long long convertDecimalToBinary(int n)

{

long long binaryNumber = 0;

int remainder, i = 1, step = 1;

while (n!=0)

{

remainder = n%2;

cout << "Step " << step++ << ": " << n << "/2, Remainder = " << remainder << ", Quotient = " << n/2 << endl;

n /= 2;

binaryNumber += remainder\*i;

i \*= 10;

}

return binaryNumber;

**Convert binary to decimal**

int cBToD(long long n)

{

int decimalNumber = 0, i = 0, remainder;

while (n!=0)

{

remainder = n%10;

n /= 10;

decimalNumber += remainder\*pow(2,i);

++i;

}

return decimalNumber;

}

**Large division**

n=strlen(a);

sum=0;

for(i=0;i<n;i++)

{

sum=sum+(a[i]-'0');

if(sum>=p)

sum=sum%p;

if(i!=n-1&&sum<p)

sum=sum\* 10;

else if(i==n-1&&sum>p)

sum=sum%p;

if(sum==0)

printf("Case %d: divisiable",j);

else

printf("Case %d: not divisiable",j);

}

for( p= line; ; p =e)

{

input = strtol (p, &e, 10);

if( p==e )

{

break;

}

count++;

}

printf("%d\n",count);

}

return 0;

}

**TO convert sting to int**

for(j=0, k=0;Condithion ; j++) {

if(dcml[j]=='MybreakPoint') k++;

else

{

dgt = dcml[j]-48;

ara1[k] = ara1[k]\*10 + dgt;

}

}

**For string strtol function(string to long corverter)**

#include<stdio.h>

#include<stdlib.h>

int main()

{

char line[1000000];

char \*p,\*e;

long input;

int count=0;

int i,tc;

scanf("%d",&tc);

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

{

scanf("%[^\n]",line);

p=line;

**For count word without different type of sign(,:”.)**

#include<stdio.h>

#include<string.h>

int main()

{

int i,j,count,t;

char str[10001];

char \*word;

scanf("%d",&t);

for(i=0;i<t;i++)

{

scanf(" %[^\n]",str);

word = strtok(str, " ,!;?. ");

count=0;

while(word != NULL)

{

if(strlen(word) > 0)

{

count++;

}

word = strtok(NULL, " ,!;?. ");

}

printf("%d\n",count);

}

return 0;

}

**int gcd**(int a,int b){

int temp;

while(b!= 0)

{

temp = b;

b = a%b;

a = temp;

}

return a;

}

**Int to string:**

int num = 321;

char snum[5];

// convert 123 to string [buf]

itoa(num, snum, 10);

int l = strlen(snum);

// print our string

printf("%d\n", l);

Vector Sort

sort( v.begin(), v.end() );

Array Short

sort( array, array + n );

**##Dis\_joint set**

void Make\_set(int x){

parent[x] = x;

rank[x] = 0;

}

int Find(int x){

if(x!=parent[x]) parent[x] = Find(parent[x]);

return parent[x];

}

int Union(int &x,int &y){

int px= Find(x),py = Find(y);

if(rank[px]>rank[py])

parent[py] = px;

else{

parent[px]= py;

if(rank[px]==rank[py])

rank[py]++;

}

}

For String data type need : getline(cin,var\_name); for whole line.

**///bidirectional graph input**

typedef pair<int , int> pii;

int main()

{

int ver,eg;

cin>>ver>>eg;

vector<pii> v[ver];

for(int i = 0; i<eg;i++)

{

int source;

pii p;

cin>>source>>p.first>>p.second;

v[source].push\_back(p);

int s2 = p.first;

p.first = source;

v[s2].push\_back(p);

//to make unidirectional just comment out last 3 line

}

for(int i=0;i< ver;i++)

{

cout<< i << " -- ";

for(int j=0 ; j<v[i].size();j++)

{

cout<< v[i][j].first<< "-" << v[i][j].second<< " ";

}

}

}

**Var\_name.at(position)** ,this function returns a value of that position.

**strchr(s1, ch);**

Returns a pointer to the first occurrence of character ch in string s1.

**strstr(s1, s2);**

Returns a pointer to the first occurrence of string s2 in string s1.

**Var\_name.substr(starting index/position,n(**মানে সামনে কত ঘর যাবে) );

String s(“hey bro”);

s.find(“bro”);// this function will return 1st starting position of bro. bro ai word ta paile e hobe. Ar ek bar thak or onek bar ata sudu starting bro ar position ta e return korbe.

output: 4 another: rfind

**strrev(name) for reversing**

**a word.**

Swap string:

String one(“something”);

String two (“nothing”);

One.swap(two);

int main()

{

char line[1000];

string s;

int c=0;

gets(line);

istringstream iS(line);//breaking string into word

while(iS>>s)

{

for(int i=s.length();i>=0;i--){

cout<<s[i-1]; ///reverse word

}

c++;

}

cout<<endl<<"Total word "<<c<<endl;

return 0;

}