

DATA STRUCTURES CS(201)

ASSIGNMENT 01

SUBMITTED TO: Dr. Sangeeta Sharma

SUBMITTED BY: Mayur Gupta

ROLL NO. : 194571

BRANCH : ECE Dual Degree

1. Consider that you are given with a database of employee records (at least 5). Each employee record having following information –

Emp_id(integer)

Emp_name(string)

Emp_city (string)

Assume that Emp_id is unique.

{Use the structure for creating database}

A) Write a function to create a database of five employees. Here, data should be taken from the user.

B) Write a function to find out the employee record from this database on the base of Emp_id.

C) Write a function to sort the employee records on the base of Emp_id. D) Write a function to count the number of employees in the database.

E) d. Write a function to add 5 more records in database.

Code :

```

#include <stdio.h>

#include<stdlib.h>

struct Empl {
    int Emp_id;
    char Emp_name[100]; char Emp_city[100];
};

struct Empl emps[5];

void inputData(){
    int i = 0;
    for (i = 0; i<5; i++){
        printf("Enter the id:\n"); scanf("%d", &emps[i].Emp_id); printf("Enter the
name :\n"); scanf("%s", &emps[i].Emp_name); printf("Enter the city :\n");
scanf("%s", &emps[i].Emp_city);
    }
}

void printData(){
    int i = 0;
    for (i = 0; i<5; i++){
        printf("id : %d\n", emps[i].Emp_id); printf("name : %s\n",
emps[i].Emp_name); printf("city : %s\n", emps[i].Emp_city);
    }
}

void find_by_id(int id){
    int i;
    for (i=0; i<5; i++){

```

```

        if (emps[i].Emp_id == id){
            printf("id : %d\n", emps[i].Emp_id); printf("name : %s\n",
emps[i].Emp_name); printf("city : %s\n", emps[i].Emp_city);
        }
    }
}

void sort_by_id(struct Empl emps[]){
    struct Empl temp;
    int i, j;
    for (i = 0; i<5-1; i++){
        for (j =0 ; j<5-i; j++){
            if(emps[i].Emp_id < emps[i+1].Emp_id){
                temp = emps[i];
                emps[i] = emps[i+1];
                emps[i+1] = temp;
            }
        }
    }
}

int main()
{
    system("cls");
    int ch;
    while(1)

```

```

{
    printf("\n\t\t*****");
printf("\n\t\t* EMPLOYEE REGISTRATION SYSTEM *");
printf("\n\t\t*****");
printf("\n");

    printf("\n\t\t\t[1]Add Employee Record "); printf("\n\t\t\t[2]Show
employee Record "); printf("\n\t\t\t[3]Find Employee Record ");
printf("\n\t\t\t[4]Sort employee Record "); printf("\n\t\t\t[5]Add more 5
employee Record "); printf("\n\t\t\t[6]Exit ");

    printf("\n\n\t\tPlease Enter Your Choice: "); fflush(stdin);

    scanf("%d",&ch);

    switch(ch)
    {
        case 1:
            inputData();

            break;

        case 2:
            printData();

            break;

        case 3:
            find_by_id(3);

            break;

        case 4:
            sort_by_id(emps);

            break;

        case 5:

```

```
inputData();

break;

case 6: printf("\n\n\t\t\t****T H A N K Y O U !!****");
printf("\n\t\t\t\t****PROGRAM TERMINATED****"); printf("\n\t\t\t\t****Made by -
Jashwant ****");

break;

system("cls");

default : printf("Wrong!");

}

}

}
```

```
*****
* EMPLOYEE REGISTRATION SYSTEM *
*****

[1]Add Employee Record
[2]Show employee Record
[3]Find Employee Record
[4]Sort employee Record
[5]Add more 5 employee Record
[6]Exit

Please Enter Your Choice:
```

2. WAP to find the number of occurrences of each element of an array in that array.

Code :

```
#include <stdio.h>

void countElement(int arr[],int size);

int main()
{
    int size;
    printf("enter the number of element - ");
    scanf("%d",&size);
    int arr[size];
    printf("enter element you want to store -");
    for(int i=0;i<size;i++)
    {
        int a;
        scanf("%d",&a);
        arr[i]=a;
    }
    countElement(arr,size);
}

void countElement(int arr[],int size)
{
    int a[size];
    for(int i=0;i<size;i++)
```

```
{
    a[i]=0;
}
for(int i=0;i<size;i++)
{
    int count=0; if(a[i]==1)
    {
        continue;
    }
    for(int j=0;j<size;j++)
    {
        if(arr[i]==arr[j])
        {
            count++;
            a[j]=1;
        }
    }
    printf("%d is present %d times\n",arr[i],count);
}
}
```

```
[(base) Apples-MacBook-Pro:C Programs apple$ gcc /Users/apple/Desktop/C\
Programs/assign_1.c -o test
[(base) Apples-MacBook-Pro:C Programs apple$ ./test
enter the number of element - 4
enter element you want to store -4
4
3
65
4 is present 2 times
3 is present 1 times
65 is present 1 times
```

3. WAP to find the nth element of the fibonacci series using recursion

Code :

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

```
int Fibo(int);
```

```
int main()
```

```
{
```

```
    int num;
```

```
    int result;
```



```
printf("Enter the nth number in Fibonacci series: "); scanf("%d",
&num);
```

```
if (num < 0)
```

```
{
```

```
printf("Fibonacci of negative numbers is not possible.\nTerms
should be in positive number\n"); }
```

```
else
```

```
{
```

```
result = Fibo(num);
```

```
printf("The %d number in Fibonacci series is %d\n", num,
result);
```

```
}
```

```
return 0;
```

```
}
```

```
int Fibo(int num)
```

```
{
```

```
if (num == 0)
```

```
{
```

```
return 0;
```

```
}
```

```
else if (num == 1)
```

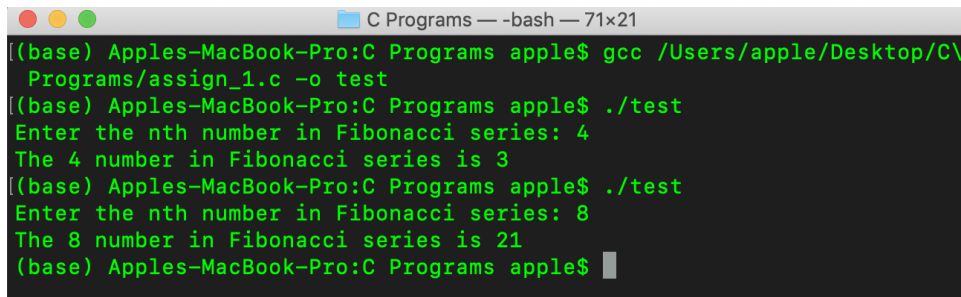
```
{
```

```
return 1;
```

```

    }
else
{
    return(Fibo(num - 1) + Fibo(num - 2));
}
}

```



A terminal window titled "C Programs — -bash — 71x21" showing the following commands and output:

```

(base) Apples-MacBook-Pro:C Programs apple$ gcc /Users/apple/Desktop/C\
Programs/assign_1.c -o test
(base) Apples-MacBook-Pro:C Programs apple$ ./test
Enter the nth number in Fibonacci series: 4
The 4 number in Fibonacci series is 3
(base) Apples-MacBook-Pro:C Programs apple$ ./test
Enter the nth number in Fibonacci series: 8
The 8 number in Fibonacci series is 21
(base) Apples-MacBook-Pro:C Programs apple$

```

4. WAP to reverse an array using pointers.

Code :

```

#include <stdio.h>

int main() {
    int values[5];
    int sum =0;
    printf("Enter elements: ");
    for (int i = 0; i < 5; ++i){

```

```

scanf("%d", values + i);
}

printf("Element entered: \n");

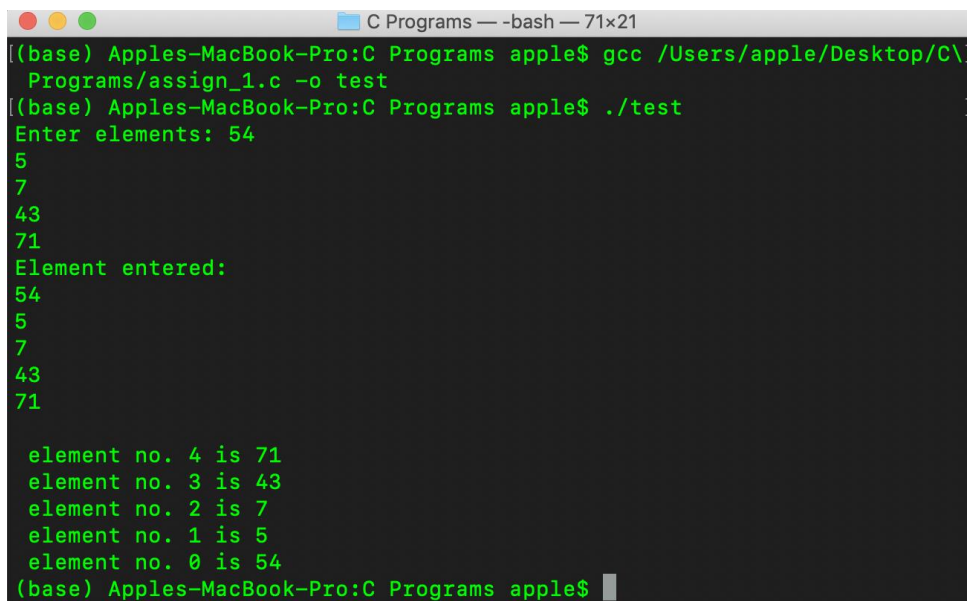
for (int i = 0; i < 5; ++i){
    printf("%d\n", *(values + i));
    /* Reversal */
}

for (int i = 4; i >= 0; i--)
{
    printf("\n element no. %d is %d ", i, *(values+i));
}

printf("\n");

return 0;
}

```



The screenshot shows a terminal window titled "C Programs — -bash — 71x21". The user enters the following commands:

```

(base) Apples-MacBook-Pro:~ apple$ gcc /Users/apple/Desktop/C\
Programs/assign_1.c -o test
(base) Apples-MacBook-Pro:~ apple$ ./test

```

The program prompts "Enter elements: 54" and the user enters the following sequence of numbers: 5, 7, 43, 71. The program then displays "Element entered:" followed by the same sequence of numbers: 54, 5, 7, 43, 71. Finally, it prints the reversed array elements with their indices:

```

element no. 4 is 71
element no. 3 is 43
element no. 2 is 7
element no. 1 is 5
element no. 0 is 54

```

The terminal prompt returns to "(base) Apples-MacBook-Pro:~ apple\$".

5. Write a program that will read a line and delete from it all occurrences of a particular word using pointers.

Code :

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main()
{
    char str[100],s[100],ch,*p,*p2,i=0,j=0;
    p2=s;
    printf("Enter a String :");
    gets(str);
    printf("Enter any Character you Want to Delete :");
    scanf("%c",&ch);
    p = str;
    while(*(p+i)!='\0')
    {
```

```

        if(*(p+i)!=ch)
        {
            *(p2+j) = *(p+i);
            j++;
        }
        i++;
    }
    printf("\n%s\n",p2);
    return 0;
}

```

```

C Programs — -bash — 71x21
[(base) Apples-MacBook-Pro:C Programs apple$ gcc /Users/apple/Desktop/C\
Programs/assign_1.c -o test
[(base) Apples-MacBook-Pro:C Programs apple$ ./test
warning: this program uses gets(), which is unsafe.
Enter a String :my name is mayur
Enter any Character you Want to Delete :m

y nae is ayur
(base) Apples-MacBook-Pro:C Programs apple$

```

6. Write a program using functions to define a structure to describe a Book with attributes such as title, author, no. of pages, price, number of sold copies, total amount of sold copies etc. Declare an array of book and do the following operations:

- a. Reading book arrays and populating values in it.
- b. Printing book array and displaying each attribute of each book in the

array. c. Find and print details of a book with maximum pages
d. Find and print details of a book with minimum cost.
e. Compute and store the total amount of money earned by selling the books.

Code:

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

```
// Structure
```

```
struct Book{  
    char title[25],author[25];  
    int pages,sold_copies;  
    float price,amount;  
}Books[5];
```

```
// a. Populating Values in array of Books
```

```
void populate(){  
    for(int i=0; i<5 ;++i){  
        printf("\n Enter Book Title : ");  
        gets(Books[i].title);  
        printf(" Enter Book Author : ");  
        gets(Books[i].author);  
        printf(" Enter number of pages : ");
```

```

scanf("%d",&Books[i].pages);
printf(" Enter Book price : Rs ");
scanf("%f",&Books[i].price);
printf(" Enter number of copies sold : ");
scanf("%d",&Books[i].sold_copies);

Books[i].amount = Books[i].price*Books[i].sold_copies;
while(getchar()!='\n'); // clears input buffer
}
}

// b. Display books

void display(){
    for(int i=0; i<5 ;++i){
        printf("\n\n Book Title : %s",Books[i].title);
        printf("\n Book Author : %s",Books[i].author);
        printf(" Enter number of pages : %d",Books[i].pages);
        printf("\n Price : Rs %.2f",Books[i].price);
        printf("\n Number of copies sold : %d",Books[i].sold_copies);
        printf("\n Amount earned of copies sold : Rs %.2f",Books[i].amount);
    }
}

```

// c. Max pages book

```
void max_pages(){
    int max=Books[0].pages,pos=0;

    for(int i=1 ; i<5;++i){
        if(max<Books[i].pages){
            max = Books[i].pages;
            pos = i;
        }
    }

    printf("\n\n Book Title : %s",Books[pos].title);
    printf("\n Book Author : %s",Books[pos].author);
    printf(" Enter number of pages : %d",Books[pos].pages);
    printf("\n Price : Rs %.2f",Books[pos].price);
    printf("\n Number of copies sold : %d",Books[pos].sold_copies);
    printf("\n Amount earned of copies sold : Rs %.2f",Books[pos].amount);
}
```

// d. Min cost

```
void min_price(){
```



```
int min=Books[0].pages,pos=0;
```

```
for(int i=1 ; i<5; ++i){
```

```
    if(min>Books[i].pages){
```

```
        min = Books[i].pages;
```

```
        pos = i;
```

```
    }
```

```
}
```

```
printf("\n\n Book Title : %s",Books[pos].title);
```

```
printf("\n Book Author : %s",Books[pos].author);
```

```
printf(" Enter number of pages : %d",Books[pos].pages);
```

```
printf("\n Price : Rs %.2f",Books[pos].price);
```

```
printf("\n Number of copies sold : %d",Books[pos].sold_copies);
```

```
printf("\n Amount earned of copies sold : Rs %.2f",Books[pos].amount);
```

```
}
```

```
// e. total money
```

```
void total_amount(){
```

```
    float sum=0;
```

```
    for(int i=0; i<5 ; ++i){
```

```
        sum += Books[i].amount;
```

```
    }
```

```
    printf("\n Rs %.2f",sum);
}

int main(){

    printf("\n-----POPULATING ARRAY OF BOOKS-----");
    populate();

    printf("\n-----DISPLAYING BOOKS-----");
    display();

    printf("\n-----BOOK WITH MAXIMUM PAGES-----");
    max_pages();

    printf("\n-----BOOKS WITH MINIMUM PRICE-----");
    min_price();

    printf("\n-----TOTAL AMOUNT OF MONEY EARNED BY SELLING
BOOKS-----");
    total_amount();

    return 0;
}
```

```
C Programs — test — 71x21
[(base) Apples-MacBook-Pro:C Programs apple$ gcc /Users/apple/Desktop/C\
Programs/assign_1.c -o test
[(base) Apples-MacBook-Pro:C Programs apple$ ./test

-----POPULATING ARRAY OF BOOKS-----
warning: this program uses gets(), which is unsafe.
Enter Book Title : 
```

7. Write a C program to read the full name of the customers and their telephone numbers. Full name includes Surname followed by first name and

then middle name. Reconstruct the customer's full names with surname first followed by comma (,) and initials of the first and middle names. Display this customer list along with their telephone no.

E.g Ghokle Suraj Ram 9876512345

should be written as

Ghokle,S.R 9876512345

Input to the program:

Enter names and telephone numbers sharma anil sudhir 1234

verma mona abhay 9876 garg sona abhi 7654 modak tinu pintu 9870 shree sahab sumit 8765 Output of the program: Customer list

garg,s.a 7654 modak,t.p 9870 sharma,a.s 1234 shree,s.s 8765
verma,m.a 9876

Code :

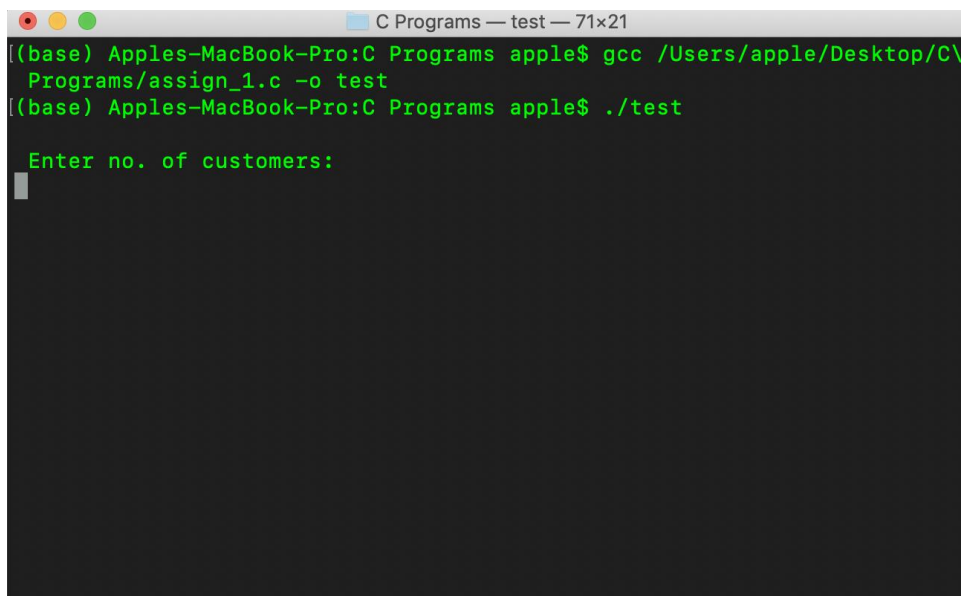
```
#include<stdio.h>
#include<string.h>
struct namedata{
    int phone;
    char a[10];
    char b[10];
    char c[10];
};
int main()
{
    struct namedata d[100];
    int i,j,temp,n;
    printf("\n Enter no. of customers:\n"); scanf("%d",&n);
    for(j=0;j<n;j++)
    {
        printf("\nEnter surname name\n"); scanf("%s",&d[j].c[i]);
        printf("Enter first name\n"); scanf("%s",&d[j].a[i]);
        printf("Enter middle name\n"); scanf("%s",&d[j].b[i]);
        printf("Enter phone number\n"); scanf("%d",&d[j].phone);
```

```

    }
    printf("\n Displaying after reconstructing details:\n");
    for(j=0;j<n;j++)
    {
        temp=strlen(d[j].c);
        printf("\n"); for(i=0;i<=temp;i++)
        {
            printf("%c",d[j].c[i]);
        }

        printf(","); printf("%c",d[j].a[0]); printf("."); printf("%c",d[j].b[0]);
        printf(" %d",d[j].phone);
    }
}

```



```

C Programs — test — 71x21
[(base) Apples-MacBook-Pro:~ apple$ gcc /Users/apple/Desktop/C\
Programs/assign_1.c -o test
[(base) Apples-MacBook-Pro:~ apple$ ./test

Enter no. of customers:

```

8. Write a program to find and replace a pattern (series of characters) in a sentence. Sentence and pattern are taken from the user.

Code :

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

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

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

```
char* replace(const char* s, const char* ow,  
              const char* nw)
```

```
{
```

```
    char* result;
```

```
    int i, cnt = 0;
```

```
    int nwlen = strlen(nw);
```

```
    int owlen = strlen(ow);
```

```
    for (i = 0; s[i] != '\0'; i++) {
```

```
        if (strstr(&s[i], ow) == &s[i]) {
```

```
            cnt++;
```

```
            i += owlen - 1;
```

```
        }
```

```
    }
```

```
    result = (char*)malloc(i + cnt * (nwlen - owlen) + 1);
```

```

i = 0;
while (*s) {
    if (strstr(s, ow) == s) {
        strcpy(&result[i], nw);
        i += nwlen;
        s += owlen;
    }
    else
        result[i++] = *s++;
}

result[i] = '\0';
return result;
}

int main()
{
    char str[] = "abcbf bcghdbc bdabchf bcchfjd";
    char c[] = "bc";
    char d[] = "..";

    char* result = NULL;

```

```

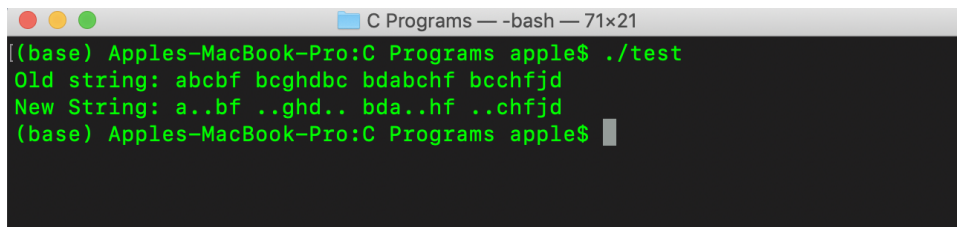
printf("Old string: %s\n", str);

result = replace(str, c, d);
printf("New String: %s\n", result);

free(result);

return 0;
}

```



```

C Programs — -bash — 71x21
(base) Apples-MacBook-Pro:C Programs apple$ ./test
Old string: abcbf bcghdbc bdabchf bcchfd
New String: a..bf ..ghd.. bda..hf ..chfd
(base) Apples-MacBook-Pro:C Programs apple$

```

9. WAP to find and remove the duplicate occurrences of an element in an array and display the modified array.

Code:

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

```
int arr[50],size;
```

```
int main(){
```

```
    printf(" Enter size of array : ");
```

```
    scanf("%d",&size);
```



```
printf(" Enter Elements of Array : - \n");
for(int i=0 ; i<size ; ++i){
    printf(" Element %d : ",i+1);
    scanf("%d",&arr[i]);
}
```

```
int a;
```

```
for(int i=0 ; i<size ; ++i){
    a=arr[i];
    for(int j=i+1 ; j<size ; ++j){
        if(a==arr[j]){
            for(int k=j ; k<size-1 ; ++k){
                arr[k] = arr[k+1];
            }
            size--;
        }
    }
}
```

```
printf("\nARRAY AFTER MODIFICATION \n");
```

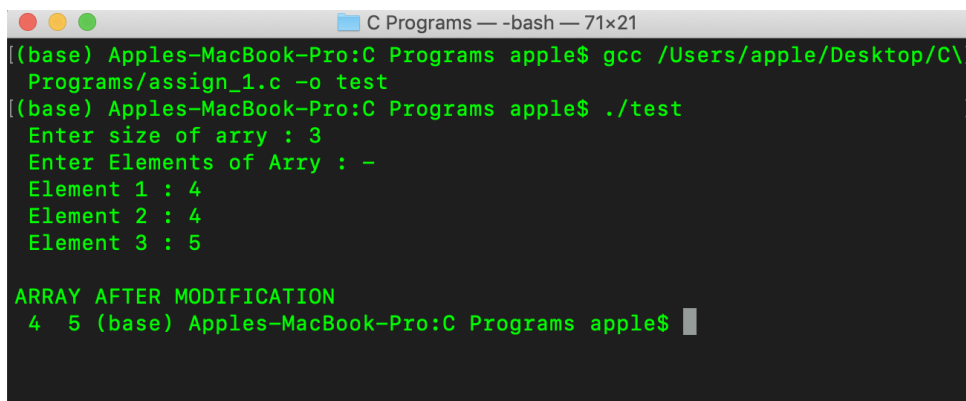
```
for(int i=0 ; i<size ; ++i){
```

```

        printf("%d ", arr[i]);
    }

    return 0;
}

```



```

C Programs — -bash — 71x21
(base) Apples-MacBook-Pro:C Programs apple$ gcc /Users/apple/Desktop/C\
Programs/assign_1.c -o test
(base) Apples-MacBook-Pro:C Programs apple$ ./test
Enter size of array : 3
Enter Elements of Array : -
Element 1 : 4
Element 2 : 4
Element 3 : 5

ARRAY AFTER MODIFICATION
4 5 (base) Apples-MacBook-Pro:C Programs apple$

```

10. Write a program to find an integer using following algorithms:

- a. Linear Search
- b. Binary Search (Take the sorted list as input)

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

```
// Linear Search
```

```
void LinearSearch(int arr[],int size,int a){
```

```
    for(int i=0 ; i<size ; ++i){
```

```

        if(a==arr[i]){
            printf("\n Element %d found at position %d .",a,i+1);
            return;
        }
    }

    printf("\n Element %d NOT FOUND!!.",a);
}

void BinarySearch(int arr[],int size,int a){
    int beg,last,mid;
    beg=0;
    last=size-1;
    while(beg<=last){
        mid = (beg+last)/2;
        if(a==arr[mid]){
            printf("\n Element %d found at position %d .",a,mid+1);
            return;
        }else if(a >arr[mid]) {
            beg = mid+1;
        }else{
            last = mid-1;
        }
    }
}

```

```

printf("\n Element %d NOT FOUND!!.",a);
}

int main(){
    int arr[] = { 1,12,7,90,6,8,84,77,69,59,71,171};
    int a;
    printf("\n----Linear Search----\n");
    printf(" Enter the Element to be found : ");
    scanf("%d",&a);
    LinearSearch(arr,13,a);

    printf("\n----Binary Search----\n");
    printf(" Enter the Element to be found : ");
    scanf("%d",&a);
    BinarySearch(arr,13,a);
}

```

```

C Programs — -bash — 71x21
[(base) Apples-MacBook-Pro:C Programs apple$ gcc /Users/apple/Desktop/C\
Programs/assign_1.c -o test
[(base) Apples-MacBook-Pro:C Programs apple$ ./test

----Linear Search----
Enter the Element to be found : 4

Element 4 NOT FOUND!!
----Binary Search----
Enter the Element to be found : 6

Element 6 NOT FOUND!!.(base) Apples-MacBook-Pro:C Programs apple$

```

11. Create an array of 10 elements using malloc functions. Further, read the array from the user and print the array.

Code:

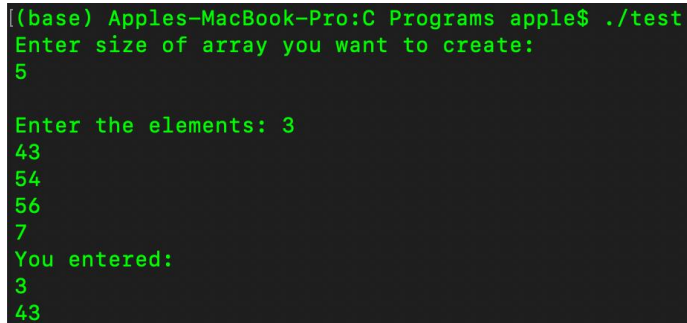
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n, i, *arr ;
    printf("Enter size of array you want to create: \n");
    scanf("%d", &n);
    arr = (int*) malloc(n * sizeof(int));
    if(arr == NULL)
    {
        printf("Memory not allocated.");
        exit(0);
    }
    printf("\nEnter the elements: ");
    for(i = 0; i < n; ++i)
    {
        scanf("%d", arr + i);
    }
    printf("You entered: \n");
```

```

for (int i = 0; i < n; ++i){
    printf("%d\n", *(arr + i));
    free(arr);
}

return 0;
}

```



```

((base) Apples-MacBook-Pro:C Programs apple$ ./test
Enter size of array you want to create:
5

Enter the elements: 3
43
54
56
7
You entered:
3
43

```

12. Write a program for matrix multiplication and matrix addition.

Code:

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

```
int main()
```

```
{
```

```
    int m, n, p, q, c, d, k, sum = 0;
```

```
    int first[10][10], second[10][10], multiply[10][10], add[10][10];
```

```
    printf("Enter number of rows and columns of first matrix\n");
```

```
    scanf("%d%d", &m, &n);
```

```
    printf("Enter elements of first matrix row wise\n");
```

```
for (c = 0; c < m; c++)
    for (d = 0; d < n; d++)
        scanf("%d", &first[c][d]);

printf("Enter number of rows and columns of second matrix\n");
scanf("%d%d", &p, &q);

printf("Enter elements of second matrix row wise\n");

for (c = 0; c < p; c++)
    for (d = 0; d < q; d++)
        scanf("%d", &second[c][d]);

if (n != p)
    printf("The multiplication isn't possible.\n");
else
{
    for (c = 0; c < m; c++) {
        for (d = 0; d < q; d++) {
            for (k = 0; k < p; k++) {
                sum = sum + first[c][k]*second[k][d];
            }

            multiply[c][d] = sum;
            sum = 0;
        }
    }
}
```

```

    }

    printf("Product of the matrices:\n");

    for (c = 0; c < m; c++) {
        for (d = 0; d < q; d++)
            printf("%d\t", multiply[c][d]);

        printf("\n");
    }
}

if(p==m && q==n){
    for (c = 0; c < m; c++) {
        for (d = 0; d < q; d++)
            add[c][d] = first[c][d] + second[c][d];
    }

    printf("Addition of the matrices:\n");
    for (c = 0; c < m; c++) {
        for (d = 0; d < q; d++)
            printf("%d\t", add[c][d]);

        printf("\n");
    }
}
else{

```

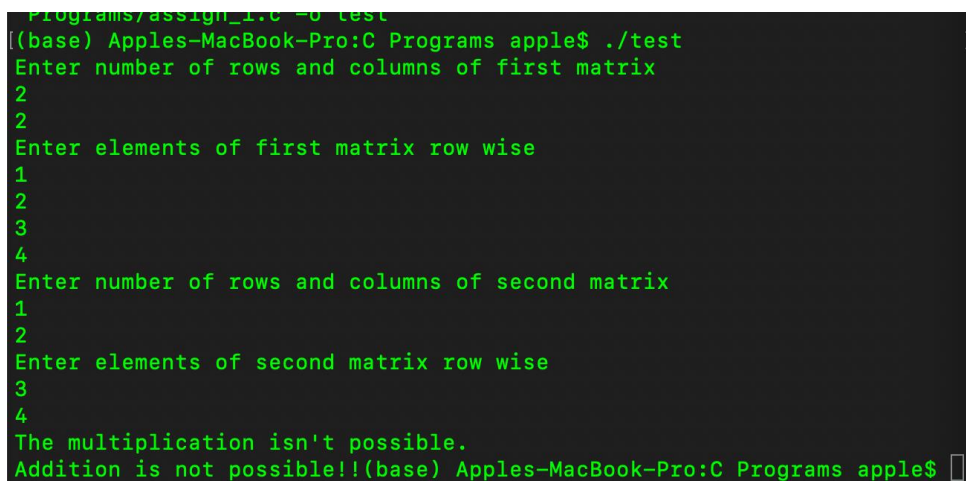


```

    printf("Addition is not possible!!");
}

return 0;
}

```



A terminal window with a dark background and green text. The prompt is `Programs/assign_1.c -0 test`. The user runs `./test`. The program prompts for the dimensions and elements of two matrices. For the first matrix, the user enters 2 rows and 2 columns, followed by elements 1, 2, 3, and 4. For the second matrix, the user enters 1 row and 2 columns, followed by elements 3 and 4. The program outputs "The multiplication isn't possible." and "Addition is not possible!!".

```

Programs/assign_1.c -0 test
(base) Apples-MacBook-Pro:~ apple$ ./test
Enter number of rows and columns of first matrix
2
2
Enter elements of first matrix row wise
1
2
3
4
Enter number of rows and columns of second matrix
1
2
Enter elements of second matrix row wise
3
4
The multiplication isn't possible.
Addition is not possible!!(base) Apples-MacBook-Pro:~ apple$

```

13. Write a program to perform various operations on 3D array i.e. Insertion, Deletion and Searching of elements.

Code :

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

```

int main(){
    int arr[10][10][10],l,b,h;

```

```
printf("\n-----INSERTION-----\n");
```

```
printf(" Enter the size of 3D array (l,b,h) (10 max.) : \n");
```

```
scanf("%d%d%d",&l,&b,&h);
```

```
if(l>10 || b>10 || h>10 || l<0 || b<0 || h<0){
```

```
    printf("\n INVALID DIMENSIONS");
```

```
    return 0;
```

```
}else{
```

```
    printf("\n Enter Elements of Array : -\n");
```

```
    for(int i=0 ; i<l ; ++i){
```

```
        for(int j=0 ; j<b ; ++j){
```

```
            for(int k=0 ; k<h ; ++k){
```

```
                printf(" Element (%d,%d,%d) : ",i+1,j+1,k+1);
```

```
                scanf("%d",&arr[i][j][k]);
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
printf("\n-----DELETION-----\n");
```

```
int l1,b1,h1;
```

```
printf("\n Enter the index of 3D array (l,b,h) to be deleted : ");
```

```
scanf("%d%d%d",&l1,&b1,&h1);
```

```

if(l1>l || b1>b || h1>h || l1<0 || b1<0 || h1<0){
    printf("\n INVALID DIMENSIONS");
}else{
    printf("\n Element %d of Array is deleted.",arr[l1-1][b1-1][h1-1]);
    arr[l1-1][b1-1][h1-1] = 0;
}

```

```

printf("\n-----SEARCHING-----\n");

```

```

int x;
printf("\n Enter the element of 3D array searched : ");
scanf("%d",&x);

```

```

for(int i=0 ; i<l ; ++i){
    for(int j=0 ; j<b ; ++j){
        for(int k=0 ; k<h ; ++k){
            if( x == arr[i][j][k]){
                printf("\n Element %d Found at (%d,%d,%d)",x,i+1,j+1,k+1);
                return 0;
            }
        }
    }
}

```

```

return 0;
}

```

```
C Programs — test — 71x21
[(base) Apples-MacBook-Pro:C Programs apple$ ./test

-----INSERTION-----
Enter the size of 3D array (l,b,h) (10 max.) :
2
2
2

Enter Elements of Array : -
Element (1,1,1) : 2
Element (1,1,2) : 1
Element (1,2,1) : 34
Element (1,2,2) : 32
Element (2,1,1) : 1
Element (2,1,2) : 45
Element (2,2,1) : 53
Element (2,2,2) : 32

-----DELETION-----
Enter the index of 3D array (l,b,h) to be deleted : 
```

14. WAP to check whether the matrix is a Sparse matrix or not.

Code :

```
#include<stdio.h>

int main()
{
    int m,n,a[10][10],i,j,count = 0;

    printf("Enter the order of the matrix i.e, rows and columns:\n");
    scanf("%d%d",&m,&n);

    printf("\nEnter the elements of the matrix:\n"); for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
}
```

```

printf("\nMatrix entered by you is:\n");
for(i=0;i<m;i++)
{
printf("\n");
for(j=0;j<n;j++)
{
printf("%d\t",a[i][j]);
}
}
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
if(a[i][j] == 0) {
count++;
}
}
}
if(count > ((m*n)/2))
printf("Given matrix is a sparse matrix");
else
printf("Given matrix is not a sparse matrix");
return 0;
}

```

```
C Programs — -bash — 71x21
((base) Apples-MacBook-Pro:C Programs apple$ ./test
Enter the order of the matrix i.e, rows and columns:
2
4
Enter the elements of the matrix:
1
2
0
3
0
0
0
5
Matrix entered by you is:
1      2      0      3
0      0      0      5      Given matrix is not a sparse matrix(bas
e) Apples-MacBook-Pro:C Programs apple$
```