

Solution of DW N° 10 with C language

1) Write a procedure that allows you to create a file of employee:” **employeefile.txt**”

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
typedef struct employee {
    int idnumber;
    char surname[30];
    char firstname [30];
    char grade [50];
    float salary;
};
typedef FILE *fiemployee;

// Procedure: create a file
void create (fiemployee fe){    //void create (FILE *fe){
    employee emp;
    fe = fopen("employeeFile.txt", "w"); // "w" pour la création et "a" pour l'ajout
    if (fe == NULL)    printf("Le fichier employeeFile.txt n'a pas pu s'ouvrir\n");
    else {
        printf("Le fichier employeeFile.txt existe\n");
        printf("Give the identifier of the first employee \n");
        scanf("%d",&emp.idnumber);
        while(emp.idnumber!=0){
            printf("Give the surname of the  employee \n");
            scanf("%s",&emp.surname);
            printf(" give the first of the  employee \n");
            scanf("%s",&emp.firstname);
            printf("Give the grade of the  employee \n");
            scanf("%s",&emp.grade);
            printf("Give the salary of the  employee \n");
            scanf("%f",&emp.salary);
            fprintf(fe,"\n %d %s %s %s %s %f",emp.idnumber,emp.surname,emp.firstname,emp.grade,emp.salary);
            printf("Give the identifier of the next employee \n");
            scanf("%d",&emp.idnumber);
        }
        fclose(fe);
    }
}

int main(){
    FILE *feGlobal;
    create(feGlobal);
}
```

Algorithm

Decalartion

Type employee =record

```
IDNumber:integer;
Surname:c.c;
Firstname:c.c;
Grade : c.c;
Salary : real;
Endrecord
```

Type fi-employee:file;

Procedure create(var fe:fi-employee),

Var em:employee;

Begin

// you can add here these instructions

//Assign("fe,employeefile");

//open(fe,"w")

//or you can follow these steps

fe←fopen("employeefile.txt","w");//open the file in write mode

write("give the id number of the first employee \n");

read(em.idnumber);

while(idnumber <>0) do

write("give the surname of the first employee \n");

read(em.surname);

write("give the firstname of the first employee \n");

read(em.firstname);

write("give the grade of the first employee \n");

read(em.grade);

write("give the salary of the first employee \n");

read(em.salary); //at this step all the information are saved in RAM memory

fwrite(fe, em);//save this information in txt file.

write("give the id number of the next employee \n");

read(em.idnumber);

endwhile //

fclose(fe);

end.

2) Write a procedure that allows you to display the content of this file

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

typedef struct employee {

int idnumber;

char surname[30];

char firstname [30];

char grade [50];

float salary;

};

typedef FILE *fiemployee;

void display(fiemployee fe){

struct employee emp;

fe = fopen("D:\\2023-2024\\Algo 2\\programmes

C\\createfileanddisplay\\employeefile.txt","r");//"r" pour la lecture

if (fe == NULL) {

printf ("The file can not be opened \n");

}

while(!feof(fe)) {

fscanf(fe,"\\n %d %s %s %s %f ", &emp.idnumber, emp.surname,

emp.firstname, emp.grade, &emp.salary);

printf ("\\n %d %s %s %s %f ", emp.idnumber, emp.surname, emp.firstname,

emp.grade, emp.salary);

}

```

        fclose(fe);
    }
int main(){
    FILE *feGlobal;
    display(feGlobal);
}
Procedure display(fe:fi-employee),
Var em:employee;
Begin
fe←fopen("employeefile.txt","r");//open the file in read mode
if (fe= NULL ) then write (" the file can't be open \n");
exit();
while(!feof(fe))do //eof :end of file is an integer value witch takes the value 1 in the end of file
fread(fe, em.idnumber, em.surname, em.firstname, em.salary, em.grade);
write (em.idnumber, em.surname, em.firstname, em.salary, em.grade);
endwhile;
fclose(fe);
end.

```