



EXPERIMENT NUMBER -9

STUDENT'S NAME – YANA SRIVASTAVA

STUDENT'S UID – 20BCS2279

CLASS AND GROUP – 23 B

SEMESTER –1st

Practical 9.1: Create a structure hospital and include the following data members:

Name of patient

Patient id (asstatic)

Bloodgroup

Contact number (longlong)

Name of disease

Date of admission

Including the functions to input and print the data for N number of patients.

ALGORITHM

STEP 1:- START

**STEP 2:- Take char name[10]; int patient_id; char group[2]; long long int contact; char
disease[10]; int DOA**

STEP 3:- By using structure

STEP 4:- Print patient name

STEP 5:- Print patient ID

STEP 6:- Print patient blood group

STEP 7:- Print patient contact number

STEP 8:- Print patient disease

STEP 9:- Print patient date of admission

STEP 10:- END

PROGRAM CODE

```
#include <stdio.h>
int i=0, n=1;

//Create the N patient structure

struct Hospital{

    char name[10];

    int patient_id;

    char group[2];

    long long int contact;

    char disease[10];

    int DOA;

}patient[5];

//Function to input patient information

void AddPatient(){

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

        printf("\t Patient name =");

        scanf("%s", patient[i].name);

        printf("\t Patient ID =");

        scanf("%d", &patient[i].patient_id);

        printf("\t Blood Group =");

        scanf("%s", patient[i].group);

        printf("\t Contact=");

        scanf("%lld", &patient[i].contact);
```



```
printf("\t Disease=");

scanf("%s", patient[i].disease);
printf("\t Date of Admission =");

scanf("%d", &patient[i].DOA);

}

}

//Function to output patient information

void PrintPatient(){

printf("Patient Record:\n\n");

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

printf("\t Patient name = %s\n", patient[i].name);

printf("\t Patient ID = %d\n", patient[i].patient_id);

printf("\t Blood Group = %s\n", patient[i].group);

printf("\t Contact = %lld\n", patient[i].contact);

printf("\t Disease = %s\n", patient[i].disease);

printf("\t Date of Admission = %d\n", patient[i].DOA);

}

}

int main()
{

AddPatient();

PrintPatient();

return 0;

}

ERRORS ENCOUNTERED DURING PROGRAM'S EXECUTION
```

(Kindly jot down the compile time errors encountered)

-Errors:0

-Warnings:0

-Output Filename: D:\PROBLEM SOLVING WITH PROGRAMMING\CPROGRAMING\PRATICAL

9.1 exe

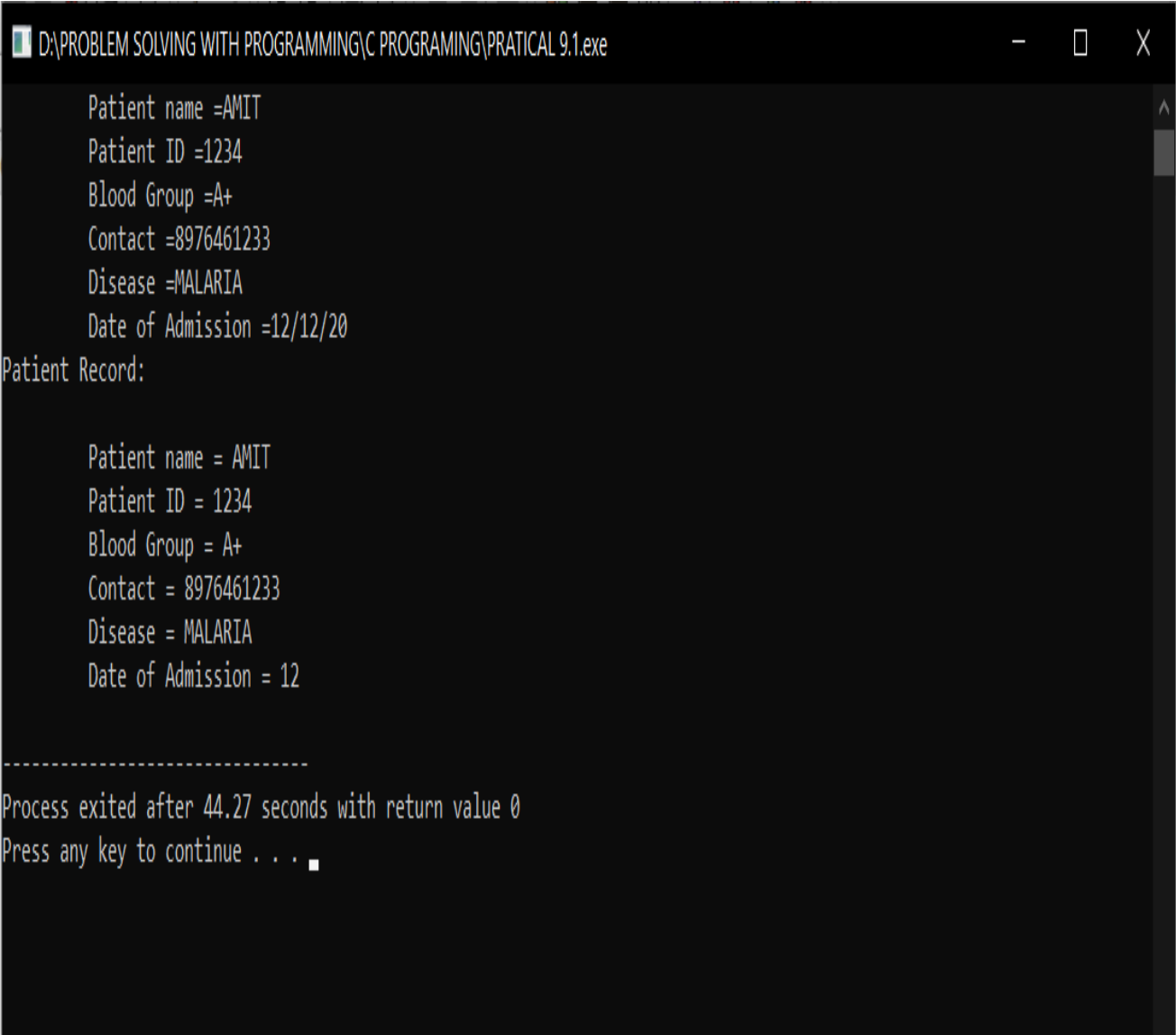
- Output Size: 129.2705078125 KiB

-Compilation Time:0.38s

PROGRAMS' EXPLANATION (in brief)

In this program we use structure hospital Name of patient Patient id (as static) Blood group Contact number (long long) Name of disease Date of admission

OUTPUT



```
D:\PROBLEM SOLVING WITH PROGRAMMING\C PROGRAMING\PRATICAL 9.1.exe

Patient name =AMIT
Patient ID =1234
Blood Group =A+
Contact =8976461233
Disease =MALARIA
Date of Admission =12/12/20
Patient Record:

Patient name = AMIT
Patient ID = 1234
Blood Group = A+
Contact = 8976461233
Disease = MALARIA
Date of Admission = 12

-----
Process exited after 44.27 seconds with return value 0
Press any key to continue . . .
```

AIM OF THE EXPERIMENT –

.Practical 9.2: What will the difference if above program is going to be implemented through union. Implement the same program through union and differentiate the output as well as memory allocation.

ALGORITHM

STEP 1:- START

STEP 2:- Take char name[10]; int patient_id;char group[2]; long long int contact; char disease[10];int DOA

STEP 4:-By implemented through union

STEP 3:- Print patient name

STEP 4:- Print patient ID

STEP 5:- Print patient blood group

STEP 6:-Print patient contact number

STEP 7:-Print patient disease

STEP 8:-Print patient date of admission

STEP 9:- END

PROGRAM CODE

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

```
int i=0, n=1;
```

```
//Create the N patient structure
```

```
union Patient{
```

```
    char name[10];
```

```
    int patient_id;
```

```
    char group[2];
```

```
    long long int contact;
```

```
char disease[10];

int DOA;

}patient[5];

//Function to input patient information

void AddPatient(){

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

printf("\t Patient name =");

scanf("%s", patient[i].name);

printf("\t Patient ID =");

scanf("%d", &patient[i].patient_id);

printf("\t Patient Blood Group =");

scanf("%s", patient[i].group);

printf("\t Patient Contact =");

scanf("%lld", &patient[i].contact);

printf("\t Patient Disease =");

scanf("%s", patient[i].disease);

printf("\t Patient Date of Admission =");

scanf("%d", &patient[i].DOA);

}

}

//Function to output patient information

void PrintPatient(){

printf("Patient Record:\n\n");
```

```
for(i=0; i<n; i++){  
  
    printf("\t Patient name = %s", patient[i].name);  
  
    printf("\t Patient ID = %d", patient[i].patient_id);  
  
    printf("\t Patient Blood Group = %s", patient[i].group);  
  
    printf("\t Patient Contact = %lld", patient[i].contact);  
  
    printf("\t Patient Disease = %s", patient[i].disease);  
  
    printf("\t Patient Date of Admission = %d", patient[i].DOA);  
  
}  
  
}  
  
int main(){  
  
    AddPatient();  
  
    PrintPatient();  
  
    return 0;  
  
}
```

ERRORS ENCOUNTERED DURING PROGRAM'S EXECUTION

(Kindly jot down the compile time errors encountered)

Compilation results...

Errors:0

Warnings:0

Output Filename: D:\PROBLEM SOLVING WITH PROGRAMMING\CPROGRAMING\PRATICAL 9.1.exe

- Output Size: 129.2705078125 KiB

Compilation Time:0.38s

PROGRAMS' EXPLANATION (in brief)

In this program is going to be implemented through union. Implement the same program through union and differentiate the output as well as memory allocation then all patients are different

OUTPUT



```
D:\PROBLEM SOLVING WITH PROGRAMMING\C PROGRAMING\practical 9.2.exe

Patient name =amit
Patient ID =12345
Patient Blood Group =b+
Patient Contact =828726839823
Patient Disease =malaria
Patient Date of Admission =4/5/20
Patient Record:

Patient name = ♦ Patient ID = 4 Patient Blood Group = ♦ Patient Contact = 27419011088121860
Patient Disease = ♦ Patient Date of Admission = 4
-----
Process exited after 50.49 seconds with return value 0
Press any key to continue . . .
```

AIM OF THE EXPERIMENT –

.Practical 9.3: Write a Program for the following statement: You are given task to store records of mothers and fathers of all students of your class in two separate structures mother and father. Each record will contain name, age, work status and height. On the basis of these records you have to print the following:

- 1. How many mothers are working? If more than 70% of women's are working then print women are job oriented.**
- 2. if difference in heights of mother and father is ≥ 10 inches you have to print like rama mismatches sham where ram and sham are names of mother and father.**
- 3. You also have to find average difference in the ages of mother and father**

ALGORITHM

STEP 1:- START

STEP 2:- By using struct mother and struct father

STEP 3:- Enter name, age, height(in inches), work_status of student's mother

STEP 4:- Enter name, age, height(in inches), work_status of student's father

STEP 5:- If more than 70% of women's are working then print women are job oriented.

STEP 6:- if difference in heights of mother and father is ≥ 10 inches you have to print like rama mismatches sham where ram and sham are names of mother and father.

STEP 7:- to find average difference in the ages of mother and father

avg_diff += abs(m[i].height - f[i].height);

STEP 8:- END

PROGRAM CODE

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

```
#include<math.h>
```

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

```
struct mother{
```



```
char name[50],work_status[50];

int age,height;

};

struct father{

char name[50],work_status[50];

int age,height;

};

int main()

{

int n;

printf("Enter total no. of students");

scanf("%d",&n);

struct mother m[5];

struct father f[5];

int i=0;

for(i=0;i<n;i++)
```

{

```
printf("\nEnter name of %d student's mother\n",i+1);  
scanf("%s",m[i].name);  
printf("\nEnter age of %d student's mother\n",i+1);  
scanf("%d",&m[i].age);  
printf("\nEnter height of %d student's mother\n",i+1);  
scanf("%d",&m[i].height);  
printf("\nEnter work_status of %d student's mother\n",i+1);  
scanf("%s",&m[i].work_status);
```

```
printf("\nEnter name of %d student's father\n",i+1);  
scanf("%s",f[i].name);  
printf("\nEnter age of %d student's father\n",i+1);  
scanf("%d",&f[i].age);  
printf("\nEnter height of %d student's father\n",i+1);  
scanf("%d",&f[i].height);  
printf("\nEnter work_status of %d student's father\n",i+1);  
scanf("%s",&f[i].work_status);
```

}

```
int working_womens=0;
```

```
for(i=0;i<n;i++)
```

```
{
```

```
if(!strcmp(m[i].work_status, "working"))
```

```
working_womens++;
```

```
}
```

```
printf("\nNo. of working womens are%d",working_womens);

if((((float)working_womens/n)*100)>70)
    printf("\nWomens are job oriented\n\n");
float avg_diff=0,diff;

for( i=0;i<n;i++)
{
    diff=abs(m[i].height-f[i].height);

    avg_diff+=abs(m[i].height-f[i].height);

    if(diff>=10)

        printf("\n%s mismatches %s\n",m[i].name,f[i].name);

}
avg_diff/=n;

printf("Average difference in the ages of mother and father is %f \n\n",avg_diff);
}
```

ERRORS ENCOUNTERED DURING PROGRAM'S EXECUTION

(Kindly jot down the compile time errors encountered)

Compilation results...

Errors:0

Warnings:0

Output Filename: D:\PROBLEM SOLVING WITH PROGRAMMING\CPROGRAMING\PRATICAL
9.1 exe

- Output Size: 129.2705078125 KiB

Compilation Time:0.38s

PROGRAMS' EXPLANATION (in brief)

In this program to store records of mothers and fathers of all students of your class in two separate structures mother and father. Each record will contain name, age, work status and height

OUTPUT

```
D:\PROBLEM SOLVING WITH PROGRAMMING\C PROGRAMING\practical 9.3.exe
Enter total no. of students1

Enter name of 1 student's mother
pooja

Enter age of 1 student's mother
23

Enter height of 1 student's mother
56

Enter work_status of 1 student's mother
working

Enter name of 1 student's father
amit

Enter age of 1 student's father
25

Enter height of 1 student's father
70

Enter work_status of 1 student's father
working

No. of working womens are1
Womens are job oriented

pooja mismatches amit
Average difference in the ages of mother and father is 14.000000

-----
Process exited after 55.87 seconds with return value 0
Press any key to continue . . .
```

AIM OF THE EXPERIMENT –

.Practical 9.4: Store N student records including fields name, First MST marks .Second MST marks ,lecture attended ,Lecture delivered , If student got average marks of two MST ≥ 80 ,He/She will be eligible for taking exams without taking care of attendance otherwise attendance should be $\geq 75\%$.Print name of all those students who are detained from final exams .

ALGORITHM

STEP 1:- START

STEP 2:- Enter total no of student

STEP 3:- By using struct student

STEP 4:-Enter name of student

STEP 5:-Enter mst1 and mst2marks

STEP 6:-Enter total lecture and attended lectures

STEP 7:-Calculate average mst marks

```
avg_marks= (s[i].mst1_marks+s[i].mst2_marks)/2;
```

STEP 8:-Calculate attendance

```
attendance_pre= (float)s[i].attended_lect/s[i].total_lect;
```

STEP 9:- if($\text{avg_marks} < 80.0f$ && $\text{attendance_pre} < 0.75f$)

STEP 10:-if(flag==1) then print list of detained students

STEP 11:- if(flag==0) then print no student detained

STEP 12:- END

PROGRAM CODE

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

```
struct student
```

```
{
```

```
char stu_name[20];
```

```
float mst1_marks,mst2_marks;
```

```
int total_lect, attended_lect;

};

int main()
{

int n,i;

printf("Enter total no of students\n");

scanf("%d",&n);

struct student s[n];

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

printf("Enter name of student\n");

scanf("%s",s[i].stu_name);

printf("Enter mst1 and mst2marks\n");

scanf("%f%f",&s[i].mst1_marks,&s[i].mst2_marks);

printf("Enter total lectures and attended lectures\n");

scanf("%d%d",&s[i].total_lect,&s[i].attended_lect);

}

int flag=0;

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

float avg_marks= (s[i].mst1_marks+s[i].mst2_marks)/2;

float attendance_pre= (float)s[i].attended_lect/s[i].total_lect;
```

```
if(avg_marks<80.0f && attendance_pre<0.75f )  
  
{ flag+=1;  
  
if(flag==1)  
  
printf("List of Detained Students:\n");  
  
printf("%s\n",s[i].stu_name);  
  
}  
  
}  
  
if(flag==0)  
  
printf("No student detained");  
  
return 0;  
  
}
```

ERRORS ENCOUNTERED DURING PROGRAM'S EXECUTION

(Kindly jot down the compile time errors encountered)

NO ERROR

PROGRAMS' EXPLANATION (in brief)

In this program Store N student records including fields name, First MST marks .Second MST marks ,lecture attended ,Lecture delivered , If student got average marks of two MST ≥ 80 ,He/She will be eligible for taking exams without taking care of attendance otherwise attendance should be $\geq 75\%$.Print name of all those students who are detained from final exams .

OUTPUT

LEARNING OUTCOMES

- Identify situations where computational methods would be useful.
- Approach the programming tasks using techniques learnt and write pseudo-code.
- Choose the right data representation formats based on the requirements of the problem.
- Use the comparisons and limitations of the various programming constructs and choose the right one for the task.

EVALUATION COLUMN (To be filled by concerned faculty only)

Sr. No.	Parameters	Maximum Marks	Marks Obtained
1.	Worksheet Completion including writing learning objective/ Outcome	10	
2.	Post Lab Quiz Result	5	



3.	Student engagement in Simulation/ Performance/ Pre Lab Questions	5	
4.	Total Marks	20	