```
/*
COMP9315 14s2 Final Exam

Build data files with rand distribution of tuples in Pages
*/

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <fcntl.h>
#include "Scan.h"
#include "Page.h"

#define MAX_TUPLE_LENGTH 200

int main(int argc, char **argv)
{
        int file;
        int initPage(Page *);

        // deal with command-line parameters
        if (argc < 4) {
                fprintf(stderr,"Usage: %s File Ntuples
Seed\n",argv[0]);
                exit(1);
        }
        if ((file = open(argv[1], O_CREAT|O_WRONLY)) < 0) {
                fprintf(stderr,"Cannot open %s\n",argv[1]);
                exit(1);
        }
        int all_tuples = atoi(argv[2]);
        int seed = atoi(argv[3]);

        // special case ... single empty page
        if (all_tuples == 0) {
                Page p;
                initPage(&p);
                writePage(file, 0, p);
                close(file);
                return EXIT_SUCCESS;
        }

        // generate
        srand(seed);
        Page p;
        int pid = 0, off = 0, id = 0;
        int tid = 0, tid_max = initPage(&p);
        int ntuples = 0;
        printf("Page %d, #tuples %d\n", pid, tid_max);
        while (ntuples < all_tuples) {
                char tup[200];
                id++;
                sprintf(tup,"a%d,b%d,c%d",id,id,id);
                sprintf(tup,"s%d,t%d",id,id);
                if (tid == tid_max) {
again:          writePage(file, pid, p);
                        tid_max = initPage(&p);
                        pid++;
                        printf("Page %d, #tuples %d\n", pid,
tid_max);
                        if (rand()%10 < 3) goto again;
                        tid = 0;
                        off = 0;
                }
                printf("(%s) -> (%d,%d)\n",tup,pid,tid);
                strcpy(&(p.tuples[off]), tup);
                p.offset[tid++] = off;
                off += strlen(tup)+1;
                p.ntuples++;
                ntuples++;
        }
        if (tid > 0) { writePage(file, pid, p); }
        close(file);

        // finish up
        return EXIT_SUCCESS;
}

int initPage(Page *p)
{
        bzero(p,1024);
        p->ntuples = 0;
        p->tuples[0] = '\0';
        return MAX_TUPLES_PER_PAGE;
        return rand()%(MAX_TUPLES_PER_PAGE-2);
}
```