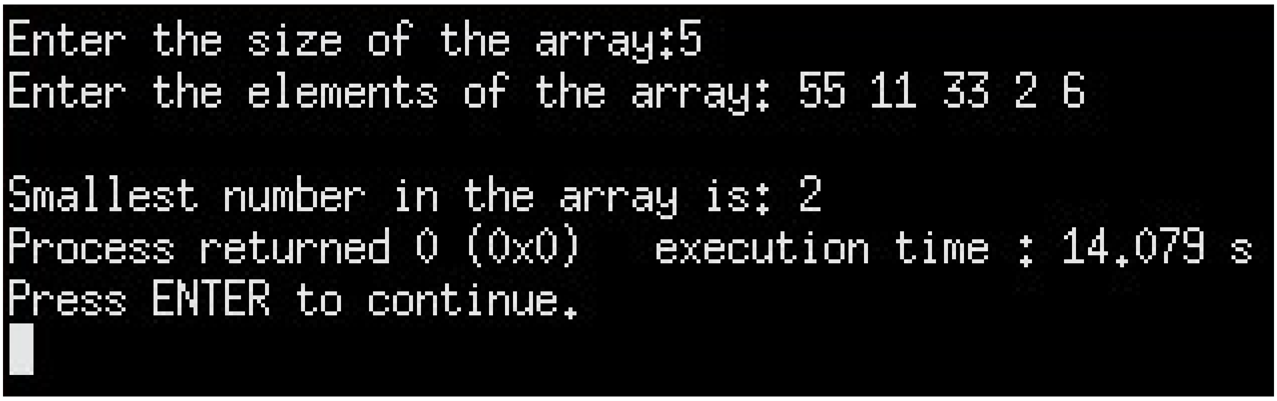
WEEK-1

1: Write a function Smallest to find the smallest element in an array using pointer. Create a dynamically allocated array and read the values from keyboard in main. Display the result in the main function.

Program:

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

int smallest(int *a, int n)
 {
    int sml=*a;
    int i, temp;
    for(i=0; i<n; i++){
        if(*(a+i)<sml){
            sml=*(a+i);
        }
    }
    return sml;
}

int main(){
int *a;
int n, i, result;
printf("Enter the size of the array:");
scanf("%d",&n);
a=(int*)calloc(n,sizeof(int));
printf("Enter the elements of the array: ");
for(i=0;i<n;i++){
    scanf("%d",a+i);
    }
result=smallest(a,n);
printf("\nSmallest number in the array is: %d",result);
}
```

Test Case:



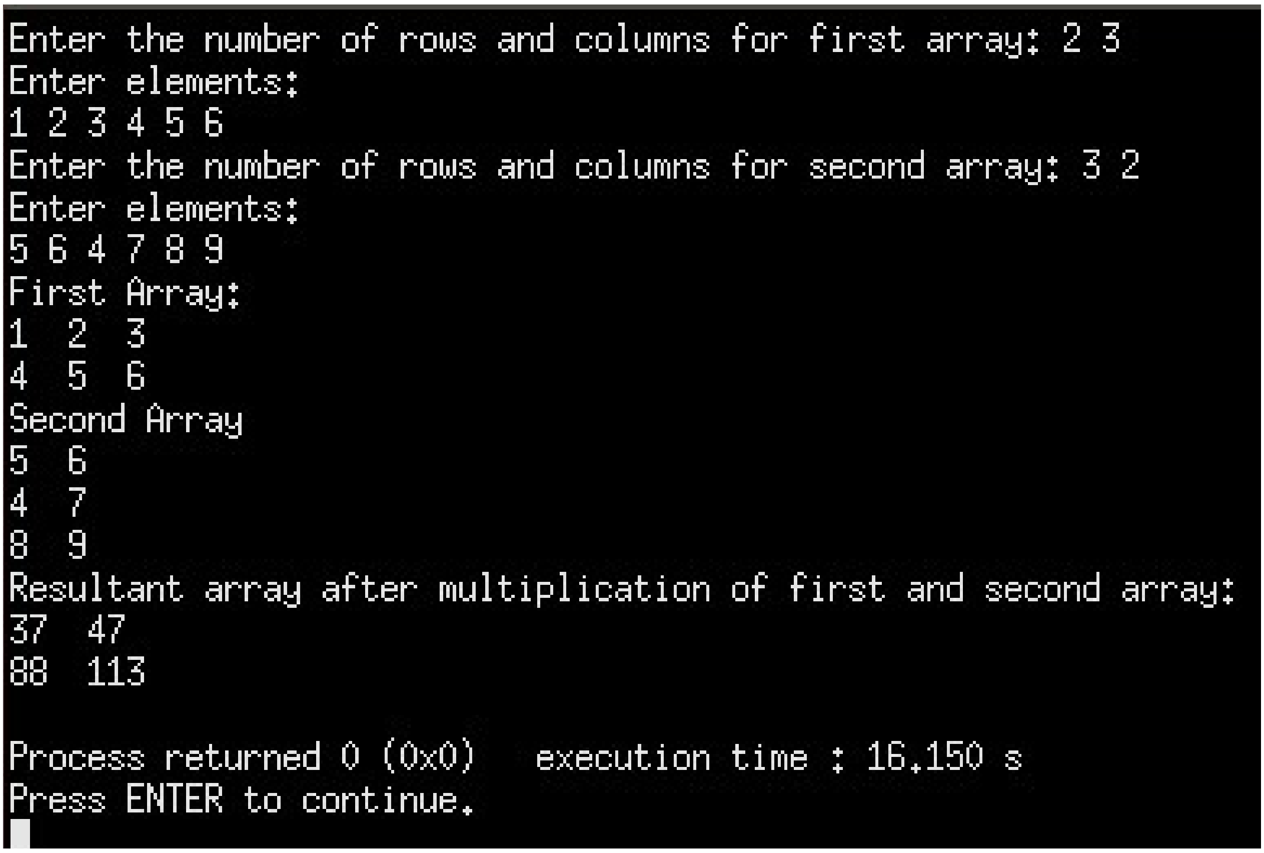
```
Enter the size of the array:5
Enter the elements of the array: 55 11 33 2 6

Smallest number in the array is: 2
Process returned 0 (0x0)   execution time : 14.079 s
Press ENTER to continue.
```

2: Implement a C program to read, display and to find the product of two matrices using functions with suitable parameters.Note that the matrices should be created using dynamic memory allocation functions and the elements are accessed using array dereferencing.

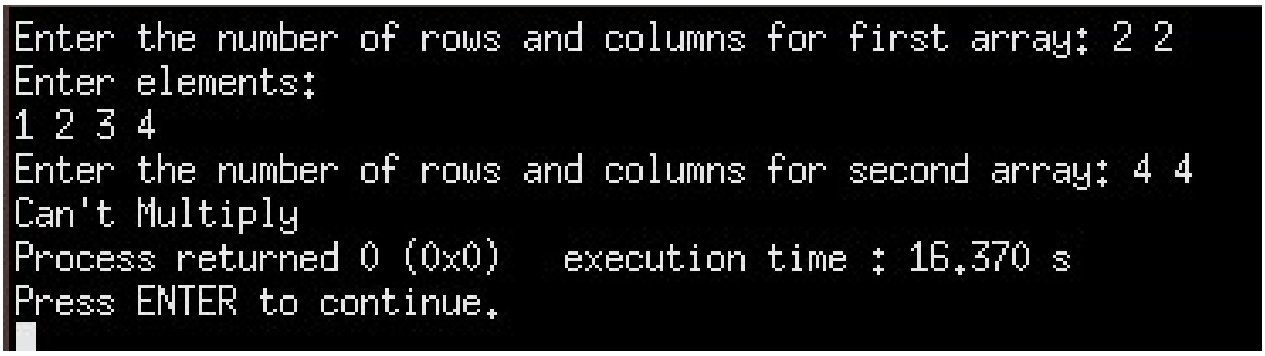
Program:

```
#include<stdio.h>
#include<stdlib.h>
int multiply(int **a,int m,int n,int **b,int p,int q)
{
int **c,i,j,k;
c=(int**)calloc(m,sizeof(int*));
for(i=0;i<m;i++)
{
*(c+i) = (int*)calloc(q,sizeof(int));
for(j=0;j<n;j++)
{
*(*(c+i)+j) = 0;
for(k=0;k<n;k++)
*(*(c+i)+j) += *(*(a+i)+k) * *(*(b+k)+j);
}
}
printf("Resultant array after multiplication of first and second array: \n");
for(i=0;i<m;i++)
{
for(j=0;j<q;j++)
printf("%d  ",*(*(c+i)+j));
printf("\n");
}
}

int main()
{
int n,**arr1,**arr2,m,p,q,i,j;
printf("Enter the number of rows and columns for first array: ");
scanf("%d %d",&m,&n);
arr1 = (int**)calloc(m,sizeof(int*));
printf("Enter elements:\n");
for(i=0;i<m;i++)
{
*(arr1+i) = (int*)malloc(n*sizeof(int));
for(j=0;j<n;j++)
{
scanf("%d",(*(arr1+i)+j))
;
}
}
printf("Enter the number of rows and columns for second array: ");
scanf("%d %d",&p,&q);
if(n!=p)
{
printf("Can't Multiply");
exit(0);
}
arr2 = (int**)calloc(p,sizeof(int*));
printf("Enter elements:\n");
for(i=0;i<p;i++)
{
*(arr2+i) = (int*)malloc(q*sizeof(int));
for(j=0;j<q;j++)
{
scanf("%d",(*(arr2+i)+j));
}
}
printf("First Array:\n");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
printf("%d  ",*(*(arr1+i)+j));
}
printf("\n");
}
printf("Second Array\n");
for(i=0;i<p;i++)
{
for(j=0;j<q;j++)
{
printf("%d  ",*(*(arr2+i)+j));
}
printf("\n");
}
multiply(arr1,m,n,arr2,p,q);
return 0;
}
```

Test Case:



```
Enter the number of rows and columns for first array: 2 3
Enter elements:
1 2 3 4 5 6
Enter the number of rows and columns for second array: 3 2
Enter elements:
5 6 4 7 8 9
First Array:
1  2  3
4  5  6
Second Array
5  6
4  7
8  9
Resultant array after multiplication of first and second array:
37  47
88  113

Process returned 0 (0x0)   execution time : 16.150 s
Press ENTER to continue.
```



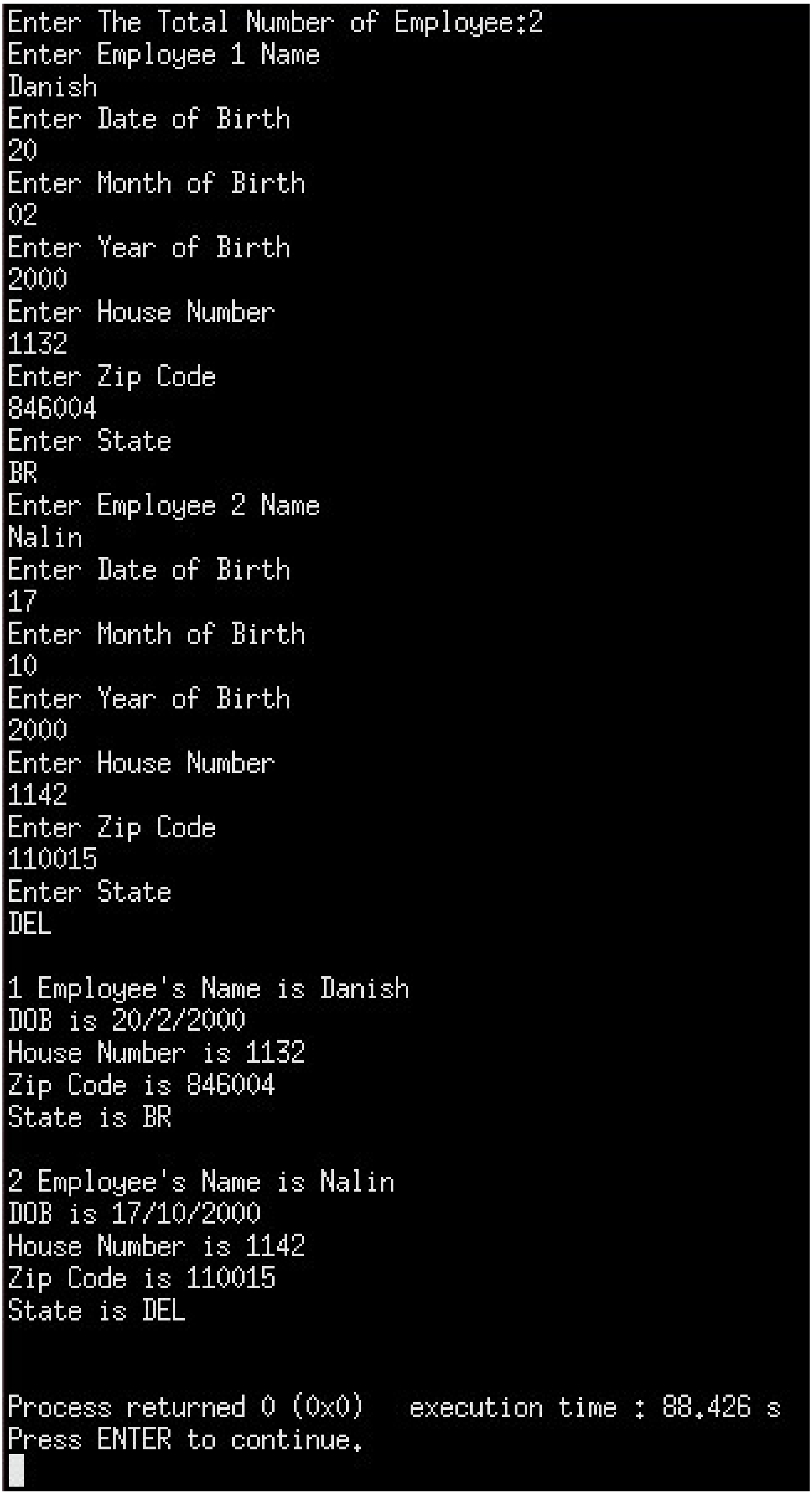
```
Enter the number of rows and columns for first array: 2 2
Enter elements:
1 2 3 4
Enter the number of rows and columns for second array: 4 4
Can't Multiply
Process returned 0 (0x0)   execution time : 16.370 s
Press ENTER to continue.
```

3: Samuel wants to store the data of his employees, which includes the following fields: (i) Name of the employee (ii) Date of birth which is a collection of {day, month, year} (iii) Address which is a collection of {house number, zip code and state}. Write a 'C' program to read and display the data of N employees using pointers to array of structures.

Program:

```
#include<stdio.h>
struct DOB
{
int day;
int month;
int year;
};
struct ADRS
{
int house_no;
long zipcode;
char state[20];
};
struct EMPLOYEE
{
char name[20];
struct DOB dob;
struct ADRS address;
};
int main()
{
struct EMPLOYEE emp[10];
struct EMPLOYEE * ptr;
int i,n;
printf("Enter The Total Number of Employee:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
ptr=emp+i;
printf("Enter Employee %d Name\n",i+1);
scanf("%s",ptr->name);
printf("Enter Date of Birth\n");
scanf("%d",&(ptr->dob.day));
printf("Enter Month of Birth\n");
scanf("%d",&(ptr->dob.month));
printf("Enter Year of Birth\n");
scanf("%d",&(ptr->dob.year));
printf("Enter House Number\n");
scanf("%d",&(ptr->address.house_no));
printf("Enter Zip Code\n");
scanf("%ld",&(ptr->address.zipcode));
printf("Enter State\n");
scanf("%s",ptr->address.state);
}
printf("\n");
for(i=0;i<n;i++)
{
ptr=emp+i;
printf("%d Employee's Name is %s\n",i+1,ptr->name);
printf("DOB is %d/%d/%d\n",ptr->dob.day,ptr->dob.month,ptr->dob.year);
printf("House Number is %d\n",ptr->address.house_no);
printf("Zip Code is %ld\n",ptr->address.zipcode);
printf("State is %s\n",ptr->address.state);
printf("\n");
}
}
```

Test Case:



```
Enter The Total Number of Employee:2
Enter Employee 1 Name
Danish
Enter Date of Birth
20
Enter Month of Birth
02
Enter Year of Birth
2000
Enter House Number
1132
Enter Zip Code
846004
Enter State
BR
Enter Employee 2 Name
Nalin
Enter Date of Birth
17
Enter Month of Birth
10
Enter Year of Birth
2000
Enter House Number
1142
Enter Zip Code
110015
Enter State
DEL

1 Employee's Name is Danish
DOB is 20/2/2000
House Number is 1132
Zip Code is 846004
State is BR

2 Employee's Name is Nalin
DOB is 17/10/2000
House Number is 1142
Zip Code is 110015
State is DEL


Process returned 0 (0x0)   execution time : 88.426 s
Press ENTER to continue.
```