Write the C code to implement the program from the following description.

Employee Details: This program will use an **employee** struct to store and work with **employee** data. Each employee has a **name**, an **emp_id** (integer), and a **salary** (float). Write a program that lets the user enter details for one array of **employee**, and then prints the **employees** out to the Terminal.

a. Declare the employee structure (5)
b. Write a read_employee function that reads in the name, id, and salary and stores these in an employee variable which is then returned. Add a loop to ensure that salary is 0 or larger, asking the user to re-enter the value while it is not valid. (5)
c. Write a read_all_employee function that receives an array of employee values in one parameter and size of the array in another parameter. This function calls read_employee function and stores the result in each index of employee array. (5)
d. Write a print_employee function that accepts an employee parameter and prints the details of this employee to the terminal. Write the "name (emp_id): salary" followed by "Level A" if the salary is less than 4000, or "Level B" if the salary is greater than 5000. (5)
e. Write a print_all_employee function that receives an array of employee values in one parameter and size of the array in another parameter. This function calls print_employee function passing employee parameter in each index. (5)
f. Write an employee_total_salary function that accepts an array of employee values in one parameter and size of array in another parameter, and returns the sum of all of their salaries.(5)
g. Write a search_largest_salary_index function that accepts an array of employee values in one parameter and the size of the array in the second parameter. The function will search the array for the largest salary and return its index (position in the array).(5)
h. Write a search_an_employee_salary function that accepts an array of employee values in one parameter, the size of the array in the second parameter and the name to search for in the third parameter. The function will search the array for an employee with the matching name and return its index (position in the array). If the employee is not found the function should return -1.(5)
i. Write a main function that implements the following pseudocode.

```
use #define SIZE 3
function : main(10)
-----------------------
Local variables:
- emp_array (an array of 3 employee detail values)
- i (an integer used as the index for the arrays)
- char str[20] to read in name of employee for search
-----------------------
1: call read_all_employee, passing in emp_array and SIZE
2: Print the message 'Employee details are'
3: call print_all_employee, passing in emp_array and SIZE
4: Print 'Total : ', employee_total_salary (emp_array, SIZE)
5: Print the message '—Employee with the largest salary is --'
6: Store in i, the search_largest_salary_index passing in emp_array and SIZE
7: Call print_employee, passing in emp_array at index i
8: Print the message '— Enter employee name for the search--'
9: read in the name in str array
10: Store in i, the search_an_employee_salary passing in emp_array, SIZE and str
11: if something was found
12:    Print the message 'The salary of xxxx is xxxx'
13: else
14:    Print the message "Array does not contain an employee named xxxx"
15: Print the message '—Employee details in reverse order are --'
16: Loop i starting from 2 to 0 for each index of emp_array
17: Call print_employee, passing in emp_array at index i
```