20th Day Assessment Abhirami

```
/*Modify the above program that uses dynamic memory allocation to store employee details and
employs recursion for specific tasks:
Input employee details dynamically using malloc or calloc.
Calculate the gross salary for each employee using the formula: Gross Salary=Basic Salary+DA
Amount+HRA Amount
Use recursion to:
Display the details of all employees.
Search for an employee by their empID and display their details.
Free all dynamically allocated memory after the program is executed.
*/
#include <stdio.h>
#include<string.h>
#include<stdlib.h>
struct Employee{
   int empID;
   char name[50];
   float basicSalary;
   float DA;
   float HRA;
   float grossSalary;
};

void calculateGrossSalary(struct Employee*emp);
void searchEmployeeByID(struct Employee*employees,int count,int empID,int index);
void displayEmployees(struct Employee*employees,int count,int index);
int main()
{

   int n,empIDToSearch;
   printf("enter the number of employees:");
   scanf("%d",&n);
   struct Employee *employees =(struct Employee *)malloc(n*sizeof(struct Employee));
   if(employees==NULL) {
      printf("Memory allocation failed\n");
      return 1;
   }

   for(int i=0;i<n;i++){
      printf("\n enter details of employee:%d\n",i+1);
      printf("\n Employee ID");
      scanf("%d",&employees[i].empID);
```

```
        printf("Name:");
        scanf(" %[^\n]",employees[i].name);
        printf("Basic salary:");
        scanf("%f",&employees[i].basicSalary);
        calculateGrossSalary(&employees[i]);
    }
    printf("\n employee details and gross salary:\n");
    printf("ID\tName\tBasicSalary\tDA\tHRA\tGross Salary\n");
    printf("\n");

    for(int i=0;i<n;i++){

printf("%d\t%s\t%.2f\t\t%.2f\t%.2f\t%.2f\n",employees[i].empID,employees[i].name,employees[i].
basicSalary,employees[i].DA,employees[i].HRA,employees[i].grossSalary);

}
printf("====\n");
displayEmployees(employees,n,0);

printf("====\n");
printf("\nenter the emp ID to search");
scanf("%d",&empIDToSearch);
searchEmployeeByID(employees,n,empIDToSearch,0);
free(employees);
printf("\n memory freed and program terminated\n");
    return 0;
}
void calculateGrossSalary(struct Employee*emp){
    emp->DA=emp->basicSalary*0.15;
    emp->HRA = emp->basicSalary*0.04;
    emp ->grossSalary =emp->basicSalary +emp->DA+emp->HRA;
}
void searchEmployeeByID(struct Employee*employees,int count,int empID,int index){
    if(index ==count) {
        printf("Employee with Id %d not found\n",empID);
        return;
    }
    if(employees[index].empID==empID) {
        printf("ID:%d\nName:%s\nBasic Salary:%.2f\nDA:%.2f\nHRA:%.2f\nGross
Salary:%.2f\n",employees[index].empID,employees[index].name,employees[index].basicSalary,
employees[index].DA,employees[index].HRA,employees[index].grossSalary);
        return;
    }
    searchEmployeeByID(employees,count,empID,index+1);
```

```
}
void displayEmployees(struct Employee*employees,int count,int index){
    if(index==count){
        return;

    }

printf("%d\t%s\t%.2f\t\t%.2f\t%.2f\t%.2f\n",employees[index].empID,employees[index].name,em
ployees[index].basicSalary,employees[index].DA,employees[index].HRA,employees[index].gros
sSalary);
    displayEmployees(employees,count,index+1);

}
```