Batch: C1-1	**Roll No.: 16010123012**
Experiment / assignment / tutorial No. 7	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: Write a program in C to demonstrate use of structures and union.

AIM: Write a program to manage an employee database using structure and union in C. Each Employee has the following information:

1. Employee ID(integer)
2. Name(string)
3. Department(string)
4. Salary(float)

You need to implement the following functionalities:

1. Create a structure named Employee with the appropriate data members to store the information mentioned above.
2. Create a union named EmployeeInfo that can hold either the Name or Department information.
3. Write a function addEmployee that takes user input for each employee's information and stores it in an array of structures.
4. Write a function printEmployeeDetails that takes an employee's ID as input and prints all available details for that employee.
5. Write a function updateEmployeeInfo that takes an employee's ID and allows the user to update either the Name or Department information using the EmployeeInfo union.
6. Implement a menu-driven program that allows the user to perform the above operations. Include options to add a new employee, print employee details, update employee information, and exit the program.

Expected OUTCOME of Experiment:

Design modular programs using functions and the use of structure and union(CO4).

Books/ Journals/ Websites referred:

1. Programming in C, second edition, Pradeep Dey and Manas Ghosh, Oxford University Press.
2. Programming in ANSI C, fifth edition, E Balagurusamy, Tata McGraw Hill.
3. Introduction to programming and problem solving , G. Michael Schneider ,Wiley India edition.

Problem Definition:
The program accepts a choice from the user using a switch case statement and generates output accordingly.

Algorithm:

1. Define a structure Employee to hold employee information like ID, name, department, and salary.
2. Define a union EmployeeInfo to hold either the name or department information.
3. Define a global array employees to store employee data and a global variable numEmployees to keep track of the number of employees.
4. Implement the addEmployee function:
   Check if the maximum number of employees has been reached.
   Prompt the user to enter employee details: ID, name, department, and salary.
   Increment the numEmployees counter.
5. Implement the printEmployeeDetails function:
   Iterate through the employees array to find the employee with the given ID.
   If found, print the employee's details.
6. Implement the updateEmployeeInfo function:
   Iterate through the employees array to find the employee with the given ID.
   Depending on the choice provided by the user:
   Update the employee's name if choice is 1.
   Update the employee's department if choice is 2.
7. In the main function:
   Display a menu-driven interface to the user with options to add an employee, print employee details, update employee information, or exit the program.
   Depending on the user's choice, call the respective function.
   Loop until the user chooses to exit the program.

Implementation details:

```
#include <stdio.h>
#include <string.h>

//maximum number of employees
#define n 10

//structure for Employee
struct Employee {
   int empID;
   char name[50];
   char department[50];
   float salary;
};

//union for EmployeeInfo
union EmployeeInfo {
   char name[50];
   char department[50];
};

// Global array to store employees
struct Employee employees[n];

// Global variable to keep track of number of employees
int numEmployees = 0;

// Function to add a new employee
void addEmployee() {
   if (numEmployees >= n) {
      printf("Maximum number of employees reached.\n");
      return;
   }
   printf("Enter employee ID: ");
   scanf("%d", &employees[numEmployees].empID);

   printf("Enter employee name:");
   scanf("%s",employees[numEmployees].name);

   printf("Enter employee department: ");
   scanf("%s",employees[numEmployees].department);

   printf("Enter employee salary: ");
   scanf("%f", &employees[numEmployees].salary);
```

```
    numEmployees++;
}

// Function to print employee details based on ID
void printEmployeeDetails(int empID) {
    int i;
    for (i = 0; i < numEmployees; i++) {
        if (employees[i].empID == empID) {
            printf("Employee ID: %d\n", employees[i].empID);
            printf("Name: %s\n", employees[i].name);
            printf("Department: %s\n", employees[i].department);
            printf("Salary: %.2f\n", employees[i].salary);
            return;
        }
    }
    printf("Employee with ID %d not found.\n", empID);
}

// Function to update employee information
void updateEmployeeInfo(int empID, int choice, union EmployeeInfo info) {
    int i;
    for (i = 0; i < numEmployees; i++) {
        if (employees[i].empID == empID) {
            switch (choice) {
                case 1:
                    strcpy(employees[i].name, info.name);
                    break;
                case 2:
                    strcpy(employees[i].department, info.department);
                    break;
                default:
                    printf("Invalid choice.\n");
                    return;
            }
            printf("Employee information updated successfully.\n");
            return;
        }
    }
    printf("Employee with ID %d not found.\n", empID);
}

int main() {
    int choice, empID;
```

```
union EmployeeInfo info;
printf("Aaryan Sharma\n16010123012");

do {
    printf("\nEmployee Database Management System\n");
    printf("1. Add Employee\n");
    printf("2. Print Employee Details\n");
    printf("3. Update Employee Information\n");
    printf("4. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
            addEmployee();
            break;
        case 2:
            printf("Enter employee ID: ");
            scanf("%d", &empID);
            printEmployeeDetails(empID);
            break;
        case 3:
            printf("Enter employee ID: ");
            scanf("%d", &empID);
            printf("Which information do you want to update?\n");
            printf("1. Name\n");
            printf("2. Department\n");
            printf("Enter your choice: ");
            scanf("%d", &choice);
            switch (choice) {
                case 1:
                    printf("Enter new name: ");
                    scanf("%s", info.name);
                    updateEmployeeInfo(empID, 1, info);
                    break;
                case 2:
                    printf("Enter new department: ");
                    scanf("%s", info.department);
                    updateEmployeeInfo(empID, 2, info);
                    break;
                default:
                    printf("Invalid choice.\n");
                    break;
            }
```

```
            break;
        case 4:
            printf("Exiting program.\n");
            break;
        default:
            printf("Invalid choice.\n");
            break;
    }
  } while (choice != 4);

  return 0;
}
```

Output(s):

```
Aaryan Sharma
16010123012
Employee Database Management System
1. Add Employee
2. Print Employee Details
3. Update Employee Information
4. Exit
Enter your choice: 1
Enter employee ID: 123
Enter employee name:Aryan
Enter employee department: comp
Enter employee salary: 1000000

Employee Database Management System
1. Add Employee
2. Print Employee Details
3. Update Employee Information
4. Exit
Enter your choice: 1
Enter employee ID: 888
Enter employee name:aduitya
Enter employee department: IT
Enter employee salary: 20000

Employee Database Management System
1. Add Employee
2. Print Employee Details
3. Update Employee Information
4. Exit
Enter your choice: 3
Enter employee ID: 888
Which information do you want to update?
1. Name
2. Department
Enter your choice: 1
Enter new name: Aditya
Employee information updated successfully.
```

```
Employee Database Management System
1. Add Employee
2. Print Employee Details
3. Update Employee Information
4. Exit
Enter your choice: 2
Enter employee ID: 369
Employee with ID 369 not found.

Employee Database Management System
1. Add Employee
2. Print Employee Details
3. Update Employee Information
4. Exit
Enter your choice: 2
Enter employee ID: 123
Employee ID: 123
Name: Aryan
Department: comp
Salary: 1000000.00

Employee Database Management System
1. Add Employee
2. Print Employee Details
3. Update Employee Information
4. Exit
Enter your choice: 4
Exiting program.

Process returned 0 (0x0)   execution time : 95.098 s
Press any key to continue.
```

Conclusion:
We have successfully Performed this experiment and learnt to use structures and unions.

Post Lab Descriptive Questions

- WAP to accept student name, roll number and percentage for 10 students using array of structures and arrange them in descending order of their percentage.

```
#include <stdio.h>
struct student {
   char name[30];
   int rno;
   float pct;
};
int swap(struct student *a,struct student *b) {
   struct student temp=*a;
   *a=*b;
   *b=temp;
   return 1;
}
int sort(struct student arr[], int n) {
   int i,j;
```

```
    for (i=0;i<n-1;i++) {
        for (j=0;j<n-i-1;j++) {
            if (arr[j].pct<arr[j+1].pct) {
                swap(&arr[j+1],&arr[j]);
            }
        }
    }
    return 1;
}
int main() {
    printf("Aaryan Sharma");
    printf("\n16010123012\n");
    struct student arr_student[10];
    int i;
    for (i = 0; i < 10; i++) {
        printf("Enter student name: ");
        scanf("%s", arr_student[i].name);
        printf("Enter student roll no.: ");
        scanf("%d", &arr_student[i].rno);
        printf("Enter percentage of student: ");
        scanf("%f", &arr_student[i].pct);
    }
    sort(arr_student,10);
    printf("Student details in descending order of percentage\n");
    for (i=0;i<10;i++) {
        printf("%d %s %2f\n", arr_student[i].rno,arr_student[i].name,arr_student[i].pct);
    }
}
```

```
Enter student name: Y
Enter student roll no.: 6
Enter percentage of student: 96
Enter student name: U
Enter student roll no.: 7
Enter percentage of student: 36
Enter student name: I
Enter student roll no.: 8
Enter percentage of student: 82
Enter student name: O
Enter student roll no.: 9
Enter percentage of student: 40
Enter student name: P
Enter student roll no.: 10
Enter percentage of student: 84
Student details in descending order of percentage
6 Y 96.000000
1 Q 90.000000
5 T 87.000000
10 P 84.000000
8 I 82.000000
3 E 77.000000
4 R 68.000000
2 W 45.000000
9 O 40.000000
7 U 36.000000

Process returned 0 (0x0)   execution time : 122.855 s
Press any key to continue.
```

- WAP to display employee name, ID and year of experience using union.

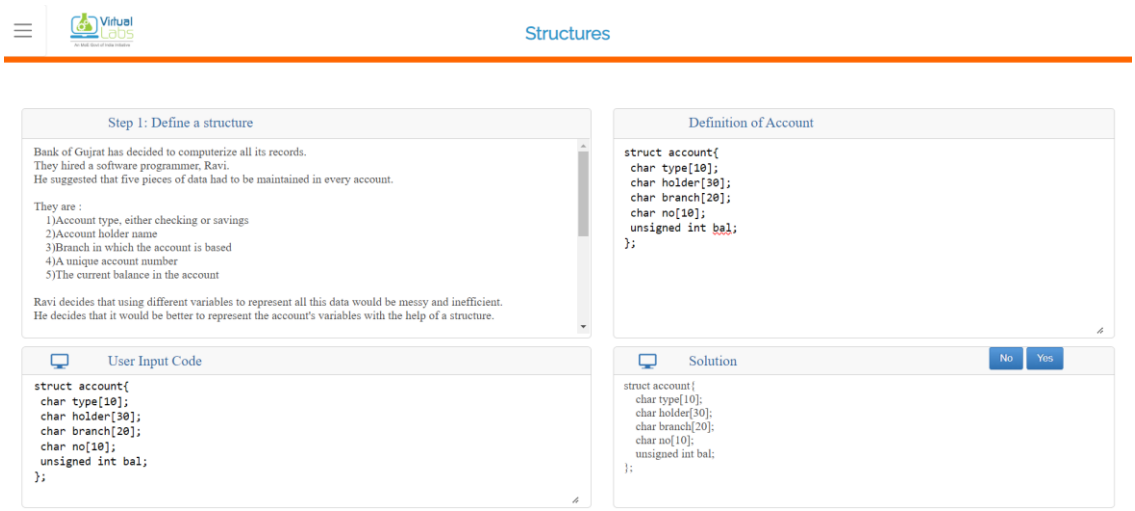
```
#include<stdio.h>

union emp{

    char name[100];

    int id;

    float yoe;
```

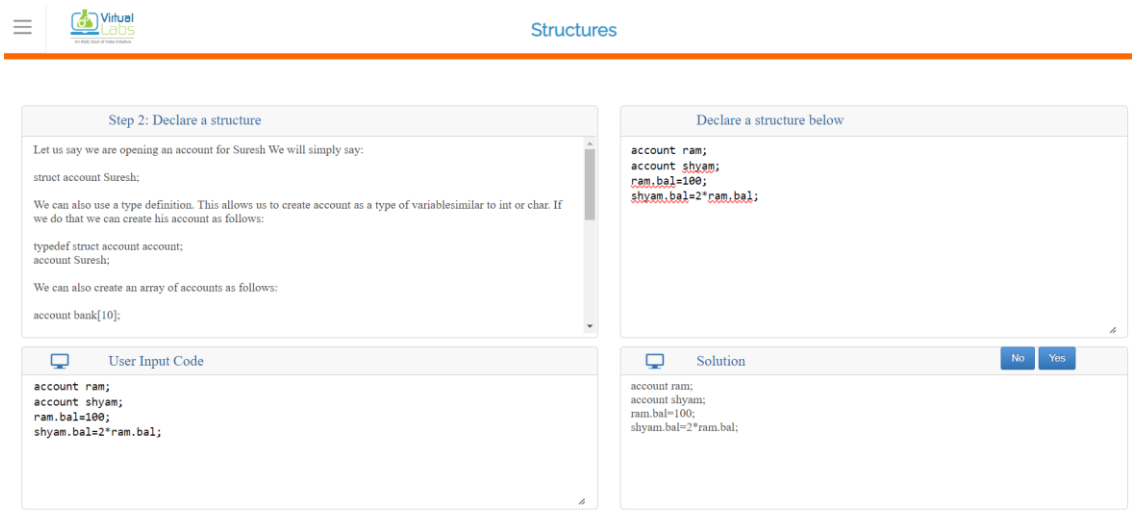
```
 }emp1;
int main(){
printf("Aaryan Sharma\n");
printf("16010123012\n");
printf("Enter employee name :");
scanf("%s",&emp1.name);
printf("Employee name : %s\n",emp1.name);
printf("Enter Employee id :",emp1.id);
scanf("%d",&emp1.id);
printf("Employee Id : %d\n",emp1.id);
printf("Enter employee years of experience : ",emp1.id);
scanf("%f",&emp1.yoe);
printf("Years of experience : %f",emp1.yoe);
}
```

```
Aaryan Sharma
16010123012
Enter employee name :Aaryan
Employee name : Aaryan
Enter Employee id :12
Employee Id : 12
Enter employee years of experience : 1.6667
Years of experience : 1.666700
Process returned 0 (0x0)   execution time : 8.218 s
```

- Virtual lab on Structure and Union

  https://cse02-iiith.vlabs.ac.in/exp/structures/simulation.html





Structures

Step 3: Function to fill a structure

Write a function to fill up an account. It takes the account variables as input and returns an account structure. Now we have to fill up Suresh's account. Insert code to fill up his account as follows.

Account Type = Savings (Only Savings(smallcase) and Current(smallcase) allowed)
Account Name = Suresh
Account Branch = M.G Road, Bangalore
Account Number = 1000000000 (Check if length = 10 characters)
Account Balance = 10000

Function to fill an account

```
account newAc;
strcpy(newAc.holder,name);
strcpy(newAc.branch,branch);
if (strcmp(type,"current")==0 || strcmp(type,"savings")==0);
strcpy(newAc.type,type);
else isErr=1;
if (strlen(number)==10)
strcpy(newAc.no,number);
else isErr=1;
newAc.bal=balance;
if (!isErr)
return (newAc);
return NULL;
```

User Input Code

```
account initAcc(char* name, char* type, char* branch, char* number, unsigned int
balance)
{
int isErr=0;
account newAc;
strcpy(newAc.holder,name);
strcpy(newAc.branch,branch);
if (strcmp(type,"current")==0 || strcmp(type,"savings")==0);
```

Solution

No Yes

```
account initAcc(char* name,char* type,char* branch,char* number,unsigned int balance){
int isErr=0;
account newAc;
strcpy(newAc.holder,name);
strcpy(newAc.branch,branch);
if (strcmp(type,"current")==0 || strcmp(type,"savings")==0);
strcpy(newAc.type,type);
else isErr=1;
if (strlen(number)==10)
```

Structures

Step 4: Use structures to handle data

Let us now write a code to find the details of the person with maximum balance in their account for the following main function..

```
//Previous code
struct account{
   char type[11];
   char holder[31];
   char branch[21];
   char no[11];
   unsigned int bal;
}
typedef struct account account;

//Assume initAcc is defined and following is the prototype
```

Find maximum balance holder

```
account findMaxBal(account src[], int size){
int i=0;
int maxBalIndex=0;
for (;iif (src[i].bal>src[maxBalIndex].bal)
maxBalIndex=i;
}
printf ("maxBalIndex is %d", maxBalIndex);
return src[maxBalIndex];
}
```

User Input Code

```
account findMaxBal(account src[], int size){
int i=0;
int maxBalIndex=0;
for (;iif (src[i].bal>src[maxBalIndex].bal)
maxBalIndex=i;
}
printf ("maxBalIndex is %d", maxBalIndex);
return src[maxBalIndex];
```

Solution

No Yes

```
account findMaxBal(account src[], int size){
int i=0;
int maxBalIndex=0;
for (;iif (src[i].bal>src[maxBalIndex].bal)
maxBalIndex=i;
}
printf ("maxBalIndex is %d", maxBalIndex);
return src[maxBalIndex];
}
```

Virtual Labs

Computer Science and Engineering > Computer programming > Experiments

Aim
Theory
Objective
Pretest
Procedure
Simulation
Posttest
References
Feedback

Structures

1. Which operator is used to access a member of a structure:
- a: .
- b: '
- c: &
- d: >

2. All the functionality of a union can be performed by a structure?
- a: True
- b: False

3. A structure can be defined inside a structure
- a: True
- b: False

4. Default values can be given to the members of the structure :
- a: True
- b: False

5. What will be the ouptut of the following function call: fun4(4,3); where fun4 is defined as:

```
typedef struct S
{
        int x;
        int y;
}S;

S func(S A, S B)
        {
          B.x=A.x;
          B.y=B.y+A.y;
          return B;
        }
        main()
        {
          S a,b;
          a.x=10;
          a.y=20;
                b.x=30;
          b.y=40;
          b=func(a,b);
          printf("%d %d\n",b.x,b.y);
        }
```

- a: 10 60
- b: 20 60
- c: 40 50
- d: 80 30

Submit Quiz

5 out of 5

Virtual Labs
An MoE Govt of India Initiative

Computer Science and Engineering > Computer programming > Experiments

Aim
Theory
Objective
Pretest
Procedure
Simulation
Posttest
References
Feedback

Structures

6. Arrays can be a part of the structure?
- a: True
- b: False

7. One can define an array of structure variables?
- a: True
- b: False

8. The following structure declaration is correct?

```
typedef struct X
{
  int x;
  int y=50;
}X;
```

- a: True
- b: False

9. A structure is a collection of variables under a _____ name .
- a: Single
- b: multiple

Submit Quiz

4 out of 4

Date: 17/03/2024 **Signature of faculty in-charge**