CS29003 Algorithms Laboratory

Assignment No: 1

Last date of submission: 27–July–2017

Let f be an integer-valued function of non-negative integers. It is given that f initially increases strictly, reaches a peak at some $n > 0$, and then decreases strictly, that is,

$$f(0) < f(1) < f(2) < \cdots < f(n-1) < f(n) > f(n+1) > f(n+2) > f(n+3) + \cdots.$$

Your task is to locate the integer n at which f reaches its peak. The function f is provided to you as a black box. When you supply i to the black box, it returns $f(i)$. Your challenge is to compute n using as few invocations of the black box as possible.

How to Use the Black Box

An implementation of f is provided to you as a pre-compiled binary file `blackbox1.o`. Download the appropriate file depending upon your compiler (gcc/g++). You need to register your roll number first. Do this using the call:

```
registerme("99FB1331");
```

If you want to evaluate f at an input i, make this call:

```
value = evaluate(i);
```

Eventually, when you have determined n, make the following call to verify whether your answer is correct:

```
verify(n);
```

What You Have to Do

Include the following lines at the beginning of your program.

```
extern void registerme ( const char [] );
extern int evaluate ( unsigned int );
extern void verify ( unsigned int );
```

Write a `main()` function like this:

```
int main ( )
{
   unsigned int n;

   registerme("99FB1331");
   n = findpeak();
   verify(n);
}
```

Now, write the function `findpeak()`.

Link the black-box code during compilation:

```
gcc myprog.c blackbox1.o
```

or

```
g++ myprog.cpp blackbox1.o
```

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