# CS29002 Algorithms Laboratory

## Assignment No: 13

Date: 16–November–2016 (Not for Submission)

---

Algobit Electronics manufactures a type of storage device which can only be accessed bit by bit. There is a read-only array $A$ of size $n$ stored in such a device. The array contains exactly $n$ of the $n+1$ integers $0, 1, 2, \ldots, n$. This means that there is exactly one integer $x$ among $0, 1, 2, \ldots, n$ missing in $A$. Your task is to find this missing $x$.

The array $A$ is unsorted, and because it is in the special device, you can only make queries of the form *readbit*$(i, j)$ in order to read the $j$-th bit of $A[i]$. Bits are numbered $0, 1, 2, \ldots$ from the least significant end. Your program must run in $O(n)$ time (and may use $O(n)$ additional space). Since there is a total of $\Theta(n \log_2 n)$ bits in the $n$ elements stored in $A$, you cannot read all the bits of all the elements of $A$. To be more precise, you can make only $\Theta(n)$ calls of *readbit*.

Algobit Electronics has supplied a pre-compiled binary code *algobit.o* which you should link at the time of compiling your program:

```
$ gcc whoismissing.c algobit.o
$ ./a.out
```

or

```
$ g++ whoismissing.cpp algobit.o
$ ./a.out
```

In the code of Algobit Electronics, you have $n = 10^6$. The code defines three functions. In order to keep your compiler happy, include the following three lines near the beginning of your program (before any function definition). Do not change the prototypes of these functions.

```
extern void registerme ( const char * );
extern int readbit ( int , int );
extern void checksolution ( int );
```

The first thing your *main* function does is the registration of your roll number. This is to be done by invoking *registerme*. You then compute the missing $x$. As explained above, you can access the $j$-th bit of $A[i]$ by calling *readbit*$(i, j)$. When you have computed $x$, invoke *checksolution*$(x)$ in order to verify that you have correctly computed $x$. So your *main* function would look as given below. Since $n = 10^6$ is fixed, you do not need to read or pass $n$ to the functions. Your task is only to write the $O(n)$-time function *findmissing*.

```
int main ()
{
   int x;

   registerme("99FB1331");
   x = findmissing();
   printf("x = %d\n", x);
   checksolution(x);
   exit(0);
}
```

---

Submit a single C/C++ source file. Do not use global/static variables.