Week 8 Lecture 1

Databases:

Find, Modify, Delete and Add

Objectives

- Understand databases implemented as arrays of structures
  - Add an array of structures
  - Print an array of structures
  - Find an element in an array of structures
  - Modify an element in an array of structures
  - Delete an element in an array of structures

Definitions

- A *database* is simply a collection of data.
- Modern databases are stored on computers and managed by a *database management system* (DBMS)
- Databases contain *records*, that contain related data.
  - E.g. Grade = {Student, Grade}
  - E.g. Student = {Serial Number, Name}

CRUD

- A database performs four tasks on the data commonly called CRUD: Create, Read, Update, Delete
  - Create: add an element to the database
  - Read: find an element in the database
  - Update: change an element in the database
  - Delete: remove an element from the database

Example: Student Attendance Record

- Attendance
  - Student Serial Number: int
  - Date: int
  - Present: int (0 or 1)
- Individual attendance can be represented by a structure

```
struct attendance_rec {
  int sno;
  int date;
  int present;
};
```

Example: Student Attendance Database

- An attendance database can be represented by an array of records
- We need to track the number of elements in the array

```
struct attendance_rec {
  int sno;
  int date;
  int present;
};

struct attendance_rec db[100];
int db_size = 0;
```

Example: Student Attendance Database

- Attendance
  - Student Serial Number: int
  - Date: int
  - Present: int (0 or 1)
-

```
struct attendance_rec {
  int sno;
  int date;
  int present;
};
```

Find

- Look at every record
- Return as soon as match is found

```
struct attendance_rec find(int sno)
{
  for (int i = 0; i < db_size; i++) {
    if (sno == db[i].sno) {
      return db[i];
    }
  }
  return not_found;
}
```

```
struct attendance_rec not_found = {.sno = -1, .date = -1, .present = -1};
```

- Need a special record if none is found.

Calling Find

- Find Student 11

```
a = find(11);
if (-1 == a.present) {
  printf("Student 11 not found\n");
} else {
  printf ("Found student 11!!!\n");
}
```

- Student not found

```
Student 11 not found
```

Add

- Database consists of array or records
- Track database size to add new records
- New record inserted at end of array
  - Size is then incremented

```
struct attendance_rec db[100];
int db_size = 0;

void add(int sno, int date, int present)
{
  db[db_size].sno = sno;
  db[db_size].date = date;
  db[db_size].present = present;
  db_size++;
}
```

Calling Add

- Add student 11
- Now student found
  - Student present

```
Student 11 not found
Student 11 Present on 11
```

```
add(11, 11, 1);
a = find(11);
if (-1 == a.present) {
  printf("Student 11 not found\n");
} else {
  printf ("Student %d %s on %d\n", a.sno, (a.present ? "Present" : "Absent"), a.date);
}
```

Modify

- Find the serial number
- Change the other values when found.
  - Return TRUE when modified.
- May return FALSE not found.

```
int modify(int sno, int date, int present)
{
  for (int i = 0; i < db_size; i++) {
    if (sno == db[i].sno) {
      db[i].date = date;
      db[i].present = present;
      return TRUE;
    }
  }
  return FALSE;
}
```

Calling Modify

- Calling modify
- Now student is absent

```
Student 11 not found
Student 11 Present on 11
Student 11 Absent on 11
```

```
modify(11, 11, 0);
a = find(11);
if (-1 == a.present) {
  printf("Student 11 not found\n");
} else {
  printf ("Student %d %s on %d\n", a.sno, (a.present ? "Present" : "Absent"), a.date);
}
```

Delete

- Find record to delete.
- Quit if not found
- Copy subsequent records to next position.
  - Keeps database compact.
- Reduce db size

```
int delete(int sno)
{
  int i = 0;

  i = find_index(sno);
  if (i < 0) {
    return FALSE;
  }
  // i is rec to be deleted
  for (; i < db_size - 1; i++) {
      db[i] = db[i+1];
  }
  db_size--;
  return TRUE;
}
```

Calling Delete

- Calling Delete
- Again student is not found

```
Student 11 not found
Student 11 Present on 11
Student 11 Absent on 11
Student 11 not found
```

```
delete(11);
a = find(11);
if (-1 == a.present) {
  printf("Student 11 not found\n");
} else {
  printf ("Student %d %s on %d\n", a.sno, (a.present ? "Present" : "Absent"), a.date);
}
```