**May 16 Lab questions**

#1. WAP to store n student’s information (i.e. student’s roll no, name,

gender, marks in 5 subjects etc) of an educational institute and display all the

data with total marks of each student, using array of structure. If full mark of

each subject is considered as 100 and pass mark as 40, then display the list of

students failed in a particular subject.

Code:

#include <stdio.h>

typedef struct students{

    char name[50];

    int rollno;

    char gender[6];

    float english;

    float maths;

    float chemistry;

    float physics;

    float programming;

}students;

int main()

{

    int n,i\_285;

    printf("Provide the number of students\n");

    scanf("%d",&n);

    int sum[n];

    students S[n];

    for(i\_285=0;i\_285<n;i\_285++)

    {

        printf("Give the name of student %d\n",i\_285+1);

        scanf("%s",S[i\_285].name);

        printf("Give the roll number\n");

        scanf("%d",&S[i\_285].rollno);

        printf("Give the gender\n");

        scanf("%s",S[i\_285].gender);

        printf("English marks\n");

        scanf("%f",&S[i\_285].english);

        printf("math marks\n");

        scanf("%f",&S[i\_285].maths);

        printf("Chemistry marks\n");

        scanf("%f",&S[i\_285].chemistry);

        printf("physics marks\n");

        scanf("%f",&S[i\_285].physics);

        printf("programming marks\n");

        scanf("%f",&S[i\_285].programming);

    }

    for(i\_285=0;i\_285<n;i\_285++)

    {

        sum[i\_285]=S[i\_285].physics+S[i\_285].chemistry+S[i\_285].maths+S[i\_285].english+S[i\_285].programming;

    }

    for(i\_285=0;i\_285<n;i\_285++)

    {

        printf("Student name= %s\n", S[i\_285].name);

        printf("\n");

        printf("Student roll number= %d\n",S[i\_285].rollno);

        printf("\n");

        printf("Student Gender= %s\n",S[i\_285].gender);

        printf("\n");

        printf("English Result= %2.2f\n",S[i\_285].english);

        printf("\n");

        printf("Chemistry result= %2.2f\n",S[i\_285].chemistry);

        printf("\n");

        printf("Physics Result= %2.2f\n",S[i\_285].physics);

        printf("\n");

        printf("maths result=%2.2f\n",S[i\_285].maths);

        printf("\n");

        printf("programming result= %2.2f\n",S[i\_285].programming);

        printf("\n");

        printf("Total marks is= %2.2f \n",sum);

        printf("\n");

        printf("%s has failed in the following subjects\n",S[i\_285].name);

        if(S[i\_285].chemistry<=40)

        {

            printf("chemistry\n");

        }

        if(S[i\_285].physics<=40)

        {

            printf("Physics\n");

        }

        if(S[i\_285].english<=40)

        {

            printf("English\n");

        }

        if(S[i\_285].maths<=40)

        {

            printf("maths\n");

        }

        if(S[i\_285].programming<=40)

        {

            printf("programming\n");

        }

        printf("=======================================================================================================\n");

        printf("=======================================================================================================\n");

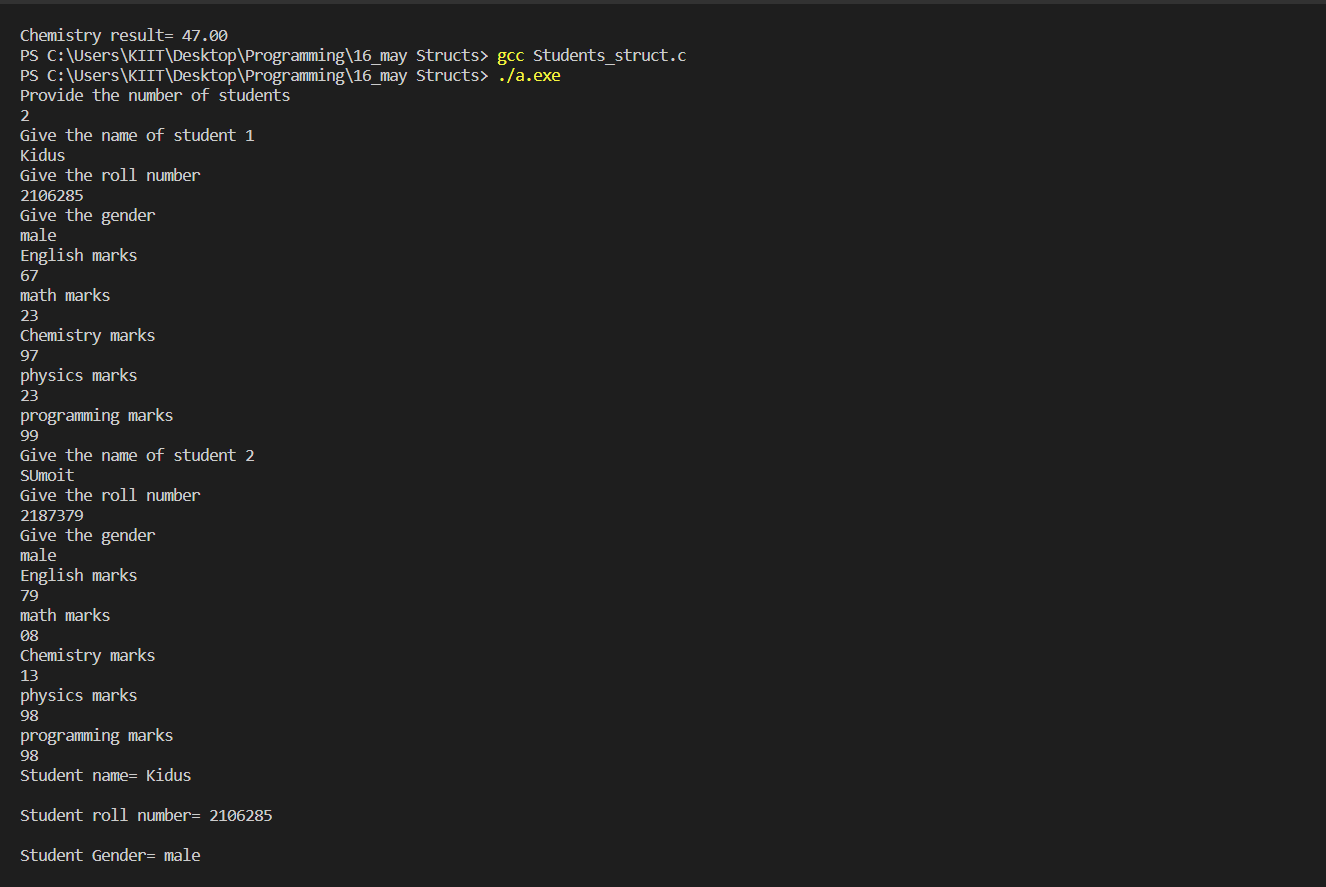
        printf("=======================================================================================================\n");

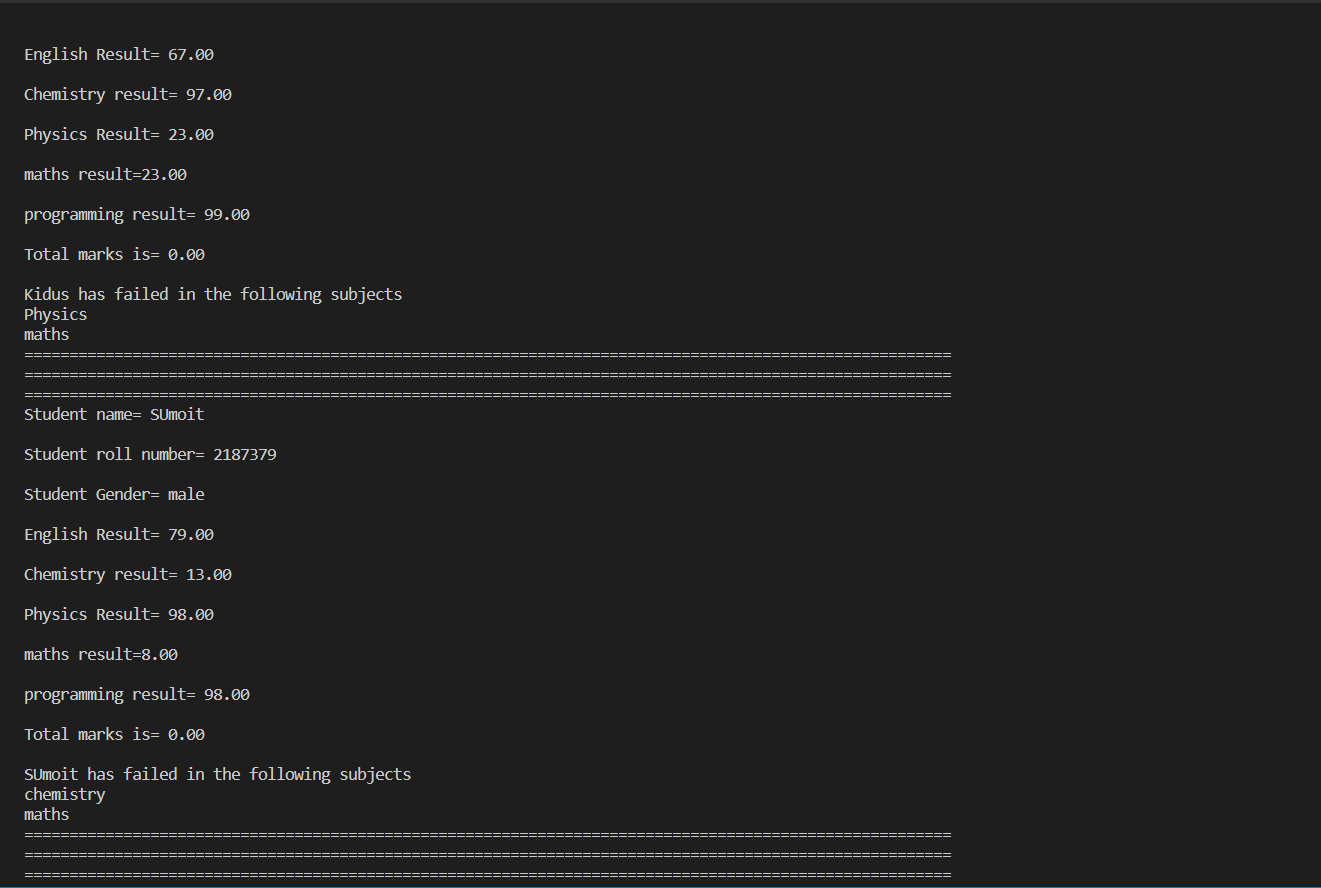
    }

    return 0;

}

Output:





#2. WAP to add two distances (in km-meter) using structures.

Code:

#include <stdio.h>

typedef struct distance{

    int kilometer\_285;

    int meter\_285;

}distance;

int main()

{

    distance s1,s2,s3;

    int km1,km2,m1,m2,KM,M;

    printf("GIve the first diatnce in km and m form respectively\n");

    scanf("%d%d",&km1,&m1);

    printf("GIve the second diatnce in km and m form respectively\n");

    scanf("%d%d",&km2,&m2);

    s1.kilometer\_285=km1;

    s1.meter\_285=m1;

    s2.kilometer\_285=km2;

    s2.meter\_285=m2;

    s3.kilometer\_285=s1.kilometer\_285+s2.kilometer\_285;

    s3.meter\_285=s1.meter\_285+s2.meter\_285;

    if(s3.meter\_285>=1000)

    {

        s3.meter\_285=s3.meter\_285-1000;

        s3.kilometer\_285++;

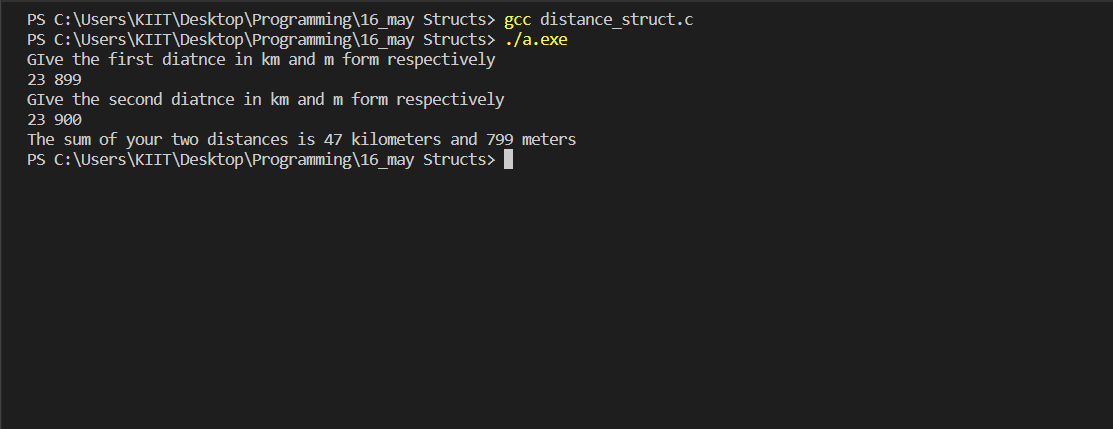
    }

    printf("The sum of your two distances is %d kilometers and %d meters\n",s3.kilometer\_285,s3.meter\_285);

    return 0;

}

Output:



#3. WAP to add two times (in hr-min-sec) by passing structure to a

function.

Code:

#include <stdio.h>

typedef struct time{

    int hour;

    int minute;

    int second;

}time;

int main()

{

    time t1\_285,t2\_285,t3\_285;

    int h1,h2,h3,m1,m2,m3,s1,s2,s3;

    printf("Please give the first timing in hour/minutes/second format\n");

    scanf("%d%d%d",&h1,&m1,&s1);

    printf("Please give the second timing in hour/minutes/second format\n");

    scanf("%d%d%d",&h2,&m2,&s2);

    t1\_285.hour=h1;

    t1\_285.minute=m1;

    t1\_285.second=s1;

    t2\_285.hour=h2;

    t2\_285.minute=m2;

    t2\_285.second=s2;

    t3\_285.hour= t1\_285.hour+t2\_285.hour;

    t3\_285.minute=t1\_285.minute+t2\_285.minute;

    t3\_285.second=t1\_285.second+t2\_285.second;

    if(t3\_285.second>=60)

    {

        t3\_285.second=t3\_285.second-60;

        t3\_285.minute++;

    }

    if(t3\_285.minute>=60)

    {

        t3\_285.minute=t3\_285.minute-60;

        t3\_285.hour++;

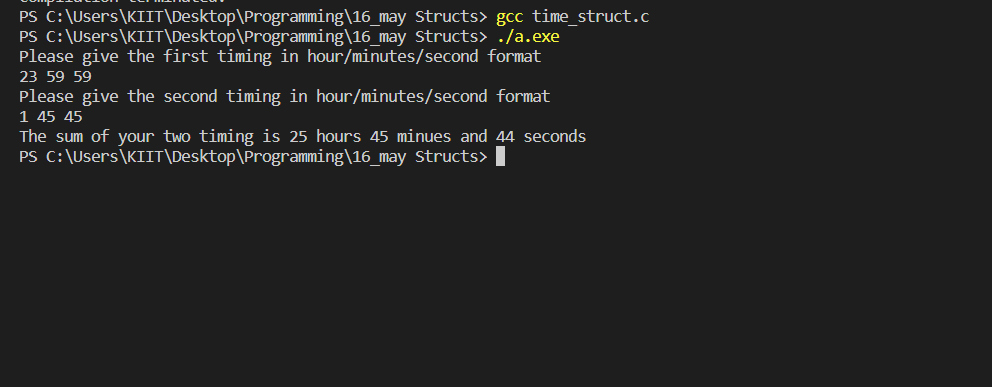
    }

    printf("The sum of your two timing is %d hours %d minues and %d seconds\n",t3\_285.hour,t3\_285.minute,t3\_285.second);

    return 0;

}

Output:



#4. WAP to store n employees data such as employee name, gender, des-

ignation, department, basic pay etc using structures with dynamically mem-

ory allocation. Calculate the gross pay of each employees as follows: Gross

pay=basic pay + HR + DA HR=25

Code:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct employee{

    char name[30];

    char gender[6];

    char designation[30];

    char department[30];

    float basicpay;

}employee;

int main()

{

    employee \*ptr\_285;

    int n,i;

    float HR\_285,DA\_285,gross\_pay;

    printf("Give the number of employees you have\n");

    scanf("%d",&n);

    ptr\_285= (employee\*) malloc(n\*sizeof(employee));

    for(i=0;i<n;i++)

    {

        printf("PLease give the details of employee %d\t",i+1);

        printf("==============\t==============\t============\n");

        printf("name=>\n");

        scanf("%s",&(ptr\_285+i)->name);

        printf("Gender=>\n");

        scanf("%s",&(ptr\_285+i)->gender);

        printf("Designation=>\n");

        scanf("%s",&(ptr\_285+i)->designation);

        printf("Department=>\n");

        scanf("%s",&(ptr\_285+i)->department);

        printf("basic pay=>\n");

        scanf("%f",&(ptr\_285+i)->basicpay);

        printf("==============\t==============\t============\n");

    }

    //assuming the dearness allowance is 20%

    for(i=0;i<n;i++)

    {

        gross\_pay= (ptr\_285+i)->basicpay + 0.25\*(ptr\_285+i)->basicpay + 0.2\*(ptr\_285+i)->basicpay;

        printf("The gross pay of employee %d is=>%f\n",i+1,gross\_pay);

    }

    return 0;

}

Output:

