Structures – (phoneBook)

Write a C program that implements the following three functions:

- – The function **readin()** reads a number of persons' names and their corresponding telephone numbers, passes the data to the caller via the parameter p, and returns the number of names that have entered. The character '#' is used to indicate the end of user input.

- – The function **printPB()** prints the phonebook information on the display. It will print the message "Empty phonebook" if the phonebook list is empty.

- – The function **search()** finds the telephone number of an input name *target*, and then prints the name and telephone number on the screen. If the input name cannot be found, then it will print an appropriate error message. The prototypes of the two functions are given below:

The prototypes of the three functions are given below:

void printPB(PhoneBk *pb, int size);
int readin(PhoneBk *pb);
void search(PhoneBk *pb, int size, char *target);

```c
#include <stdio.h>
#include <string.h>
#define MAX 100
typedef struct {
  char name[20];
  int telno;
} PhoneBk;
void printPB(PhoneBk *pb, int size);
int readin(PhoneBk *pb);
void search(PhoneBk *pb, int size, char *target);
int main()
{
  PhoneBk s[MAX];
  char t[20], *p;
  int size=0, choice;

  char dummychar;

  printf("Select one of the following options: \n");
  printf("1: readin()\n");
  printf("2: search()\n");
  printf("3: printPB()\n");
  printf("4: exit()\n");
  do {
    printf("Enter your choice: \n");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
          scanf("%c", &dummychar);
          size = readin(s);
          break;
        case 2:
          scanf("%c", &dummychar);
          printf("Enter search name: \n");
          fgets(t, 20, stdin);
          if (p=strchr(t,'\n')) *p = '\0';
          search(s,size,t);
          break;
        case 3:
          printPB(s, size);
          break;
      }
  } while (choice < 4);
  return 0;
}
```

2

```c
void printPB(PhoneBk *pb, int size)
{
  int i;

  printf("The phonebook list: \n");
  if (size==0)
    printf("Empty phonebook\n");
  else {
    for (i=0; i<size; i++) {
      printf("Name: %s\n", (pb+i)->name);
      printf("Telno: %d\n", (pb+i)->telno);
    }
  }
}
```

```c
int readin(PhoneBk *pb)
{
  int size=0;
  char *p, dummy[80];

  while (1) {
    printf("Enter name: \n");
    fgets(pb->name, 20, stdin);
    if (p=strchr(pb->name,'\n')) *p = '\0';
    if (strcmp(pb->name,"#")==0)
      break;
    printf("Enter tel: \n");
    scanf("%d",&(pb->telno));
    fgets(dummy, 80, stdin);
    pb++;
    size++;
  }
  return size;
}
```

```c
void search(PhoneBk *pb, int size, char *target)
{
  int i;

  for (i=0;i<size;i++,pb++){
    if (strcmp(pb->name,target)==0){
      printf("Name = %s, Tel = %d\n",target,pb->telno);
      break;
    }
  }
  if (i==size)
    printf("Name not found!\n");
}
```