Implement Phone Book App (v1)

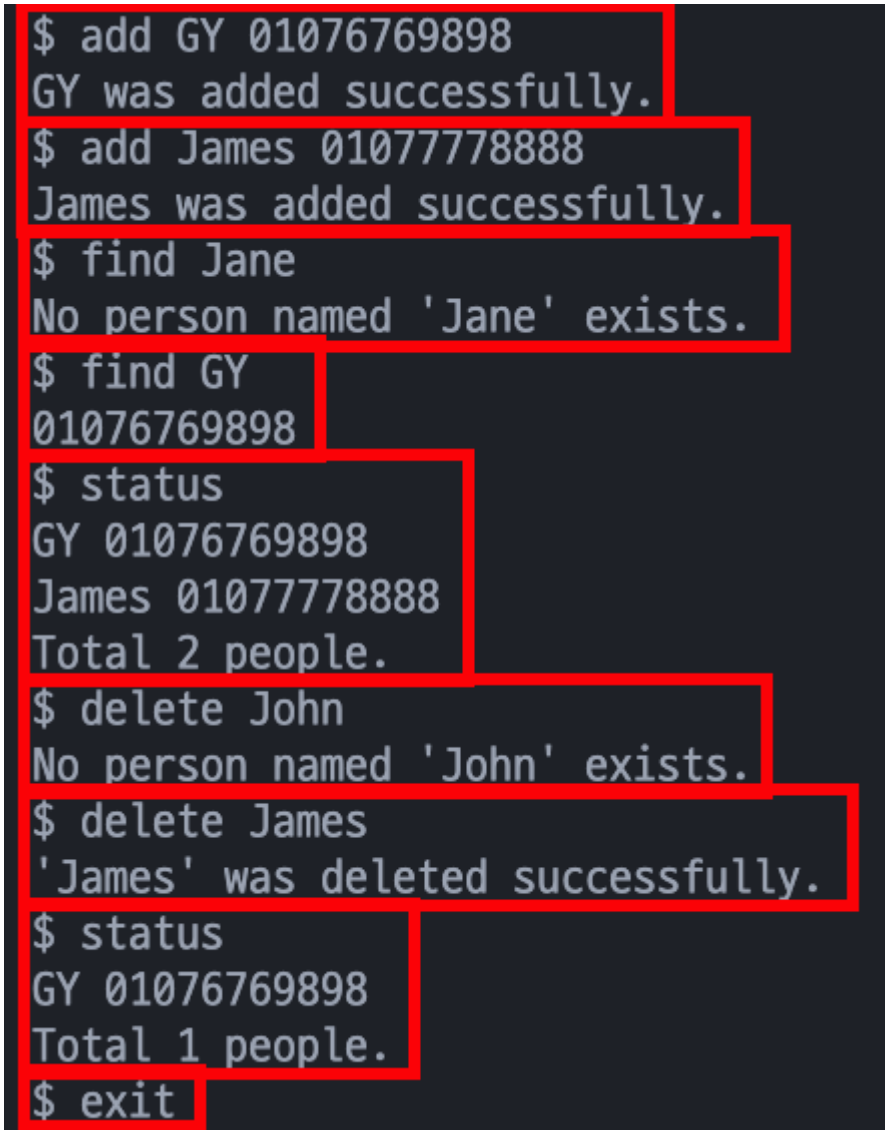
Using string parsing and array data structure,

we implement a simple phone book app.

We will make four functions to implement a phone book app.

- add()
- find()
- status()
- delete()

Example of completion



add

Add a person's name and phone number to the phone book

```
void add() {
  char buf1[BUFFER_SIZE], buf2[BUFFER_SIZE]; // name, phone number
  /* Fill this blank */
}
```

If you need, you can use “strdup” or “strcpy” function in “string.h”

Strdup

strdup is implemented in “string.h” similar with this code

```
char *strdup(char *s)
{
  char *p;
  p = (char *)malloc(strlen(s)+1);
  if (p != NULL)
    strcpy(p, s);
  return p;
}
```

find

Find phone number in the phone book through the name and print it out

```
void find() {
  char buf[BUFFER_SIZE];

  scanf("%s", buf);

  /* Fill this blank */

  printf("No person named '%s' exists.\n", buf);
}
```

status

It prints the status of all the contents currently stored in the phone book

```
void status() {
  /* Fill this blank */
  printf("Total %d people.\n", n);
}
```

delete

Take the user's name and delete the corresponding phone number from the phone book

```
void delete() {
  char buf[BUFFER_SIZE];
```

```
  scanf("%s", buf);

  /* Fill this blank */
  printf("No person named '%s' exists. \n", buf);
}
```

Answer Code Example

```
#include <stdio.h>
#include <string.h>

#define CAPACITY 100
#define BUFFER_SIZE 20

char *names[CAPACITY];
char *numbers[CAPACITY];
int n = 0;

void add();
void find();
void status();
void delete();

int main() {
  char command[BUFFER_SIZE];

  while (1) {
    printf("$ ");

    scanf("%s", command);

    if (strcmp(command, "add") == 0) {
      add();
        }
    else if (strcmp(command, "find") == 0) {
      find();
        }
    else if (strcmp(command, "status") == 0) {
      status();
        }
    else if (strcmp(command, "delete") == 0) {
      delete();
        }
    else if (strcmp(command, "exit") == 0) {
      break;
        }
        else {
            printf("error \n");
```

```
        }
  }
  return 0;
}

void add() {
  char buf1[BUFFER_SIZE], buf2[BUFFER_SIZE];

  scanf("%s", buf1);
  scanf("%s", buf2);

  names[n] = strdup(buf1);
  numbers[n] = strdup(buf2);
  n++;

  printf("%s was added successfully.\n", buf1);
}

void find() {
  char buf[BUFFER_SIZE];

    scanf("%s", buf);

    for (int i = 0; i < n; i++) {
    if (strcmp(buf, names[i]) == 0) {
      printf("%s\n", numbers[i]);
      return;
    }
  }
  printf("No person named '%s' exists.\n", buf);
}

void status() {
  for (int i = 0; i < n; i++) {
    printf("%s %s\n", names[i], numbers[i]);
    }
  printf("Total %d people.\n", n);
}

void delete() {
  char buf[BUFFER_SIZE];

  scanf("%s", buf);

  for (int i = 0; i < n; i++) {
    if (strcmp(buf, names[i]) == 0) {
      names[i] = names[n - 1];
      numbers[i] = numbers[n - 1];
      n--;
      printf("'%s' was deleted successfully. \n", buf);
      return;
    }
  }
  printf("No person named '%s' exists. \n", buf);
}
```