```c
#include <stdio.h>
#include <ctype.h>

#define NAME_LENGTH 100
#define MARITAL_STATUS 10

struct Employee
{
    char name[NAME_LENGTH];
    int age;
    char sex;
    float salary;
    char marital_status;
};

int main(void)
{
    //function prototype
    void MyGetString(char[], int);

    //variable delarations
    struct Employee *pEmployeeRecord = NULL;
    int num_employees, i;

    //code
    printf("\n\n");
    printf("Enter Number Of Employees Whose Details You Want To Record : ");
    scanf("%d", &num_employees);

    printf("\n\n");
    pEmployeeRecord = (struct Employee *)malloc(sizeof(struct Employee) *
      num_employees);
    if (pEmployeeRecord == NULL)
    {
        printf("FAILED TO ALLOCATED MEMORY FOR %d EMPLOYEES !!! EXITTING NOW ...
          \n\n", num_employees);
        exit(0);
    }
    else
        printf("SUCCESSFULLY ALLOCATED MEMORY FOR %d EMPLOYEES !!!\n\n",
          num_employees);

    // ****** USER INPUT INITIALIZATION OF ARRAY OF 'struct Employee' ******
    for (i = 0; i < num_employees; i++)
    {
        printf("\n\n\n\n");
        printf("********** DATA ENTRY FOR EMPLOYEE NUMBER %d **********\n", (i +
          1));

        printf("\n\n");
        printf("Enter Employee Name : ");
        MyGetString(pEmployeeRecord[i].name, NAME_LENGTH);
```

```
49
50          printf("\n\n\n");
51          printf("Enter Employee's Age (in years) : ");
52          scanf("%d", &pEmployeeRecord[i].age);
53
54          printf("\n\n");
55          printf("Enter Employee's Sex (M/m For Male, F/f For Female) : ");
56          pEmployeeRecord[i].sex = getch();
57          printf("%c", pEmployeeRecord[i].sex);
58          pEmployeeRecord[i].sex = toupper(pEmployeeRecord[i].sex);
59
60          printf("\n\n\n");
61          printf("Enter Employee's Salary (in Indian Rupees) : ");
62          scanf("%f", &pEmployeeRecord[i].salary);
63
64          printf("\n\n");
65          printf("Is The Employee Married? (Y/y For Yes, N/n For No) : ");
66          pEmployeeRecord[i].marital_status = getch();
67          printf("%c", pEmployeeRecord[i].marital_status);
68          pEmployeeRecord[i].marital_status = toupper(pEmployeeRecord
              [i].marital_status);
69      }
70
71
72      // *** DISPLAY ***
73      printf("\n\n\n\n");
74      printf("********** DISPLAYING EMPLOYEE RECORDS **********\n\n");
75      for (i = 0; i < num_employees; i++)
76      {
77          printf("*********** EMPLOYEE NUMBER %d **********\n\n", (i + 1));
78          printf("Name           : %s\n", pEmployeeRecord[i].name);
79          printf("Age            : %d years\n", pEmployeeRecord[i].age);
80
81          if (pEmployeeRecord[i].sex == 'M')
82              printf("Sex            : Male\n");
83          else if(pEmployeeRecord[i].sex == 'F')
84              printf("Sex            : Female\n");
85          else
86              printf("Sex            : Invalid Data Entered\n");
87
88
89          printf("Salary         : Rs. %f\n", pEmployeeRecord[i].salary);
90
91          if (pEmployeeRecord[i].marital_status == 'Y')
92              printf("Marital Status : Married\n");
93          else if (pEmployeeRecord[i].marital_status == 'N')
94              printf("Marital Status : Unmarried\n");
95          else
96              printf("Marital Status : Invalid Data Entered\n");
97
98
99          printf("\n\n");
```

```
100     }
101
102     if (pEmployeeRecord)
103     {
104         free(pEmployeeRecord);
105         pEmployeeRecord = NULL;
106         printf("MEMORY ALLOCATED TO %d EMPLOYEES HAS BEEN SUCCESSFULLY FREED !!!
              \n\n", num_employees);
107     }
108
109     return(0);
110 }
111
112 // *** SIMPLE RUDIMENTARY IMPLEMENTATION OF gets_s() ***
113 // *** IMPLEMENTED DUE TO DIFFERENT BEHAVIOUR OF gets_s() / fgets() / fscanf() ON
       DIFFERENT PLATFORMS ***
114 // *** BACKSPACE / CHARACTER DELETION AND ARROW KEY CURSOR MOVEMENT NOT
      IMPLEMENTED ***
115
116 void MyGetString(char str[], int str_size)
117 {
118     //variable declarations
119     int i;
120     char ch = '\0';
121
122     //code
123     i = 0;
124     do
125     {
126         ch = getch();
127         str[i] = ch;
128         printf("%c", str[i]);
129         i++;
130     }while ((ch != '\r') && (i < str_size));
131
132     if (i == str_size)
133         str[i - 1] = '\0';
134     else
135         str[i] = '\0';
136 }
137
138
```