#include<bits/stdc++.h>

#define pp pop\_back

#define pb push\_back

#define int long long int

#define INF 1e18

#define vec vector<int>

#define pii pair<int,int>

#define REP(i,a,b) for(i=a;i<b;i++)

using namespace std;

int ComputeXorFrom1ToN(int n)

{

if(n%4==0)

return n;

else if(n%4==1)

return 1;

else if(n%4==2)

return n+1;

else

return 0;

}

int isPowerOfTwo(int x)

{

return x && (!(x&(x-1)));

}

void swapno(int a,int b)

{

a^=b;

b^=a;

a^=b;

}

int MSB(int n)

{

// for 32 bit integer

n|=n>>1;

n|=n>>2;

n|=n>>4;

n|=n>>8;

n|=n>>32;

n+=1;

return (n>>1);

}

int clearLSB(int i,int x)

{

// clear all bits from LSB to the i-th bit in x

int mask=~((1<<(i+1))-1);

x&=mask;

return x;

}

int countsetbits(int n)

{

int c=0;

while(x)

{

x&=(x-1);

c++;

}

return c;

}

int setabit(int n,int pos)

{

//setting the bit at position pos

n|=(1<<pos);

return n;

}

int unsetbit(int n,int pos)

{

//unsetting the bit at position pos

n&=(~(1<<pos));

return n;

}

int toggling(int n,int pos)

{

//toggling the bit at position pos

n^=(1<<pos);

return n;

}

int check(int n,int pos)

{

//check bit at position pos is set or unset

return n&(1<<pos);

}

int onecomplement(int n)

{

return (~n);

}

int twocomplement(int n)

{

// -n or (~n+1)

return (~n+1);

}

int strippingLSB(int n)

{

return x & (x-1);

}

int getLSB(int n)

{

return x&(-x);

}

int32\_t main()

{

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL);

cout.tie(NULL);

int t=1;

cin>>t;

while(t--)

{

}

}