```cpp
#include<bits/stdc++.h>

using namespace std;

struct Point{
    double x,y;
    Point(double x=0,double y=0):x(x),y(y){}
};

const double eps = 1e-10;

typedef Point Vector;

Vector operator + (Vector A,Vector B){ return Vector(A.x+B.x,A.y+B.y); }
Vector operator - (Vector A,Vector B){ return Vector(A.x-B.x,A.y-B.y); }
Vector operator * (Vector A,double p){ return Vector(A.x*p,A.y*p); }
Vector operator / (Vector A,double p){ return Vector(A.x/p,A.y/p); }

bool operator < (const Point &a,const Point &b){
    return a.x < b.x || (a.x == b.x && a.y < b.y);
}

int dcmp(double x){
    if( fabs(x) < eps )  return 0;
    else return x < 0 ? -1 : 1;
}

bool operator == (const Point &a,const Point &b){
    return dcmp(a.x-b.x) == 0 && dcmp(a.y-b.y) == 0;
}

double Cross(Vector A,Vector B){ return A.x*B.y - A.y*B.x; }

double Len(Point a,Point b){ return sqrt( (a.x-b.x)*(a.x-b.x) + (a.y-b.y)*
(a.y-b.y) ); }

// 围成凸包的点在 ch 里面

void ConvexHull(Point *p,int n,Point *ch){
    sort(p,p+n);
    int m = 0;
    for(int i=0;i<n;i++)
    {
        while( m > 1 && Cross(ch[m-1]-ch[m-2],p[i]-ch[m-2]) <= 0 ) m--;
        ch[m++] = p[i];
    }
    int k = m;
    for(int i=n-2;i>=0;i--)
    {
        while( m > k && Cross(ch[m-1]-ch[m-2],p[i]-ch[m-2]) <= 0 ) m--;
        ch[m++] = p[i];
    }
    if( n > 1 )  m--;
    double Ans = 0;
    for(int i=1;i<m;i++)
```

```
        Ans += Len(ch[i],ch[i-1]);
    Ans += Len(ch[0],ch[m-1]);
    printf("%.2f",Ans);
}

void read(int &p){
    int t=0;
    char c;
    c=getchar();
    while( c < '0' || c > '9' )
        c=getchar();
    while( c >= '0' && c <= '9' )
    {
        t=t*10+c-'0';
        c=getchar();
    }
    p = t;
}

Point p[10010],ans[10010];

int main(int argc,char ** argv){
    int n;
    read(n);
    for(int i=0;i<n;i++)
    {
        int x,y;
        read(x);
        read(y);
        p[i].x = x;
        p[i].y = y;
    }
    ConvexHull(p,n,ans);
    return 0;
}
```