Direct Access to Files: Examples in C^{++}

Silly Example

```
// Author: Arnold Rosenbloom
#include <iostream.h>
#include <fstream.h>

int main(int argc, char * argv[]) {
   fstream fs;
   fs.open(argv[1],ios::in|ios::out);

   char newCharacter;
   int byteCount;
   newCharacter='W';

   fs.seekp(0,ios::end); // go to the end of the file
   byteCount=fs.tellp();
   cout << argv[1] << " has " << byteCount << " bytes\n";

   if(byteCount>=5){
       // go to byte 5 and write the new character
       fs.seekp(4, ios::beg);
       fs.write(&newCharacter,sizeof(char));
   }

   if(byteCount>=23){
       // go to the byte 23 and write the newCharacter
       fs.seekp(22,ios::beg);
       fs << newCharacter; // a different way to write
   }
   fs.close();
   exit(0);
}
```


Index Builder

```
#include <fstream.h>
#include <string.h>
class entry {
  public:
     int key;
     int pointer;
};
void makeIndex (char * filename, entry * index,
                int & indexSize) {
  in.open (filename, ios::in);
  indexSize = 0;
  while (!in.fail()) {
     char line[100];
     int byteCount;
     char stnum_string[10];
     int stnum;

     byteCount = in.tellg();
     in.getline(line, 100, '\n');

     strncpy(stnum_string, line, 9);
     stnum_string[9] = 0;
     stnum = atoi(stnum_string);

     indexSize++;
     index[indexSize].key = stnum;
     index[indexSize].pointer = byteCount;
  }
  in.close();
}
```



```
char * lookup (int student_num, entry * index,
               int indexSize, fstream in) {

  for (int i=0; i<indexSize; i++) {
        if (index[i].key == student_num) {
                char line[100];
                in.seekg(index[i].pointer);
                in.getline(line, 100, '\n');
                return line;
        }
  }
  return 0;
}
```

```
void doLookups (entry * index, int indexSize, char * filename)

  fstream in;
  in.open (filename, ios::in);

  while (1) {
     int num;
     char * student_rec;
     cout << "Student number to search for: ";
     cin >> num;
     if (num == -1) break;
     student_rec = lookup (num, index, indexSize, in);
     if (student_rec == 0)
        cout << "Not found.\n";
     else
        cout << student_rec << "\n";
  }
}


int main(void) {
  entry index[500];
  int indexSize;
  makeIndex ("smallfile", index, indexSize);
  doLookups (index, indexSize, "smallfile");
  exit(0);
}
```


Binary Search

```
// Adapted from code written by Arnold Rosenbloom.

int BinarySearch(fstream & fs, Record & t){

        Record try;
        int low=0; int high;
        fs.seekg(0,ios::end);
        high=((fs.tellg()-1)/Record::recordSize);
        while(low<high){
                int middle=(high+low)/2;
                fs.seekg(Record::recordSize*middle);
                fs >> try;
                if(try.compare(t)==0)return(try);
                if(try.compare(t)<0)low=middle+1;
                else high=middle-1;
        }
        return(low);
}
```


Using RRNs

Assume that you have a data file that is already open. Given a relative record number, write a C++ method to go to the beginning of that record in the file in preparation for reading it.

Include in the parameter list all information that is needed by the method; do not assume that you have access to any variables outside the method.

```
void seekToRelRec( fstream & file, const int recSize;
                   int recNum ) {

          file.seekg( recSize*recNum, ios::beg );

}
```