//Bit Manipulation

bool Check\_ON(int mask,int pos) //Check if pos th bit (from right) of mask is ON

{

if( (mask & (1<<pos) ) == 0 )return false;

return true;

}

int SET(int mask,int pos) //Save the returned mask into some var //Turn on pos th bit in mask

{

return (mask | (1<<pos));

}

int RESET(int mask,int pos) //Save the returned mask into some var //Turn off pos th bit in mask

{

return (mask & ~(1<<pos));

}

int FLIP(int mask,int pos) //Save the returned mask into some var //Toggle/Flip pos th bit in mask

{

return (mask ^ (1<<pos));

}

int LSB(int mask) // The actual LSB mask

{

return (mask & (-mask));

}

int LSB\_pos(int mask) // 0 based position

{

int mask\_2 = (mask & (-mask));

for(int pos = 0;pos<=15;pos++)

{

if(Check\_ON(mask\_2,pos))

return pos;

}

return -1;//

}

int ON\_Bits(int mask)

{

return \_\_builtin\_popcount(mask);

}

inline int clz(int N) { // O(1) way to calculate log2(X) (int s only)

return N ? 32 - \_\_builtin\_clz(N) : -INF;

}

Taking integer input from a single line string

char buff[100000];

gets(buff);

stringstream ss(buff);

int i = 1;

while(ss>>Arr[i++]); // The string is copied to Arr

N = i-2;