bool miller_rabin_pass(uint base, uint m, T exp, T mod, float80 modi)

{

        T n = pow_mod(base, exp, mod, modi);

        if(n == 1)

        return true;

        for(uint i = 0; i < m; ++i) {

                if(n == mod - 1)

                        return true;

                n = mul_mod(n, n, mod, modi);

        }

        return n == mod - 1;

}

inline uint ilog2(uint64 x)

{
```

```
    union Data {
        uint64 u64;
        double d;
    } n;
    n.d = double(x) + 0.5;
    return (n.u64 >> 52) - 1023;
}

bool miller_rabin(uint64 n)
{
    static const uint BASES1[] = {2, 3};
    static const uint BASES2[] = {2, 299417};
    static const uint BASES3[] = {2, 7, 61};
    static const uint BASES4[] = {15, 176006322, 4221622697u};
    static const uint BASES5[] = {2, 2570940, 211991001, 3749873356u};
    static const uint BASES6[] = {2, 2570940, 880937, 610386380, 4130785767u};
    static const uint BASES7[] = {2, 325, 9375, 28178, 450775, 9780504, 1795265022};
    if(n <= 1)
    {
        return false;
    }
    if(n <= 3) {
        return true;
    }
    uint64 d = n - 1;
    uint s = ilog2(d & -d);
    d >>= s;
    uint bases_size;
    const uint* bases;
    if(n < 1373653)
    {
```

```
        bases_size = 2;
        bases = BASES1;
} else if(n < 19471033) {
        bases_size = 2;
        bases = BASES2;
} else if(n < 4759123141ull) {
        bases_size = 3;
        bases = BASES3;
} else if(n < 154639673381ull) {
        bases_size = 3;
        bases = BASES4;
} else if(n < 47636622961201ull) {
        bases_size = 4;
        bases = BASES5;
} else if(n < 3770579582154547ull) {
        bases_size = 5;
        bases = BASES6;
} else {
        bases_size = 7;
        bases = BASES7;
}
if(n < 0x100000000ull) {
        for(uint rep = 0; rep < bases_size; ++rep) {
                if(!miller_rabin_pass<uint>(bases[rep], s, d, n, 0))
                        return false;
        }
} else {
        float80 modi = float80(1) / n;
        for(uint rep = 0; rep < bases_size; ++rep) {
                if(!miller_rabin_pass<uint64>(bases[rep], s, d, n, modi))
                        return false;
```

```
		}
	}
	return true;
}

template <typename T>
T pollard_rho(T n)
{
	if(!(n & 1))
	{
		return 2;
	}
	T y = randrange(n - 1) + 1;
	T c = randrange(n - 1) + 1;
	T m = POLLARD_RHO_M;
	T g = 1, q = 1;
	T x, ys;
	uint64 r = 1;
	float80 n_inv = float80(1) / n;
	while(g == 1) {
		x = y;
		for(uint i = 0; i < r; ++i) {
			y = square_add_mod(y, c, n, n_inv);
		}

		T k = 0;
		while(k < r && g == 1) {
			ys = y;
			T end = (r - k < m ? r - k : m);
			for(uint i = 0; i < end; ++i) {
				y = square_add_mod(y, c, n, n_inv);
```

```
                T dif = (x >= y ? x - y : y - x);
                q = mul_mod(q, dif, n, n_inv);
            }
            g = gcd(q, n);
            k += m;
        }
        r <<= 1;
    }
    if(g == n) {
        while(1) {
            ys = square_add_mod(ys, c, n, n_inv);
            T dif = (x >= ys ? x - ys : ys - x);
            g = gcd(dif, n);
            if(g > 1)
            break;
        }
    }
    return g;
}



uint64 ps[1000];
uint64 pwr[1000];
uint factors(uint64 n)
{
    if(n <= 1)
    return 0;
    uint pos = 0;
    uint v = sqrt(n);

    if(uint64(v) * v == n && miller_rabin(v))
```

```
{
        ps[pos] = v;
        pwr[pos] = 2;
        ++pos;
        return pos;
}

uint e = ilog2(n & -n);
if(e > 0)
{
        n >>= e;
        ps[pos] = 2;pwr[pos++] = e;
        v = sqrt(n);
}
uint end = (n > THRESHOLD ? PRIME_MAX : v + 1);

uint p_idx = 1;
uint p = primes[p_idx++];
while(p < end) {
        if(n % p == 0) {
                n /= p;
                uint e = 1;
                while(n % p == 0) {
                        n /= p;
                        ++e;
                }
                end = (n > THRESHOLD ? PRIME_MAX : sqrt(n) + 1);
                ps[pos] = p;pwr[pos++] = e;
        }
        p = primes[p_idx++];
}
```

```
p = primes[p_idx-1];
uint64 cut_off = uint64(p) * p;
if(n > 1)
{
	if(cut_off > n || miller_rabin(n))
	{
		ps[pos] = n;pwr[pos++] = 1;
		return pos;
	}
	while(1)
	{
		uint64 p;
		if(n < 0x100000000ull)
		{
			p = pollard_rho<uint>(n);
		}
		else
		{
			p = pollard_rho<uint64>(n);
		}
		if(!miller_rabin(p))
		continue;
		n /= p;
		uint e = 1;
		while(n % p == 0) {
			n /= p;
			++e;
		}
		ps[pos] = p;pwr[pos++] = e;
		if(n <= cut_off || miller_rabin(n)) {
			if(n > 1) {
```

```
                                                  ps[pos] = n;pwr[pos++] = 1;
                                        }
                                        break;
                              }
                    }
          }
          return pos;
}

pair<uint64,uint64>A[100];
void solve()
{
  uint64 n,temp;
  scanf("%llu",&n);
  int sz = factors(n);

  for(int i=0;i<sz;i++)
  {
    A[i] = make_pair(ps[i],pwr[i]);
  }
  sort(A,A+sz);
  for(int i=0;i<sz;i++)
  {
    ll X = A[i].first;
    ll upto = A[i].second;
    for(int j=0;j<upto;j++)
    {
      printf("%llu ",X);
    }
  }
  printf("\n");
```

}

When n=20;

Prime divisor =2,5 // ps []

Koto bar ache =2 ,1 // pwr[] koto bar ache

All divisor of sum =((2^(2+1))-1)*((5^(1+1))-1)/(2-1)*(5-1)

=(7*24)/4

= 168/4=42 // ll is sum of divisor

Number of divisor=(2+1)*(1+1)=3*2=6 //cnt is number of divisor

10.prime factor , a number how many prime factor and ki ki prime factor ache . 10^7 projanto ninnoy kora jabe.

```
vector<int>primefc[1001];
int check[1001];
int prime_factor()
{
  int i,j,k,nnn;
  for(i=2;i<=1000;i+=2)
  {
    primefc[i].push_back(2);
    check[i]=1;
  }
  for(i=3;i<=1000;i+=2)
  {
    if(!check[i])
    {
      for(j=i;j<=1000;j+=i)
      {
        primefc[j].push_back(i);
        check[j]=1;
      }
    }
  }
  for(i=2;i<=1000;i++)
  reverse(primefc[i].begin(),primefc[i].end());
  return 0;
}
```

Whe N=20;

Tahole N er prime factor =2,5

11.Phi function ,relative_prime,co_prime , Euler Totient Function and gcd(1,n)=1…..gcd(n,n) =1. used to 10^7 projanto .

```
#define MAX 5000000

int etf[MAX + 1];

inline void coprime()

{

   register int i, j;

   for(etf[2] = 1, j = 4; j <= MAX; j+=2)

   etf[j] = j >> 1;

   for(i = 3; i <= MAX; i+=2)

   {

      if(!etf[i])

      {

         for(etf[i] = i-1, j = i<<1; j <= MAX; j+=i)

         {

            if(!etf[j])

            etf[j] = j;

            etf[j] = etf[j] / i * (i-1);

         }

      }

   }

}
```

Example: n=10;

Prime factor=2,5

Ans=10*(2-1)*)*(5-1)/(2*5);

=4

Discuss: 1/10,2/10,3/10,4/10,5/10,6/10,7/10,8/10,9/10,10/10 ; here vagfoler man <=1 tt=5 ; //ans+1 hobe .

Gcd(1,10)=1,gcd(Gcd(2,10)=2, Gcd(3,10)=1, Gcd(4,10)=2, Gcd(5,10)=5, Gcd(6,10)=2, Gcd(7,10)=1, Gcd(8,10)=1, Gcd(9,10)=3, Gcd(10,10)=10

Here gcd(a,b)=1 horche ans =4;

12.Phi function,erular.coprime.reletive.prime.phi_funtion

10^12 projanto ninnoy kora jai .

```
long long l,r,k,tests;

int pr[1<<20];

vector<int> vec[1<<20];

int ans;

long long get_phi(long long val,vector<int>&primes)

{

        long long res=val;

        for (int i=0;i<primes.size();i++)

        {

                while (val%primes[i]==0)

                val/=primes[i];

                res=res/primes[i]*(primes[i]-1);

        }

        if (val>1)

        res=res/val*(val-1);

        return res;

}

//www.hackerearth.com/code-monk-number-theory-iii/algorithm/monk-and-etf/description/

int main(){

        pr[1]=1;

        for (int i=2;i<=1000000;i++)

                if (pr[i]==0)

                        for (int j=i*2;j<=1000000;j+=i)

                                pr[j]=1;


        cin>>tests;

        for (;tests;--tests)
```

```
	{
		cin>>l>>r;
		for (long long i=l;i<=r;i++)
		vec[i-l].clear();
		for (int i=2;i<=1000000;i++)
		{
			if (pr[i])
			continue;
			long long bnd=l/i*i;
			if (l%i>0)
			bnd+=i;
			for(long long j=bnd;j<=r;j+=i)
			vec[j-l].push_back(i);
		}
		ans=0;
		cout<<l<<" from "<<r<<" Euler Totient Function value is : ";
		for (long long i=l;i<=r;i++)
		{
			long long Q=get_phi(i,vec[i-l]);
			cout<<Q<<" ";
		}
		cout<<endl;
	}
	return 0;
}
```

When l=2 and r=5;

Ans =1,2,2,4;

Ager problem ta dekhle hobe . agertar are tar modhe difference holo ager sdudhu 10^7 projanto ninnoy kora jabe . kintu atar L and R er difference 10^ 6 hole hobe . L and R er man 10^12 projnat ninnoy kora jai .

13.a^b sum of divisor .A and B very large number ;

A=10;

B=5;

Ans=A^B

Ans=10^5;

=10*10*10*10*10;

Ai sob problem er khetre A er sob prime divisor ninnoy korte hobe .and Prime divisor gulo koto bar ache ta ninnoykorte hobe .

Sieve divisor algorithm diye korte hobe .

10=2,5 prime divisor ache

2= 1 bar , 5= 1 bar

Ans=10^5=((2^1)*(5^1))^5

=(2^5)*(5^5);

Sum of divisor =((2^(5+1))-1)*(5^(5+1)-1)/((2-1)*(5-1))   // this is theory

Jeheto , sum er man onek boro hobe tai

Big mod , inversr mod - diye kaj korte hobe .

14.BIG MOD diye onek kaj kora jai .

```
ll mod_pow(ll a,ll b,ll m)  // a^b  and m is mod

{

        ll r=1;

        while(b)

        {

                if(b%2==1)

                r=r*a%m;

                a=(a*a)%m;

                b/=2;

        }

        return r;

}
```

This algorithm a^b . here a and b is very large .

BIG NUMBER code diye inverse mod kaj kora jai . mod of couse Prime number hote hobe

If h=((a^b)%mod)/(c%mod)

Ans=(a^b)%mod ; // ai kaj ti big mod diye korte hobe bigmod(a,b,mod)

Ans1=c%mod // ai kaj tuku bigmod(c-1,mod-2,mod)

Ans=ans*ans1; // ans holo ans .

15. Extended Euclidean Algo , INVERSE MOD this algorithm very important

```
void EE(ll a, ll b, ll *x, ll *y)
{
        if(a==0)
        {
                *x=0;
                *y=1;
                return;
        }
        ll temp_x,temp_y;
        EE(b%a, a, &temp_x, &temp_y);
        *y=temp_x;
        *x=temp_y - (b/a)*temp_x;

}
ll inverse_mod(ll a,ll m)     //gcd(a,m)=1 hote hobe
{
        ll x,y;
        EE(a,m,&x,&y);
        while(x<0)
         x+=m;  // Importantly make it positive
        return x;
}
```

This algorithm a is hoy 1/a and m is mod , mod must be gcd(a,m)=1 hote hobe.

16.BIG mod (a^p)%m this problem is a,b,c is 10^18 projanto kora jai .

```
typedef long long vlong;

vlong bigmul ( vlong a, vlong b, vlong c )

{

   if ( b == 0 ) return 0;

   if ( b & 1 ) {

      return ( a + bigmul ( a, b - 1, c ) ) % c;

   }

   else {

      return ( 2 * bigmul ( a, b / 2, c ) ) % c;

   }

}


vlong bigmod ( vlong a, vlong p, vlong m )

{

   vlong res = 1, x = a % m;

   while ( p ) {

      if ( p & 1 ) res = bigmul ( res, x, m );

      x = bigmul ( x, x, m );

      p >>= 1;

   }

   return res;

}
```

This problem 10^18 *10^18 gun korle rakha jai na . tai ata kea i algorithm diye korte hobe .

17.Base Conversion any base to any convert .

```
long   any_base_to_decimal_base(long int n,long int  m)
{
   long int  i=0;
   long int  ar[1000];
   while(n>0)
   {
      ar[i]=n%10;
      i++;
      n=n/10;
   }
   ar[i]=0;
   long int  ans=0,j;
   for(j=0;j<i;j++)
   {
      ans+=ar[j]*pow(m,j);
   }
   return ans;
}
 char letters(int r)
{
   if(r<=35)
   {
      for(int i=10;i<=35;i++)
      {
         if(i==r)
         return (i-10+'A');
      }
   }
   else
```

```
    {
       for(int i=36;i<=61;i++)
       {
          if(i==r)
          return (i-35+'a');
       }
    }
}
void decimal_base_to_any_base(long int N, long int b)
{
    if (N == 0)
    return;
    long int x = N % b;
    N /= b;
    if(x < 0)
    N+=1;
    decimal_base_to_any_base(N, b);
    if(x>9)
    cout<<letters(x);
    else
    cout<< x < 0 ? x + (b * -1) : x;
    return;
}
```

19.single GCD ,LCM check algorithm :

```
LL gcd(LL a,LL b)

{

   while(b>0)

   {

      a=a%b;

      a=a^b;

      b=b^a;

      a=a^b;

   }

   return a;

}


LL lcm(LL a,LL b)

{

   LL x=(a*b)/gcd(a,b);

   return x;

}
```

If a=10,b=12;

A=2*5

B=2*2*3

Gcd=2;

Lcm=2*2*3*5;

20.NcR algorithm is :-

```
LL ncr1(LL n ,LL r)
{
   if(r>(n/2))
   r=n-r;
   LL s=1,i;
   for(i=0;i<r;i++)
   {
      s=s*(n-i);
      s=s/(1+i);
   }
   return 0;
}


N=5;
R=2;
Ncr=5!/(2!*(5-2)!)
```

21.npr algorithm:

```
LL npr(LL n,LL r)
{
   LL s=1,i;
   for(i=1;i<=r;i++)
   s*=(n-i+1);
   return s;
}
N=5
R=2;
Npr=5!/(2!);
```