ESc101N: Fundamentals of computing(Lab Session 3)

August 20, 2009

Instructions

1. Please read the question carefully and write the program accordingly
2. Make sure that the TA has graded you program
3. The marks are distributed as follows. You get 60% of the marks if the basic algorithm is current, 20% if you manage to compile and execute and 20% for writing the code cleanly, i.e. using proper variable names, intending and making the code more readable.

Question 1. Recall the following recursive equation on binomail coefficients

$$\binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r}.$$

(a) (5 marks) Use the above recursive formula to write a function `int nChooseRMod2(int, int)` to compute the binomial coefficients $\binom{n}{r}$ mod 2.

(b) (5 marks) Use the function defined in the above exercise to draw, given an integer n, the Pascals triangle (mod 2) of height n. Recall that the Pascals triangle (mod 2) of height n consists of $n+1$ lines of integers 0 and 1 where for $1 \leq r \leq \ell \leq n$, the $r+1$st integer in the $\ell+1$st line is the value of $\binom{\ell}{r}$ mod 2.

Hint: Write a file `choose.c` which contains the definition of the function `int nChooseRMod2(int, int)`. Compile and debug it. Write two files `check.c` and `pascal.c` where `check.c` has a main function that takes two integers n and r are prints $\binom{n}{r}$ and the file `pascal.c` has a main function that prints the Pascals triangle. Here how you can compile them.

```
$ gcc -std=c99 -c choose.c
$ gcc -std=c99 -c check.c
$ gcc -std=c99 -c pascal.c
$ gcc -o check check.o choose.o
$ gcc -o pascal pascal.o choose.o
$ # the sample output

$ ./check
enter the value of n: 4
enter the value of r: 2
the value of 4 choose 2 mod 2 is 0
```

```
$ ./pascal
enter the height of the pascals triangle: 25
1
11
101
1111
10001
110011
1010101
11111111
100000001
1100000011
10100000101
111100001111
1000100010001
11001100110011
101010101010101
1111111111111111
10000000000000001
110000000000000011
1010000000000000101
11110000000000001111
100010000000000010001
1100110000000000110011
10101010000000001010101
111111110000000011111111
1000000010000000100000001
11000000110000001100000011
```