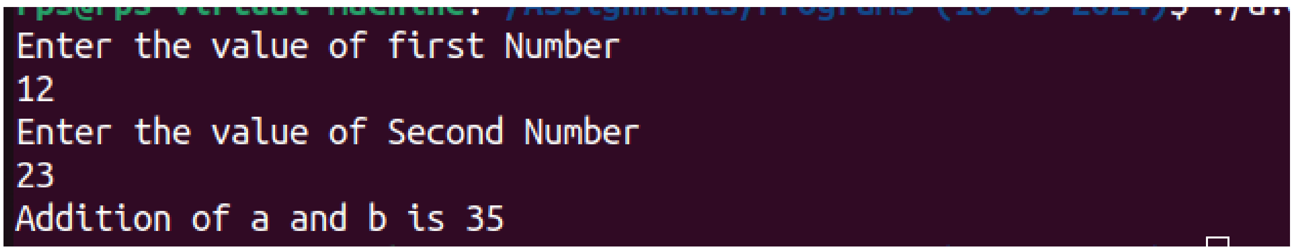
Tasks [C-Programming]

DAY-1

1. Find sum of Two Numbers.

```
#include<stdio.h>
int main()
{
        int a=0;
        int b=0;
        printf("Enter the value of first Number\n");
        scanf("%d",&a);
        printf("Enter the value of Second Number\n");
        scanf("%d",&b);

        printf("Addition of a and b is %d\n",a+b);
        return 0;
}
```

Output:



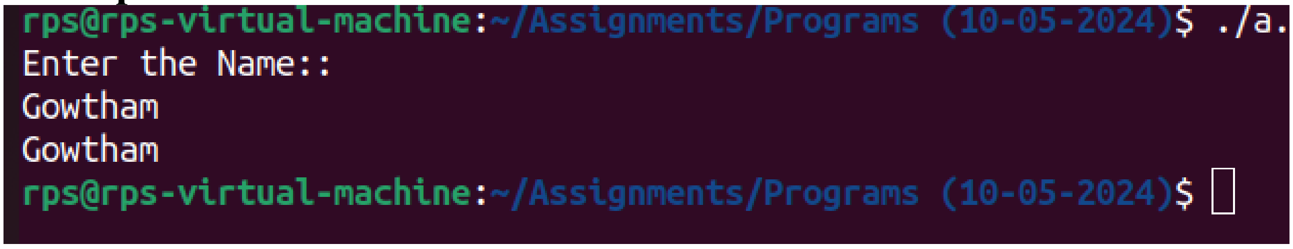
```
Enter the value of first Number
12
Enter the value of Second Number
23
Addition of a and b is 35
```

2.Print the String by using scanf by giving user inputs.

Code:

```
#include<stdio.h>
int main()
{
        printf("Enter the Name::\n");
        char s[20];
        scanf("%s",&s);
        printf("%s",s);
        return 0;
}
```

Output:



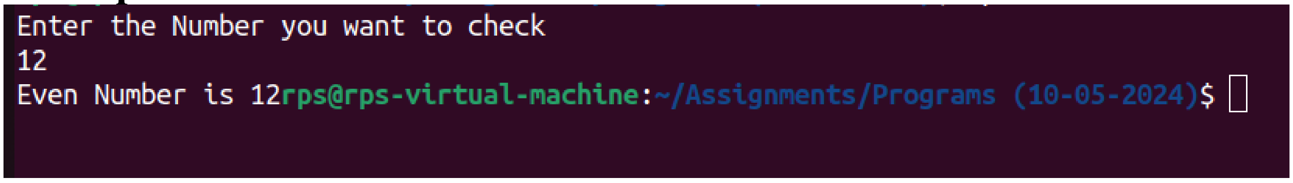
```
rps@rps-virtual-machine:~/Assignments/Programs (10-05-2024)$ ./a.
Enter the Name::
Gowtham
Gowtham
rps@rps-virtual-machine:~/Assignments/Programs (10-05-2024)$
```

3.Check the Given Number is Even or odd.

Code:

```
#include<stdio.h>
int main()
{
        printf("Enter the Number you want to check\n");
        int a;
        scanf("%d",&a);
        if(a%2==0){
                printf("Even Number is %d",a);
        }
        else
                printf("Odd number is %d",a);
        return 0;
}
```

Output:



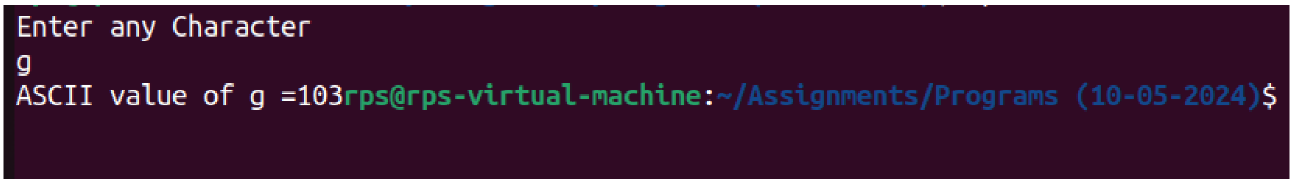
```
Enter the Number you want to check
12
Even Number is 12rps@rps-virtual-machine:~/Assignments/Programs (10-05-2024)$
```

4.Find the ASCII value of an Character.

Code:

```
#include<stdio.h>
int main()
{
        char c;
        printf("Enter any Character\n");
        scanf("%c",&c);
        printf("ASCII value of %c =%d",c,c);
        return 0;
}
```

Output:

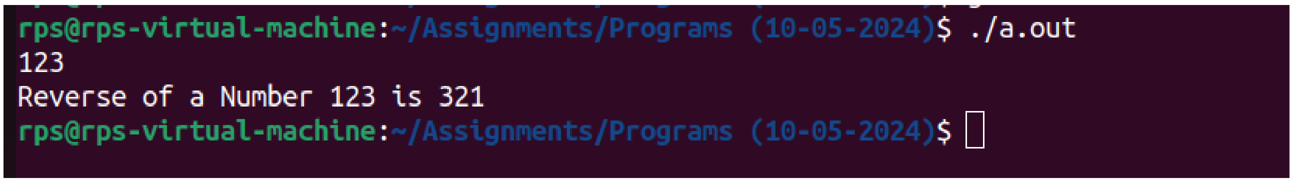


```
Enter any Character
g
ASCII value of g =103rps@rps-virtual-machine:~/Assignments/Programs (10-05-2024)$
```

5. Find the Reverse of an Number?
Code:

```
        #include<stdio.h>
        int main()
        {
                int n;
                scanf("%d",&n);
                int rev=0,t=n;
                while(n>0){
                        int r=n%10;
                        n=n/10;
                        rev=rev*10+r;
                }
                n=t;
                printf("Reverse of a Number %d is %d\n",n,rev);
                return 0;
        }
```

Output:

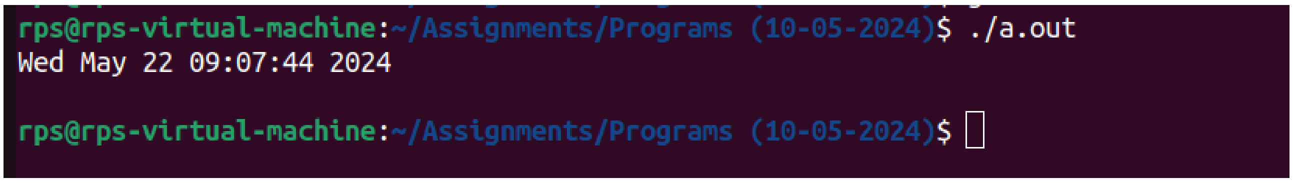


```
rps@rps-virtual-machine:~/Assignments/Programs (10-05-2024)$ ./a.out
123
Reverse of a Number 123 is 321
rps@rps-virtual-machine:~/Assignments/Programs (10-05-2024)$
```

6. Find the time and Date by using <time.h> header?
Code:

```
#include<stdio.h>
#include<time.h>
int main()
{
        time_t t=time(NULL);
        struct tm *ct=localtime(&t);
        printf("%s\n",asctime(ct));
        return 0;
}
```

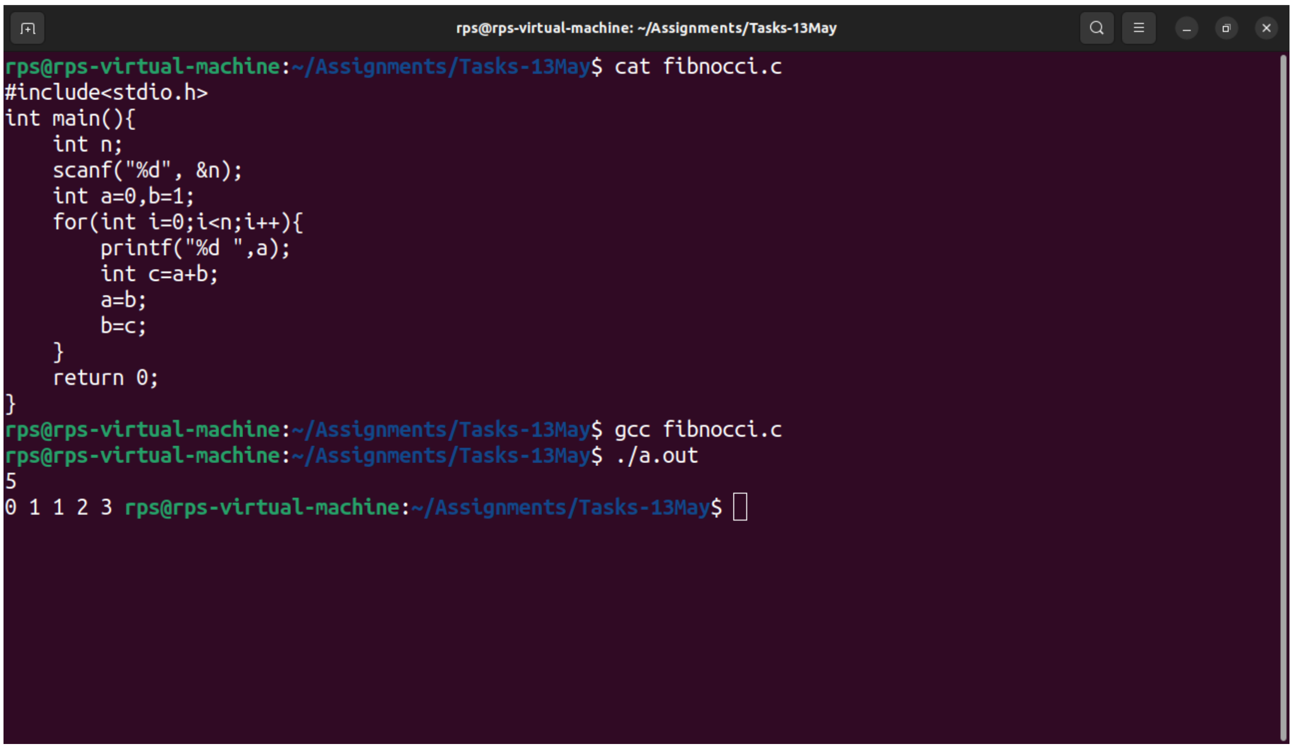
Output:



```
rps@rps-virtual-machine:~/Assignments/Programs (10-05-2024)$ ./a.out
Wed May 22 09:07:44 2024

rps@rps-virtual-machine:~/Assignments/Programs (10-05-2024)$ 
```

DAY-2

1. Find N Fibnocci Numbers by using for loop.



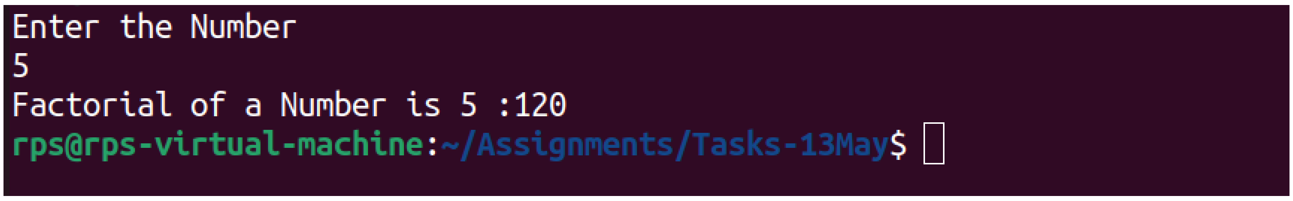
```
rps@rps-virtual-machine:~/Assignments/Tasks-13May$ cat fibnocci.c
#include<stdio.h>
int main(){
    int n;
    scanf("%d", &n);
    int a=0,b=1;
    for(int i=0;i<n;i++){
        printf("%d ",a);
        int c=a+b;
        a=b;
        b=c;
    }
    return 0;
}
rps@rps-virtual-machine:~/Assignments/Tasks-13May$ gcc fibnocci.c
rps@rps-virtual-machine:~/Assignments/Tasks-13May$ ./a.out
5
0 1 1 2 3 rps@rps-virtual-machine:~/Assignments/Tasks-13May$ 
```

2. Find the Factorial of a Number?

Code:

```
#include<stdio.h>
int main(){
        printf("Enter the Number \n");
        int a;
        scanf("%d",&a);
        int fact=1;
        for(int i=1;i<=a;i++){
        fact=fact*i;
        }
        printf("Factorial of a Number is %d :%d",a,fact);
```

```
		return 0;
	}
```
Output:



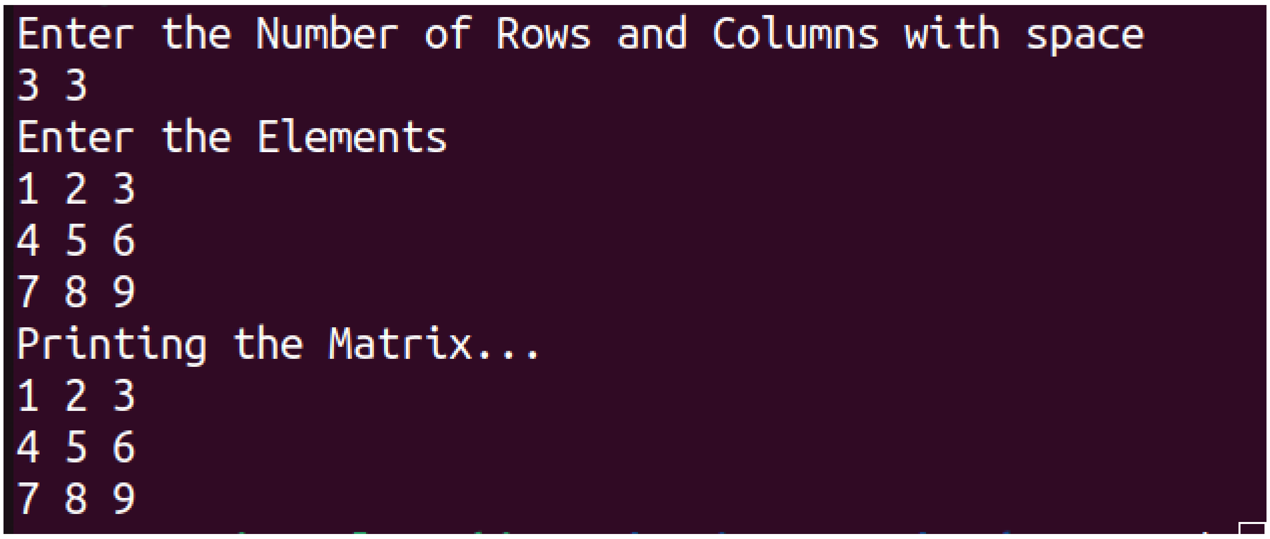
```
Enter the Number
5
Factorial of a Number is 5 :120
rps@rps-virtual-machine:~/Assignments/Tasks-13May$
```

3. **Take the inputs from the user and print the matrix.**

Code:

```
	#include<stdio.h>
	int main(){
		int r,c;
		printf("Enter the Number of Rows and Columns with   space\n");
		scanf("%d %d",&r,&c);
		printf("Enter the Elements\n");
		int a[r][c];
		for(int i=0;i<r;i++){
		for(int j=0;j<c;j++){
				scanf("%d",&a[i][j]);
		}
		}

		printf("Printing the Matrix...");
		for(int i=0;i<r;i++){
		for(int j=0;j<c;j++){
				printf("%d ",a[i][j]);
		}
		printf("\n");
		}
}
```

Output:

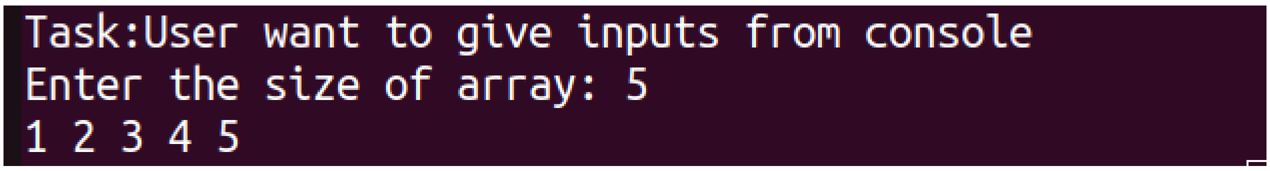

```
Enter the Number of Rows and Columns with space
3 3
Enter the Elements
1 2 3
4 5 6
7 8 9
Printing the Matrix...
1 2 3
4 5 6
7 8 9
```

4. **Take the inputs from the user and stored into an arrays.**

Code:

```
#include<stdio.h>
int main(){
        printf("Task:User want to give inputs from console\n");
        int n;
        printf("Enter the size of array: ");
        scanf("%d", &n);
        int a[n];
        for(int i=0; i<n; i++){
                scanf("%d", &a[i]);
        }
}
```

Output:


```
Task:User want to give inputs from console
Enter the size of array: 5
1 2 3 4 5
```

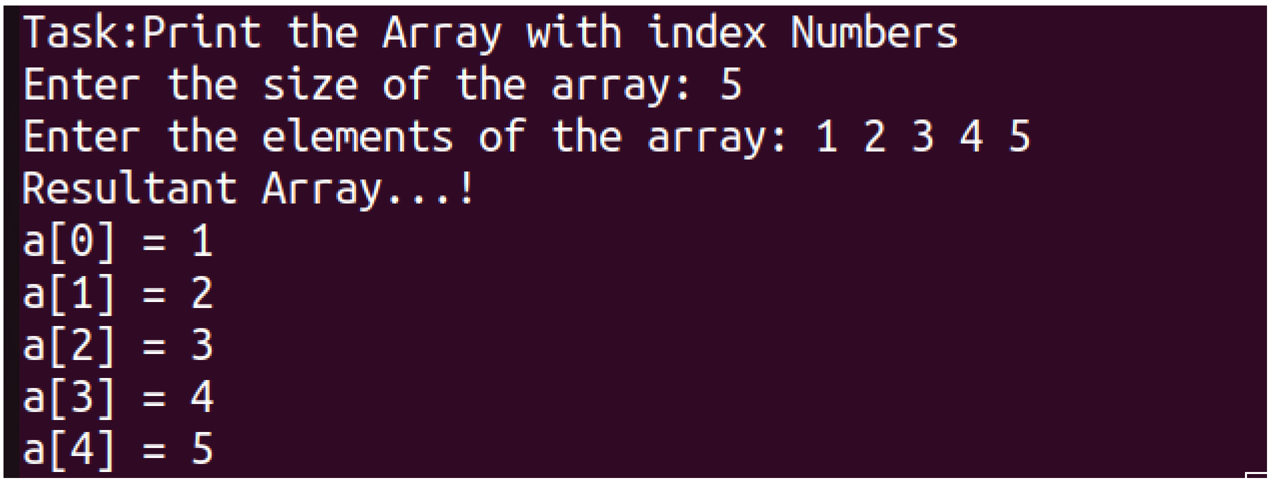
5. **Print all the elements in an array with index Numbers?**

Code:

```
#include<stdio.h>
int main(){
    printf("Task:Print the Array with index Numbers\n");
    int n;
    printf("Enter the size of the array: ");
    scanf("%d",&n);
    int a[n];
```

```
        printf("Enter the elements of the array: ");
        for(int i=0;i<n;i++){
            scanf("%d",&a[i]);
        }
        printf("Resultant Array...!\n");
        for(int i=0;i<n;i++){
            printf("a[%d] = %d\n",i,a[i]);
        }
        return 0;
    }
```

Output:

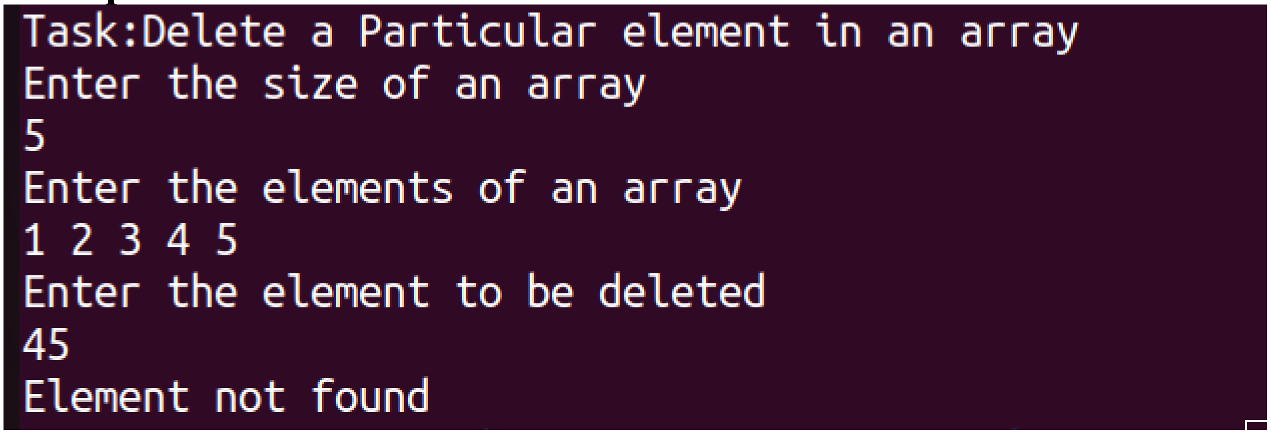


```
Task:Print the Array with index Numbers
Enter the size of the array: 5
Enter the elements of the array: 1 2 3 4 5
Resultant Array...!
a[0] = 1
a[1] = 2
a[2] = 3
a[3] = 4
a[4] = 5
```

6. **Delete a Particular Element in an array.**

Code:

```
    #include<stdio.h>
    int main(){
        printf("Task:Delete a Particular element in an array\n");
        int n;
        printf("Enter the size of an array\n");
        scanf("%d",&n);
        int a[n];
        printf("Enter the elements of an array\n");
        for(int i=0;i<n;i++){
            scanf("%d",&a[i]);
        }
        int k;
        printf("Enter the element to be deleted\n");
        scanf("%d",&k);
        for(int i=0;i<n;i++){
            if(a[i]==k){
                a[i]=0;
                printf(" Deleted Element is %d at index of %d\n",k,i);
                printf("Updated value from %d index is  %d\n",i,a[i]);
                break;
            }
            else if(i==n-1){
                printf("Element not found\n");
            }
        }
    }
```
Output:

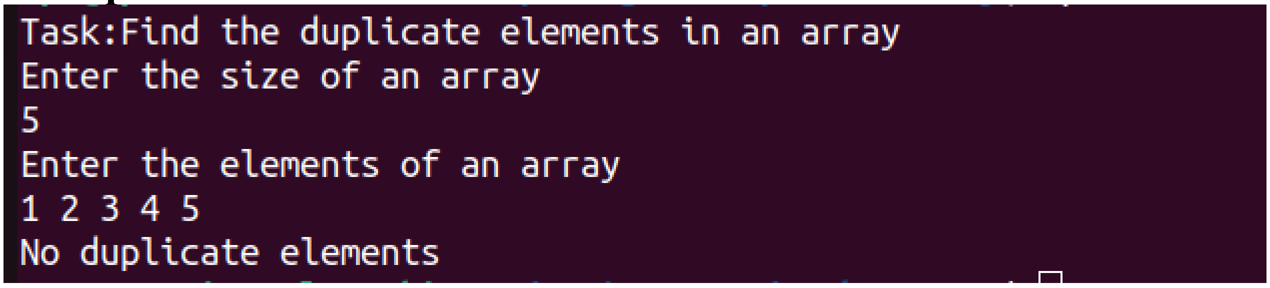


```
Task:Delete a Particular element in an array
Enter the size of an array
5
Enter the elements of an array
1 2 3 4 5
Enter the element to be deleted
45
Element not found
```

7. **Find Duplicate elements in the array?**

Code:
```
#include<stdio.h>
int main(){
    // int a[20]={1,2,3,4,5,6,7,8,9,10,2,12,13,14,15,16,18,33,3,1};
    printf("Task:Find the duplicate elements in an array\n");
    int n;
    printf("Enter the size of an array\n");
    scanf("%d",&n);
    int a[n];
    printf("Enter the elements of an array\n");
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int d=0;
    for(int i=0;i<n;i++){
        int c=0;
        for(int j=i;j<n;j++){
            if(a[i]==a[j]){
                c++;
            }
        }
        if(c>1){
            d++;
            printf("%d\n",a[i]);
        }
    }
    if(d==0){
        printf("No duplicate elements\n");
    }
}
```

Output:



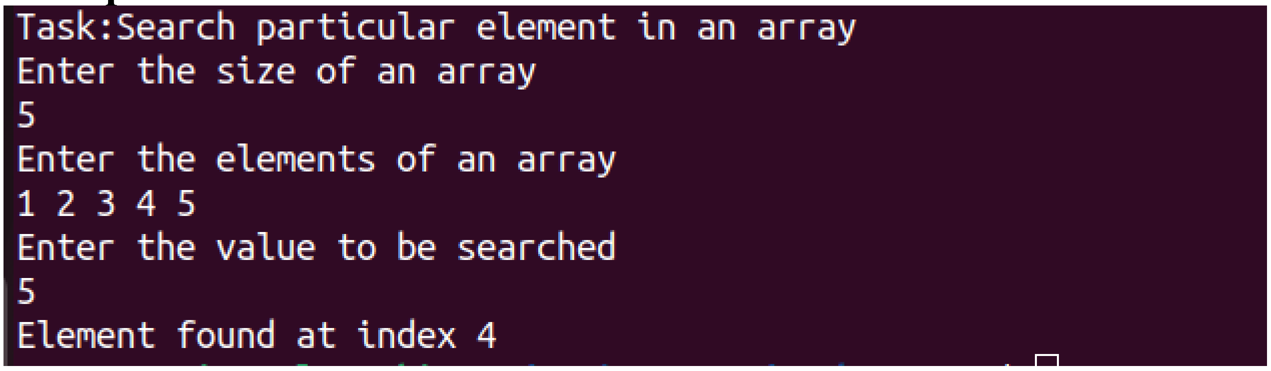
```
Task:Find the duplicate elements in an array
Enter the size of an array
5
Enter the elements of an array
1 2 3 4 5
No duplicate elements
```

8. Search a Element in an array

Code:

```
#include<stdio.h>
int main(){
    printf("Task:Search particular element in an array\n");
    int n;
    printf("Enter the size of an array\n");
    scanf("%d",&n);
    int a[n];
    printf("Enter the elements of an array\n");
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int k;
    printf("Enter the value to be searched\n");
    scanf("%d",&k);
    int i;
    for(i=0;i<n;i++){
        if(a[i]==k){
            printf("Element found at index %d\n",i);
            break;
        }
        else if(i==n-1){
                printf("Element not found\n");
            }
        }
    }
```

Output:



```
Task:Search particular element in an array
Enter the size of an array
5
Enter the elements of an array
1 2 3 4 5
Enter the value to be searched
5
Element found at index 4
```

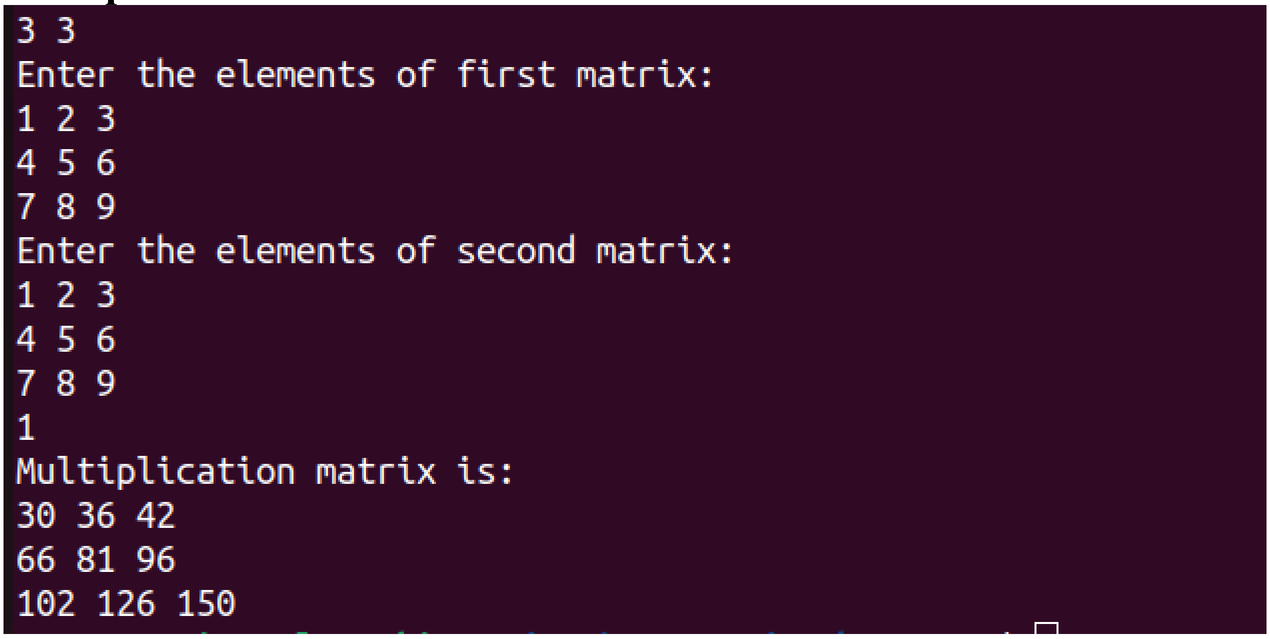
9. Multipilication of Matrix.

Code:

```
#include<stdio.h>
int main(){
    int r,c;
    scanf("%d %d",&r,&c);
    int a[r][c];
    printf("Enter the elements of first matrix:\n");
    for(int i=0;i<r;i++){
        for(int j=0;j<c;j++){
            scanf("%d",&a[i][j]);
        }
    }
    printf("Enter the elements of second matrix:\n");
    int b[r][c];
    for(int i=0;i<r;i++){
        for(int j=0;j<c;j++){
            scanf("%d ",&b[i][j]);
        }
//      printf("\n");
    }
    int res[r][c];
    printf("Multiplication matrix is:\n");
    for(int i=0;i<r;i++){
        for(int j=0;j<c;j++){
            res[i][j]=0;
            for(int k=0;k<c;k++){
                res[i][j]+=a[i][k]*b[k][j];
```

```
                        }
                        printf("%d ",res[i][j]);
                    }

                    printf("\n");
                }
            }
```

Output:



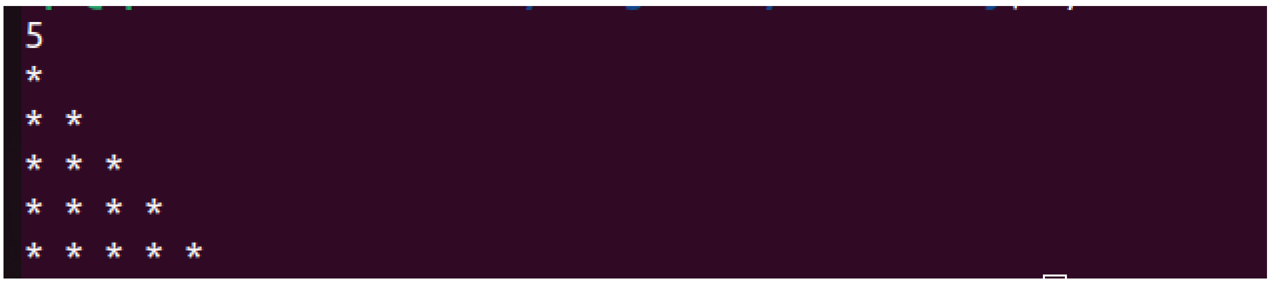
```
3 3
Enter the elements of first matrix:
1 2 3
4 5 6
7 8 9
Enter the elements of second matrix:
1 2 3
4 5 6
7 8 9
1
Multiplication matrix is:
30 36 42
66 81 96
102 126 150
```

10.Print a Star Program.

Code:

```
#include<stdio.h>
int main()
{
        int n;
        scanf("%d",&n);
        for(int i=1;i<=n;i++)
        {
                for(int j=1;j<=i;j++){
                        printf("* ");
                }
                printf("\n");
        }
}
```

Output:



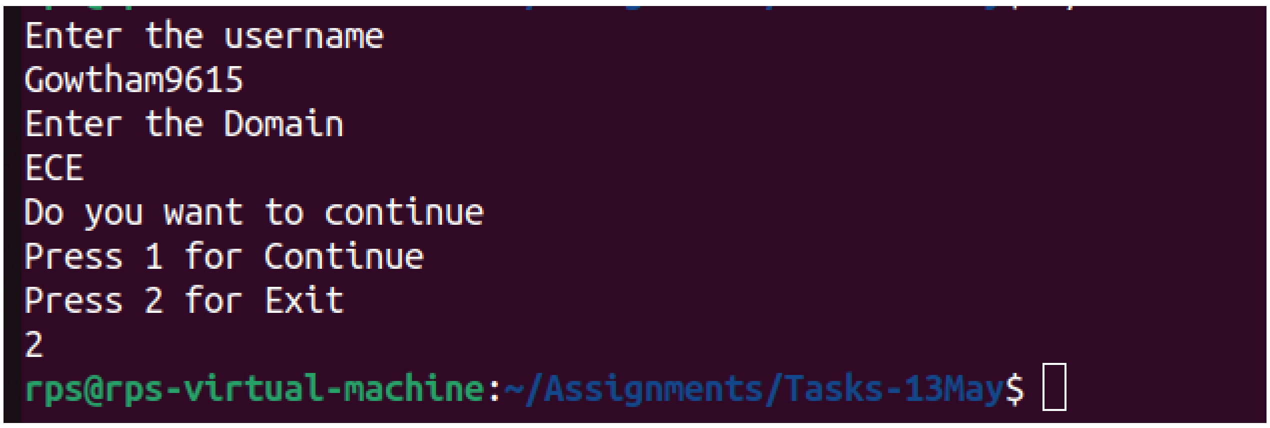
```
5
*
* *
* * *
* * * *
* * * * *
```

11. Enter the username and domain in While loop the code Behaviour depends on the user interactions.

Code:

```
#include<stdio.h>
#include<stdbool.h>
int main(){
        while(true)
        {
                start:
                printf("Enter the username\n");
                char ch[20];
                scanf("%s",ch);
                printf("Enter the Domain\n");
                char ch1[20];
                scanf("%s",ch1);
                printf("Do you want to continue\n");
                printf("Press 1 for Continue\n");
                printf("Press 2 for Exit\n");
                int a;
                scanf("%d",&a);
                if(a==2)
                        break;
                else if(a==1)
                        goto start;
                return 0;
        }

}
```

Output:



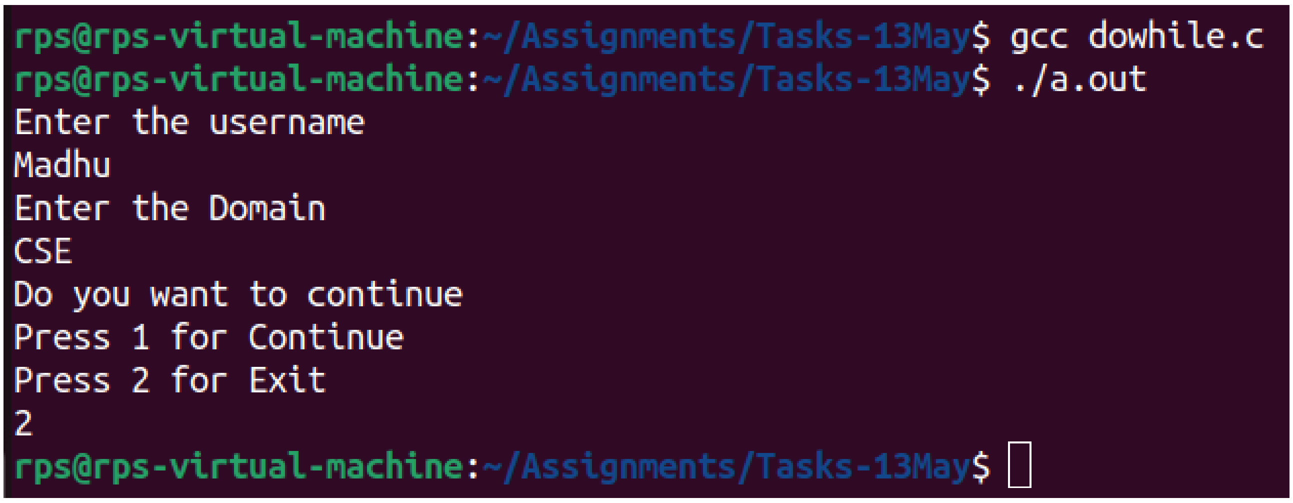
```
Enter the username
Gowtham9615
Enter the Domain
ECE
Do you want to continue
Press 1 for Continue
Press 2 for Exit
2
rps@rps-virtual-machine:~/Assignments/Tasks-13May$
```

12. Satisfy the do while condition by asking username and domain and exit of the program depends on the user.

Code:

```
#include<stdio.h>
int main(){
        do{
                start:
                printf("Enter the username\n");
            char ch[20];
            scanf("%s",ch);
            printf("Enter the Domain\n");
            char ch1[20];
            scanf("%s",ch1);
            printf("Do you want to continue\n");
            printf("Press 1 for Continue\n");
            printf("Press 2 for Exit\n");
            int a;
            scanf("%d",&a);
            if(a==2)
                  break;
            else if(a==1)
                  goto start;
        }
        while(1);
}
```

Output:


```
rps@rps-virtual-machine:~/Assignments/Tasks-13May$ gcc dowhile.c
rps@rps-virtual-machine:~/Assignments/Tasks-13May$ ./a.out
Enter the username
Madhu
Enter the Domain
CSE
Do you want to continue
Press 1 for Continue
Press 2 for Exit
2
rps@rps-virtual-machine:~/Assignments/Tasks-13May$
```

DAY-3

1. **Write a program to create a data ,delete a data ,view a data ,modify the data in a files by using file handling**

Code:

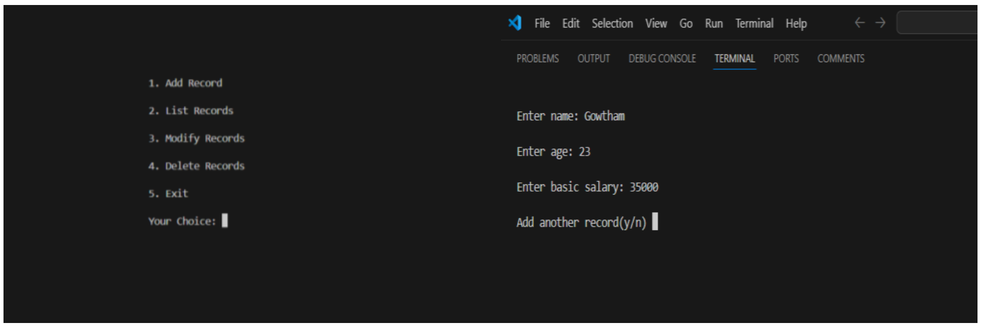
```
#include <stdio.h>

#include <conio.h>

#include <windows.h>

#include <string.h>

COORD coord = {0,0};

void gotoxy(int x,int y)

{

    coord.X = x;

    coord.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),coord
);

}


int main()

{

    FILE *fp, *ft;
```

```
char another, choice;
struct emp
{
    char name[40];
    int age;
    float bs;
};
struct emp e;
char empname[40];
long int recsize;
fp = fopen("D:\\Assignments\\Demo.txt","rb+");
if(fp == NULL)
{
    fp = fopen("D:\\Assignments\\Demo.txt","wb+");
    if(fp == NULL)
    {
        printf("Connot open file");
        exit(1);
    }
}
recsize = sizeof(e);
while(1)
{
    system("cls");
    gotoxy(30,10);
    printf("1. Add Record");
    gotoxy(30,12);
```

```
printf("2. List Records");
gotoxy(30,14);
printf("3. Modify Records");
gotoxy(30,16);
printf("4. Delete Records");
gotoxy(30,18);
printf("5. Exit");
gotoxy(30,20);
printf("Your Choice: ");
fflush(stdin);
choice  = getche();
switch(choice)
{
case '1':
    system("cls");
    fseek(fp,0,SEEK_END);

    another = 'y';
    while(another == 'y')
    {
        printf("\nEnter name: ");
        scanf("%s",e.name);
        printf("\nEnter age: ");
        scanf("%d", &e.age);
        printf("\nEnter basic salary: ");
        scanf("%f", &e.bs);
        fwrite(&e,recsize,1,fp);
        printf("\nAdd another record(y/n) ");
        fflush(stdin);
        another = getche();
    }
    break;
case '2':
    system("cls");
    rewind(fp);
    while(fread(&e,recsize,1,fp)==1)
    {
        printf("\n%s %d %.2f",e.name,e.age,e.bs);
    }
    getch();
    break;
case '3':
    system("cls");
    another = 'y';
    while(another == 'y')
    {
        printf("Enter the employee name to modify: ");
        scanf("%s", empname);
        rewind(fp);
        while(fread(&e,recsize,1,fp)==1)
        {
            if(strcmp(e.name,empname) == 0)
            {
                printf("\nEnter new name,age and bs: ");
                scanf("%s%d%f",e.name,&e.age,&e.bs);
                fseek(fp,-recsize,SEEK_CUR);
                fwrite(&e,recsize,1,fp);
                break;
            }
        }
        printf("\nModify another record(y/n)");
        fflush(stdin);
        another = getche();
    }
    break;
case '4':
    system("cls");
    another = 'y';
    while(another == 'y')
    {
        printf("\nEnter name of employee to delete: ");
        scanf("%s",empname);
        ft = fopen("D:\\Assignments\\Temp.txt","wb");
        rewind(fp);
        while(fread(&e,recsize,1,fp) == 1)
        {
            if(strcmp(e.name,empname) != 0)
            {
                fwrite(&e,recsize,1,ft);
            }
        }
```

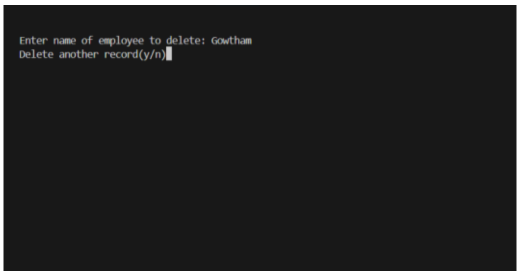
```
            fclose(fp);
            fclose(ft);
            remove("D:\\Assignments\\Demo.txt");
        rename("D:\\Assignments\\Temp.txt","D:\\Assignments\\Demo.txt");
            fp = fopen("D:\\Assignments\\Demo.txt", "rb+");
            printf("Delete another record(y/n)");
            fflush(stdin);
            another = getche();
        }
        break;
    case '5':
        fclose(fp);
        exit(0);
    }
  }
  return 0;
}
```

Output:


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
1. Add Record
2. List Records
3. Modify Records
4. Delete Records
5. Exit
Your Choice:
```

```
Enter name: Gowtham
Enter age: 23
Enter basic salary: 35000
Add another record(y/n)
```



```
Enter name of employee to delete: Gowtham
Delete another record(y/n)
```

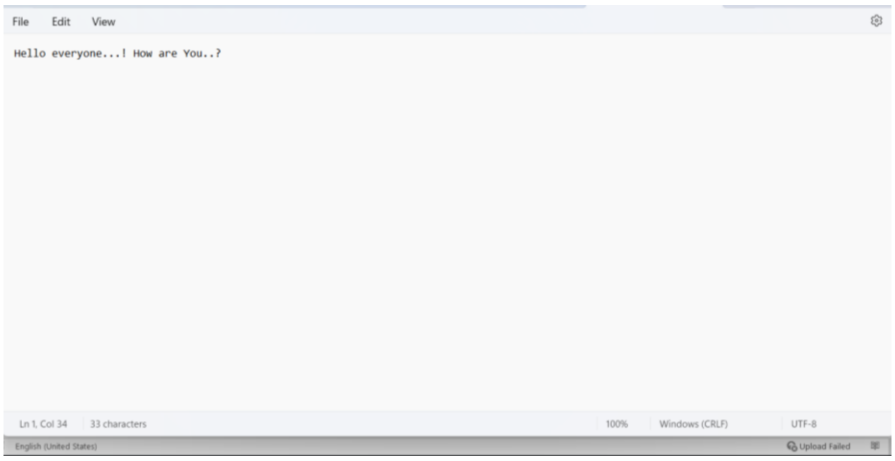
2.Read the data form file.

Code:

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

void main() {
    FILE *fptr;
    fptr = fopen("D:\\Assignments\\t1.txt", "r");
    if (fptr == NULL) {
        printf("Error!!!!!\n");
        exit(1);
    }

    char a[100];
    while (fgets(a, sizeof(a), fptr) != NULL) {
        printf("%s", a);
    }
    fclose(fptr);
}
```

Output:

Note:Whatever the data inside your txt file it will print like that Only. This is my t1.txt as you can observe Below.



```
Hello everyone...! How are You..?
```

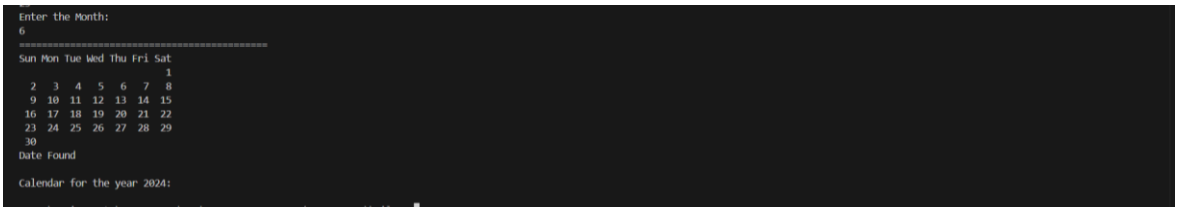
3.Write a Program to find the month of a calender,The month has to enter by the user.

Code:

```
#include <stdio.h>
#include <time.h>
#include<ctype.h>
int choice(){
    printf("Enter the Month:\n");
    int a;
    scanf("%d", &a);
    printf("==============================================\n");

    if(a==1)
    {
        return 0;
    }
    else if(a==2){
        return 1;
    }
    else if(a==3){
```

```
    return 2;
}
else if(a==4){
    return 3;
}
else if(a==5){
    return 4;
}
else if(a==6){
    return 5;
}
else if(a==7){
    return 6;
}
else if(a==8){
    return 7;
}
else if(a==9){
    return 8;
}
else if(a==10){
    return 9;
}
else if(a==11){
    return 10;
}
else if(a==12){
        return 11;
    }
    return -1;
}
int main() {
    int year = 2024;
    struct tm date = {0};
    date.tm_year = year - 1900;
    date.tm_mon = 0;
    date.tm_mday = 1;

    printf("Enter the date to travel: \n");
    int da;
    scanf("%d", &da);
    int a=choice();
    if(a==-1){
        printf("Enter the correct month: \n");
        return 1;
    }
    date.tm_mon=a;
    mktime(&date);
    printf("Sun Mon Tue Wed Thu Fri Sat\n");
    for (int i = 0; i < date.tm_wday; i++) {
        printf("    ");
    }
    int c=0;
    while (date.tm_mon == a) {
            printf("%3d ", date.tm_mday);
            if(da==date.tm_mday){
                c=1;
            }
            if (date.tm_wday == 6) {
                printf("\n");
            }
            date.tm_mday++;
            mktime(&date);
        }
        if(c==1){
            printf("\nDate Found\n");
        }
        printf("\nCalendar for the year %d:\n\n", year);




        return 0;
    }
```

Output:



```
Enter the Month:
6
==============================================
Sun Mon Tue Wed Thu Fri Sat
                          1
  2   3   4   5   6   7   8
  9  10  11  12  13  14  15
 16  17  18  19  20  21  22
 23  24  25  26  27  28  29
 30
Date Found

Calendar for the year 2024:
```

Day-4

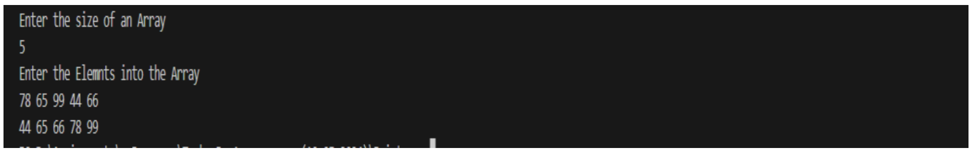
1. Write a program to implement bubble sort.

Code:

```
#include<stdio.h>

void bubblesort(int *a,int n){
    for(int i=0; i<n-1;i++){
        for(int j=0; j<n-i-1; j++){
            if(a[j]>a[j+1]){
                int temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
}
int main(){
    printf("Enter the size of an Array\n");
    int n;
    scanf("%d",&n);
    int a[n];
    printf("Enter the Elemnts into the Array\n");
    for(int i=0; i<n; i++){
        scanf("%d",&a[i]);
    }
    bubblesort(a,n);
    for(int i=0; i<n; i++)
```

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

Output:



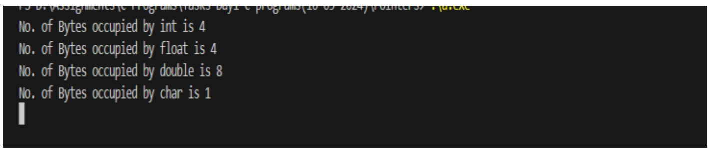
```
Enter the size of an Array
5
Enter the Elemnts into the Array
78 65 99 44 66
44 65 66 78 99
```

2. Write a program to print the size of all data types?

Code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
   printf ("No. of Bytes occupied by int is %d \n", sizeof(int));
   printf ("No. of Bytes occupied by float is %d \n", sizeof(float));
   printf ("No. of Bytes occupied by double is %d \n", sizeof(double));
   printf ("No. of Bytes occupied by char is %d \n", sizeof(char));
   getch();
}
```

Output:



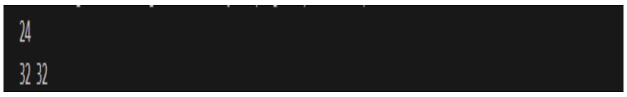
```
No. of Bytes occupied by int is 4
No. of Bytes occupied by float is 4
No. of Bytes occupied by double is 8
No. of Bytes occupied by char is 1
```

3. Write a program to print the address of pointer and value of that pointer.

Code:

```
#include<stdio.h>
int main(){
    int *p,n;
    p=&n;
    n=0x18;
    printf("%d\n",n);
    *p=*p+4;
    n=*p+4;
    printf("%d %d\n",n,*p);
}
```

Output:



```
24
32 32
```

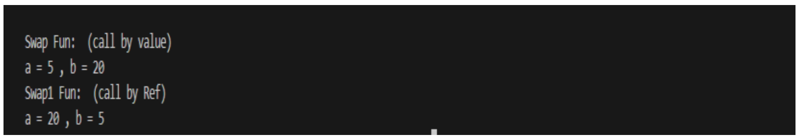
4. **Write a program to swap the Two numbers by using call-by-value and call-by-reference.**

Code:

```
#include <stdio.h>
 void swap(int ,int);
 void swap1(int* ,int*);
int main()
{
    int a,b;
    a=5, b=20;
    swap (a,b);
    printf ("\n Swap Fun:  (call by value)  \n a = %d , b = %d ", a,b);
    swap1 (&a, &b);

    printf ("\n Swap1 Fun:  (call by Ref)  \n a = %d , b = %d ", a,b);
    return 0;


}
```

```
void swap (int x, int y)
{

    int tmp;
    tmp = x;
    x=y;
    y=tmp;

}
void swap1 (int *x1, int *y1)
{

    int tmp1;
    tmp1 = *x1;
    *x1=*y1;
    *y1=tmp1;

}
```

Output:



```
Swap Fun:  (call by value)
a = 5 , b = 20
Swap1 Fun:  (call by Ref)
a = 20 , b = 5
```

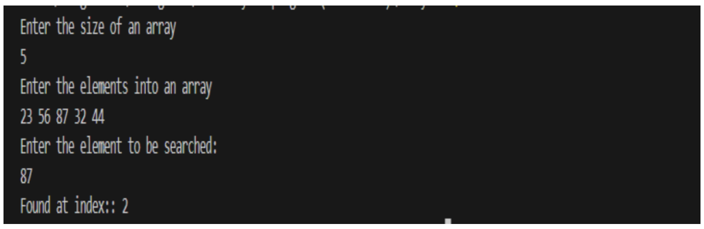
Day-5

1. **Write a program of binary search taking the inputs by the user[Arrays]**

   Code:

```
#include<stdio.h>
int main(){
   printf("Enter the size of an array\n");
   int n;
   scanf("%d",&n);
   int a[n];
   printf("Enter the elements into an array\n");
   for(int i=0; i<n; i++){
      scanf("%d",&a[i]);
   }
   int s=0,e=n-1,mid;
   printf("Enter the element to be searched: \n");
   int k;
   scanf("%d",&k);
   while(s<e){
      int mid=(s+e)/2;
      if(a[mid]==k){
         printf("Found at index:: %d",mid);
         return 1;
      }
      else if(k>a[mid]){
         s=mid+1;
      }
      else
         e=mid-1;
   }
   printf("Element not found\n");
}
```

Output:



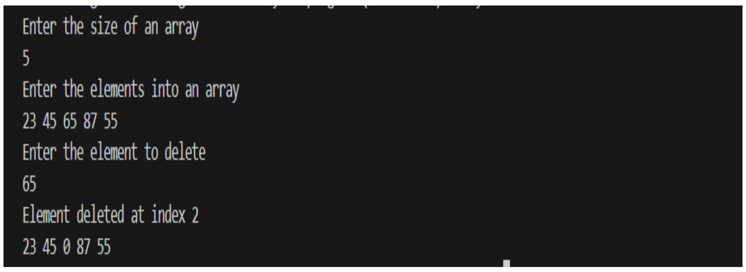
```
Enter the size of an array
5
Enter the elements into an array
23 56 87 32 44
Enter the element to be searched:
87
Found at index:: 2
```

2.Write a program to delete an element in an array.

Code:

```
#include<stdio.h>
int main(){
    printf("Enter the size of an array\n");
    int n;
    scanf("%d",&n);
    int a[n];
    printf("Enter the elements into an array\n");
    for(int i=0; i<n; i++){
        scanf("%d",&a[i]);
    }
    printf("Enter the element to delete\n");
    int del;
    scanf("%d",&del);
    for(int i=0; i<n; i++){
        if(del==a[i]){
            printf("Element deleted at index %d \n", i);
            a[i] = 0;
        }
    }
    for(int i=0; i<n; i++){
        printf("%d ", a[i]);
    }
    return 0;
}
```

Output:



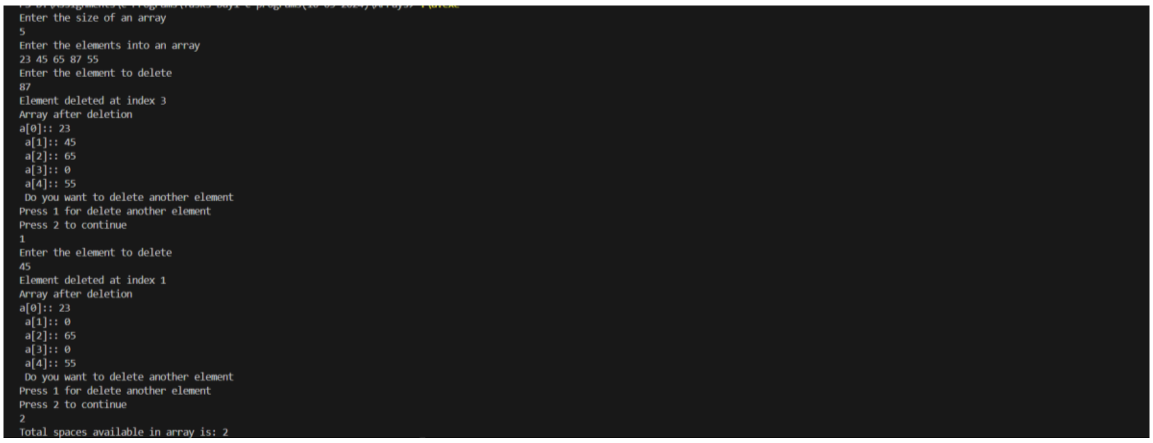
```
Enter the size of an array
5
Enter the elements into an array
23 45 65 87 55
Enter the element to delete
65
Element deleted at index 2
23 45 0 87 55
```

3.write a program to find how many elements delete in the array by asking user behavior.

Code:

```
#include<stdio.h>
int main(){
    printf("Enter the size of an array\n");
    int n;
    scanf("%d",&n);
    int a[n];
    printf("Enter the elements into an array\n");
    for(int i=0; i<n; i++){
        scanf("%d",&a[i]);
    }
    delete:
    printf("Enter the element to delete\n");
    int del;
    scanf("%d",&del);
    for(int i=0; i<n; i++){
        if(del==a[i]){
            printf("Element deleted at index %d \n", i);
            a[i] = 0;
        }
    }
    printf("Array after deletion\n");
    for(int i=0; i<n; i++){
        printf("a[%d]:: %d\n ",i, a[i]);
    }
    printf("Do you want to delete another element\n");
    printf("Press 1 for delete another element\n");
    printf("Press 2 to continue\n");
    int choice;
    scanf("%d", &choice);
    if(choice == 2){
        int c=0;
        for(int i=0; i<n; i++){
            if(a[i]==0)
                c++;
        }
        printf("Total spaces available in array is: %d\n",c);
    }
    else if(choice == 1){
        goto delete;
    }
    return 0;
}
```

Output:



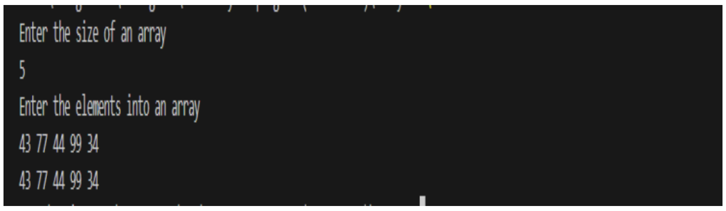
```
Enter the size of an array
5
Enter the elements into an array
23 45 65 87 55
Enter the element to delete
87
Element deleted at index 3
Array after deletion
a[0]:: 23
 a[1]:: 45
 a[2]:: 65
 a[3]:: 0
 a[4]:: 55
 Do you want to delete another element
Press 1 for delete another element
Press 2 to continue
1
Enter the element to delete
45
Element deleted at index 1
Array after deletion
a[0]:: 23
 a[1]:: 0
 a[2]:: 65
 a[3]:: 0
 a[4]:: 55
 Do you want to delete another element
Press 1 for delete another element
Press 2 to continue
2
Total spaces available in array is: 2
```

4. Write a program to print the Elements in an array.

Code:

```
#include<stdio.h>
void print(int a[],int n){
    for(int i=0; i<n; i++){
        printf("%d ", a[i]);
    }
}
int main(){
    printf("Enter the size of an array\n");
    int n;
    scanf("%d",&n);
    int a[n];
    printf("Enter the elements into an array\n");
    for(int i=0; i<n; i++){
        scanf("%d",&a[i]);
    }
    print(a,n);
}
```

Output:



```
Enter the size of an array
5
Enter the elements into an array
43 77 44 99 34
43 77 44 99 34
```

Day-6

Implementation of linked List:

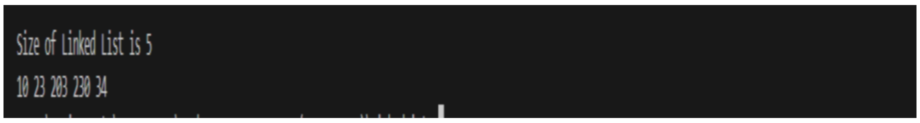
Code:

```
#include <stdio.h>

#include<stdlib.h>

struct Node{

    int data;

    struct Node *next;

};

void display();

struct Node* head;

void insertStart(int data){

    struct Node* nn=(struct Node*)malloc(sizeof(struct Node));

    if(head==NULL){

        nn->data=data;

        nn->next=NULL;

        head=nn;

    }

    else

    {

        nn->data=data;

        nn->next=head;
        head=nn;
    }
}
void insertindex(int index,int data){
    struct Node* nn=(struct Node*)malloc(sizeof(struct Node));
    struct Node* h=head;
    if(index==0)
        insertStart(data);
    else
    {
        nn->data=data;
        for(int i=0;i<index-1;i++){
            h=h->next;
        }
        nn->next=h->next;
        h->next=nn;

    }
}
void display(){
    struct Node* h=head;
    while(h->next!=NULL){
        printf("%d ",h->data);
        h=h->next;
    }
    printf("%d\n",h->data);
    return ;
```

```
}
int size(){

    int c=0;
   struct Node *h=head;
    while(h->next!=NULL){
        c++;
        h=h->next;
    }
    c++;
    return c;
}
void insert(int data){
    struct Node *h=head;
    struct Node *nn=(struct Node*)malloc(sizeof(struct Node));
    if(h==NULL){
        insertStart(data);
        return ;
    }
    while(h->next!=NULL){
        h=h->next;
    }
    nn->data=data;
    h->next=nn;
    nn->next=NULL;
    return ;
}
```

```
int main()
{
    insertStart(10);
    insertindex(1,23);
    insertindex(2,203);
    insertindex(3,230);
    insert(34);
    printf("\nSize of Linked List is %d\n",size());
    display();
    return 0;
}
```

Output:



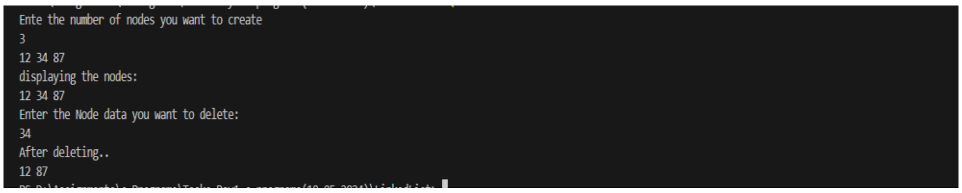
```
Size of Linked List is 5
10 23 203 230 34
```

2. Delete a Node in the List.

Code:

```
#include<stdio.h>
#include<stdlib.h>
struct Node{
    int data;
    struct Node *next;
};
struct Node *head;
void delete(int);
void insert(int data){
    struct Node *h=head;
    struct Node *nn=(struct Node*)malloc(sizeof(struct Node));
    nn->data=data;
    nn->next=NULL;
    if(h==NULL){

        head=nn;
        return ;
    }
    while(h->next!=NULL){
        h=h->next;
    }
    h->next=nn;
    return ;
}
void display(){
    struct Node* h=head;
    while(h->next!=NULL){
        printf("%d ",h->data);
        h=h->next;
    }
    printf("%d\n",h->data);
    return ;
}
void delete(int delete){
    // printf("Entering...");
    struct Node *h=head;
    int index=-1;
    int i=0;
    while(h!=NULL){
        i++;
        if(h->data==delete){
            index=i;
            break;
        }

        h=h->next;
    }
    if(index==-1){
        printf("Element not found\n");
        return ;
    }
    struct Node *h1=head;
    for(int i=1;i<index-1;i++){
        h1=h1->next;
    }
    struct Node *del=h1->next;
    h1->next=del->next;



}
int main(){
    struct Node s;
    printf("Ente the number of nodes you want to create\n");
    int n;
    scanf("%d",&n);
    for(int i=1; i<=n; i++){
        int data;
        scanf("%d",&data);
        insert(data);
    }
    printf("displaying the nodes:\n");
    display();
    printf("Enter the Node data you want to delete:\n");
    int de;
    scanf("%d",&de);
    delete(de);
    printf("After deleting..\n");
    display();
}
```

Output:



```
Ente the number of nodes you want to create
3
12 34 87
displaying the nodes:
12 34 87
Enter the Node data you want to delete:
34
After deleting..
12 87
```

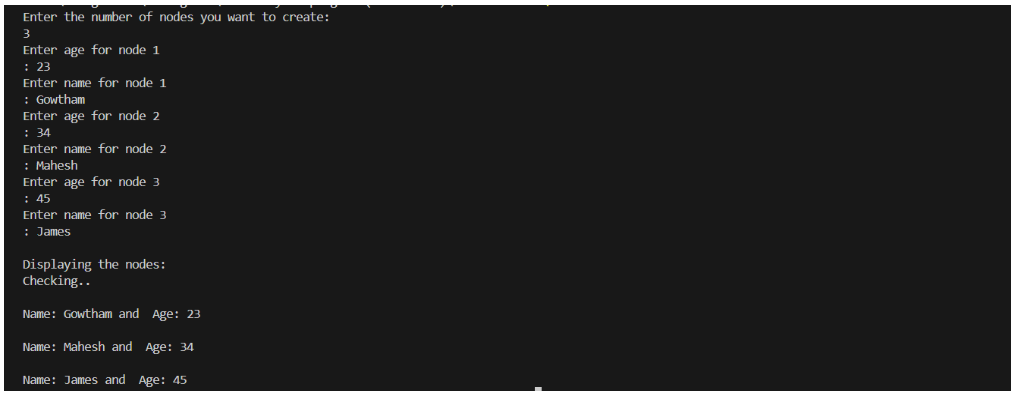
3. **Write a program to create a structure with fields like name and age and insert that date into a nodes.**

Code:

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

struct Node {
    char name[20];
    int age;
    struct Node *next;
};

struct Node *head;

void insert(char name[], int age) {
    struct Node *h = head;
    struct Node *nn = (struct Node*)malloc(sizeof(struct Node));
    nn->age = age;
    strcpy(nn->name, name);
    nn->next = NULL;

    if (h == NULL) {
        head = nn;
        return;
    }

    while (h->next != NULL) {
        h = h->next;
    }

    h->next = nn;
}

void display() {
    printf("Checking..\n");
    struct Node* h = head;
    while (h != NULL) {
        printf("\nName: %s and  Age: %d\n", h->name, h->age);
        h = h->next;
    }
}

int main() {
    printf("Enter the number of nodes you want to create:\n");
    int n;
    scanf("%d", &n);

    for (int i = 1; i <= n; i++) {
        int age;
        char name[20];
        printf("Enter age for node %d\n: ", i);
        scanf("%d", &age);
        printf("Enter name for node %d\n: ", i);
        scanf("%s", name);
        insert(name, age);
    }

    printf("\nDisplaying the nodes:\n");
    display();

    return 0;
}
```

}

Output:



```
Enter the number of nodes you want to create:
3
Enter age for node 1
: 23
Enter name for node 1
: Gowtham
Enter age for node 2
: 34
Enter name for node 2
: Mahesh
Enter age for node 3
: 45
Enter name for node 3
: James

Displaying the nodes:
Checking..

Name: Gowtham and  Age: 23

Name: Mahesh and  Age: 34

Name: James and  Age: 45
```

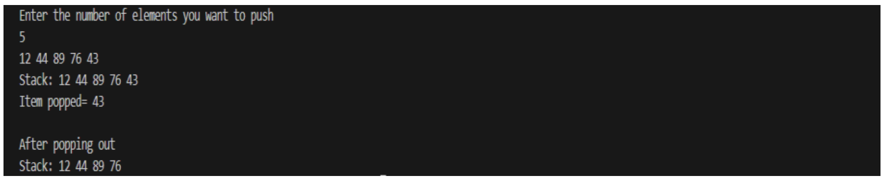
DAY-7

Implementation of Stack:

Code:

```
#include <stdio.h>

#include <stdlib.h>

#define MAX 10


int count = 0;

struct stack {

  int items[MAX];

  int top;

};

typedef struct stack st;


void createEmptyStack(st *s) {

  s->top = -1;

}


int isfull(st *s) {
  if (s->top == MAX - 1)
    return 1;
  else
    return 0;
}

int isempty(st *s) {
  if (s->top == -1)
    return 1;
  else
    return 0;
}

void push(st *s, int newitem) {
  if (isfull(s)) {
    printf("STACK FULL");
  } else {
    s->top++;
    s->items[s->top] = newitem;
  }
  count++;
}

void pop(st *s) {
  if (isempty(s)) {
    printf("\n STACK EMPTY \n");
  } else {
    printf("Item popped= %d", s->items[s->top]);
    s->top--;
  }
  count--;
  printf("\n");
}

void printStack(st *s) {
  printf("Stack: ");
  for (int i = 0; i < count; i++) {
    printf("%d ", s->items[i]);
  }
  printf("\n");
}

int main() {
  int ch;
  st *s = (st *)malloc(sizeof(st));

  createEmptyStack(s);

  printf("Enter the number of elements you want to push\n");
  int n;
  scanf("%d", &n);
  int a;

  for(int i = 0; i < n; i++){
    scanf("%d", &a);
    push(s,a);
  }
//   push(s, 1);
//   push(s, 2);
//   push(s, 3);
//   push(s, 4);

  printStack(s);

  pop(s);

  printf("\nAfter popping out\n");
  printStack(s);
}
```

Output:



```
Enter the number of elements you want to push
5
12 44 89 76 43
Stack: 12 44 89 76 43
Item popped= 43

After popping out
Stack: 12 44 89 76
```

Implementation of Queue:

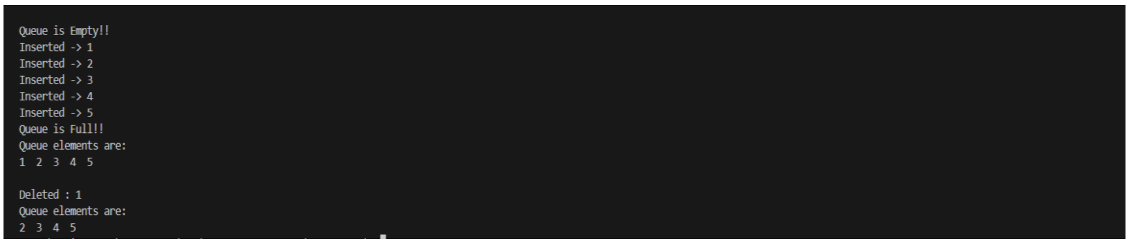
Code:

```
#include <stdio.h>
#define SIZE 5

void enQueue(int);
void deQueue();
void display();
```

```
int items[SIZE], front = -1, rear = -1;

int main() {

  deQueue();

  enQueue(1);
  enQueue(2);
  enQueue(3);
  enQueue(4);
  enQueue(5);

  // 6th element can't be added to because the queue is full
  enQueue(6);

  display();

  //deQueue removes element entered first i.e. 1
  deQueue();

  //Now we have just 4 elements
  display();

  return 0;
}
```

```
void enQueue(int value) {
  if (rear == SIZE - 1)
    printf("\nQueue is Full!!");
  else {
    if (front == -1)
      front = 0;
    rear++;
    items[rear] = value;
    printf("\nInserted -> %d", value);
  }
}

void deQueue() {
  if (front == -1)
    printf("\nQueue is Empty!!");
  else {
    printf("\nDeleted : %d", items[front]);
    front++;
    if (front > rear)
      front = rear = -1;
  }
}

// Function to print the queue
void display() {
  if (rear == -1)
    printf("\nQueue is Empty!!!");
  else {
    int i;
    printf("\nQueue elements are:\n");
    for (i = front; i <= rear; i++)
      printf("%d  ", items[i]);
  }
  printf("\n");
}
```

Output:



```
Queue is Empty!!
Inserted -> 1
Inserted -> 2
Inserted -> 3
Inserted -> 4
Inserted -> 5
Queue is Full!!
Queue elements are:
1  2  3  4  5

Deleted : 1
Queue elements are:
2  3  4  5
```

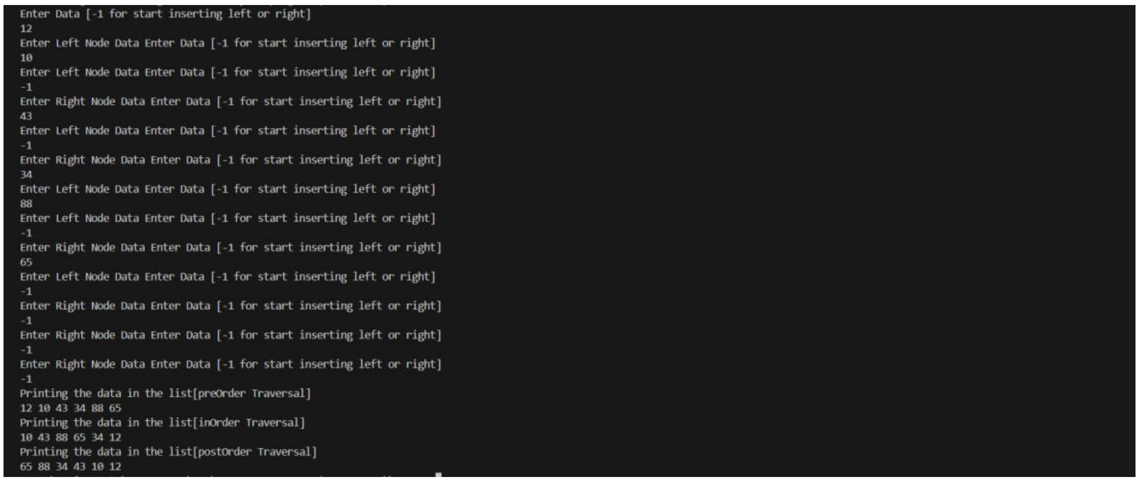
Implementation of Binary Tree:

Code:

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


struct Node {
    int a;
    struct Node *left;
    struct Node *right;
};


struct Node *root = NULL;
```

```c
struct Node* insert() {
    int data;
    struct Node *nn=(struct Node*)malloc(sizeof(struct Node));
    printf("Enter Data [-1 for start inserting left or right]\n");
    scanf("%d", &data);
    if(data == -1)
        return 0;
    nn->a = data;
    printf("Enter Left Node Data ");
    nn->left=insert();
    printf("Enter Right Node Data ");
    nn->right=insert();
    return nn;
}

void preorder(struct Node *root) {
    if (root == NULL) {
        return;
    }
    printf("%d ", root->a);
    preorder(root->left);
    preorder(root->right);
}
void inorder(struct Node *root) {
    if(root == NULL) {
        return ;
    }
    inorder(root->left);
    printf("%d ", root->a);
    inorder(root->right);
}
void postorder(struct Node *root) {
    if(root == NULL) {
        return ;
    }
    postorder(root->left);
    postorder(root->right);
    printf("%d ", root->a);

}
int main() {

    root = insert();
    printf("Printing the data in the list[preOrder Traversal]\n");
    preorder(root);

    printf("\nPrinting the data in the list[inOrder Traversal]\n");
    inorder(root);

    printf("\nPrinting the data in the list[postOrder Traversal]\n");
    postorder(root);

    return 0;
}
```

Output:



```
Enter Data [-1 for start inserting left or right]
12
Enter Left Node Data Enter Data [-1 for start inserting left or right]
10
Enter Left Node Data Enter Data [-1 for start inserting left or right]
-1
Enter Right Node Data Enter Data [-1 for start inserting left or right]
43
Enter Left Node Data Enter Data [-1 for start inserting left or right]
-1
Enter Right Node Data Enter Data [-1 for start inserting left or right]
34
Enter Left Node Data Enter Data [-1 for start inserting left or right]
88
Enter Left Node Data Enter Data [-1 for start inserting left or right]
-1
Enter Right Node Data Enter Data [-1 for start inserting left or right]
65
Enter Left Node Data Enter Data [-1 for start inserting left or right]
-1
Enter Right Node Data Enter Data [-1 for start inserting left or right]
-1
Enter Right Node Data Enter Data [-1 for start inserting left or right]
-1
Enter Right Node Data Enter Data [-1 for start inserting left or right]
-1
Printing the data in the list[preOrder Traversal]
12 10 43 34 88 65
Printing the data in the list[inOrder Traversal]
10 43 88 65 34 12
Printing the data in the list[postOrder Traversal]
65 88 34 43 10 12
```

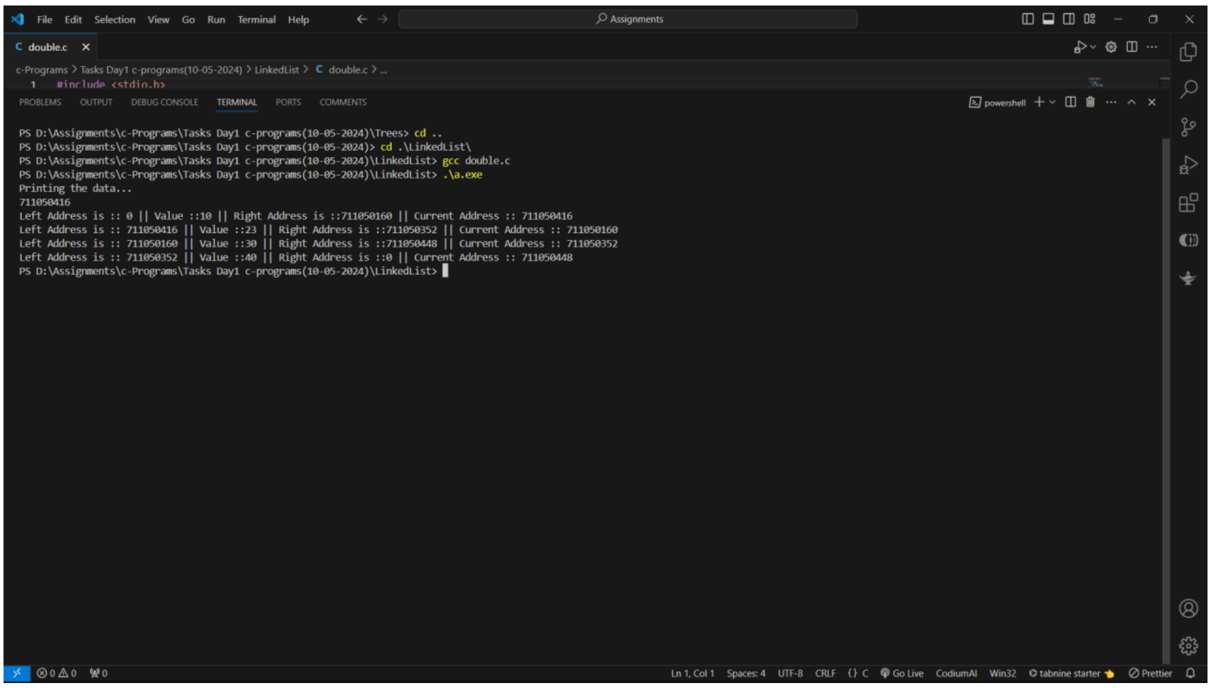
DAY-8

Implementation of Double Linked List:

Code:

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

struct Node {
    int data;
    struct Node *left;
    struct Node *right;
};

struct Node *head;
void create(int data){
    struct Node * nn=(struct Node *)malloc(sizeof(struct Node));
    struct Node *h=head;
    nn->data=data;
    nn->left=NULL;
    nn->right=NULL;
    if(h==NULL){
        head=nn;
        return ;
    }
    while(h->right!=NULL){
        h=h->right;
    }
    h->right=nn;
    nn->left=h;
}
void display(){
    struct Node *h=head;
    printf("%u\n",h);
    while(h!=NULL){
        printf("Left Address is :: %u || Value ::%d || Right Address is ::%u ||
Current Address :: %u\n",h->left,h->data,h->right,h);
        h=h->right;
    }
}
int main(){
    create(10);
    create(23);
    create(30);
    create(40);
    printf("Printing the data...\n");
    display();
}
```

Output:



```
PS D:\Assignments\c-Programs\Tasks Day1 c-programs(10-05-2024)\Trees> cd ..
PS D:\Assignments\c-Programs\Tasks Day1 c-programs(10-05-2024)> cd .\LinkedList\
PS D:\Assignments\c-Programs\Tasks Day1 c-programs(10-05-2024)\LinkedList> gcc double.c
PS D:\Assignments\c-Programs\Tasks Day1 c-programs(10-05-2024)\LinkedList> .\a.exe
Printing the data...
711050416
Left Address is :: 0 || Value ::10 || Right Address is ::711050160 || Current Address :: 711050416
Left Address is :: 711050416 || Value ::23 || Right Address is ::711050352 || Current Address :: 711050160
Left Address is :: 711050160 || Value ::30 || Right Address is ::711050448 || Current Address :: 711050352
Left Address is :: 711050352 || Value ::40 || Right Address is ::0 || Current Address :: 711050448
PS D:\Assignments\c-Programs\Tasks Day1 c-programs(10-05-2024)\LinkedList>
```

DAY-9

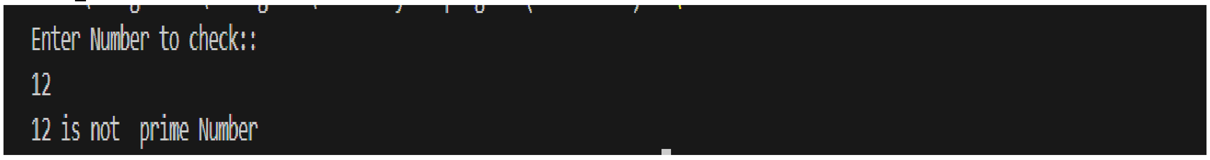
1. Write a Program to check whether the given Number is prime or not?

Code:

```
#include<stdio.h>
int main(){
    printf("Enter Number to check::\n");
    int a;
    scanf("%d",&a);
    prime(a);
}
void prime(int a){
    if(a<=1){
     printf("%d is not prime Number\n",a);
     return ;
    }
```

```
   int c=0;

   for(int i=2;i<a;i++){

      if(a%i==0){

         c++;

      }

   }

   if(c==0){

      printf("%d is prime Number\n",a);

   }

   else

      printf("%d is not  prime Number\n",a);


}
```

Output:



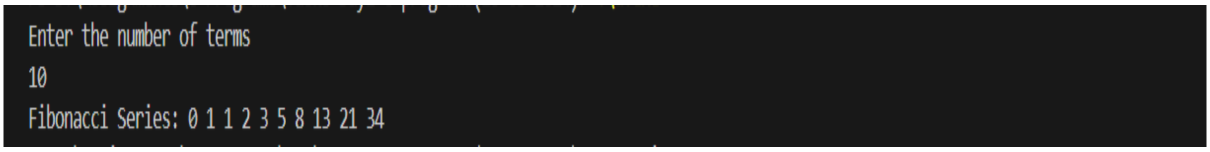
```
Enter Number to check::
12
12 is not  prime Number
```

2. **Write a program to find n Number of fibnocci Numbers by using Recursion.**

Code:

```
#include <stdio.h>

int fibonacci(int n) {
            if(n == 0)
                    return 0;
           else if(n == 1)
                     return 1;
           else
                     return (fibonacci(n-1) + fibonacci(n-2));
}

int main() {
        int n;

        printf("Enter the number of terms\n");
        scanf("%d", &n);

        printf("Fibonacci Series: ");

        for (int i = 0; i < n; i++) {
                printf("%d ", fibonacci(i));
        }

        return 0;
}
```

Output:



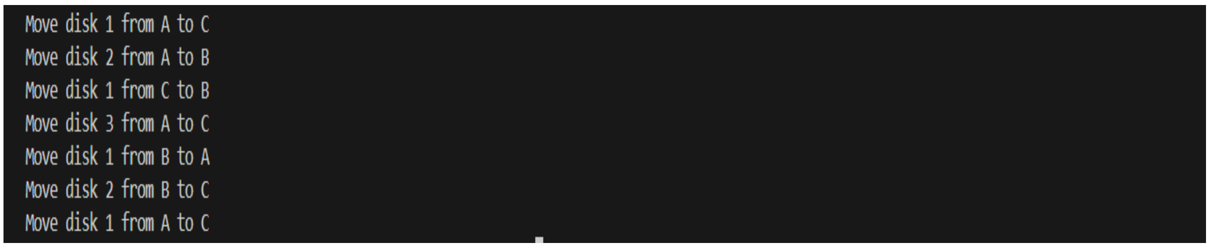
```
Enter the number of terms
10
Fibonacci Series: 0 1 1 2 3 5 8 13 21 34
```

3. Write a program to move the disks which is called as a Tower of Hanoi by using recursion.

Code:

```
#include <stdio.h>
void hanoi(int n, char from, char to, char via) {
  if(n == 1){
    printf("Move disk 1 from %c to %c\n", from, to);
  }
  else{
    hanoi(n-1, from, via, to);
    printf("Move disk %d from %c to %c\n", n, from, to);
    hanoi(n-1, via, to, from);
  }
}
int main() {
  int n = 3;
  char from = 'A';
  char to = 'B';
  char via = 'C';
  hanoi(n, from, via, to);
}
```

Output:



```
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
```

DAY-10

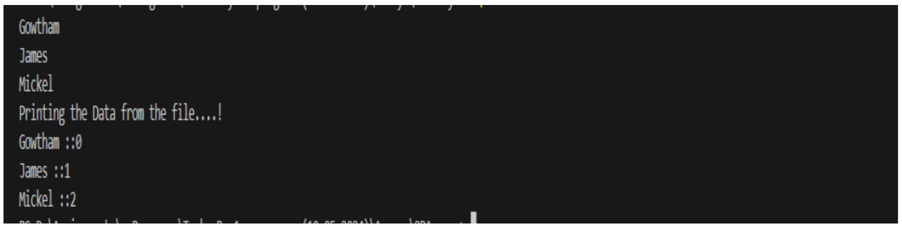
1. **Write a program to ask the user has to enter the 3names and stored it in a file with index number and print the data to console?**

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void file(const char *data) {
    FILE *f = fopen("D://Assignments//2d.txt", "ab");
    if (f == NULL) {
        printf("Error opening the file\n");
        return;
    }
    fwrite(data, sizeof(char), strlen(data), f);
    fwrite("\n", sizeof(char), 1, f);
    fclose(f);
}

void display() {
    char a[21];
    FILE *f = fopen("D://Assignments//2d.txt", "rb");
    if (f == NULL) {
        printf("Error opening the file\n");
        return;
    }
    while (fgets(a, sizeof(a), f) != NULL) {
        printf("%s", a);
    }
    fclose(f);
}
```

```
int main() {
    char a[10][20];
    for (int i = 0; i < 3; i++) {
        scanf("%s", a[i]);
        char k[7];
        snprintf(k, sizeof(k), " ::%d", i);
        strcat(a[i], k);
        file(a[i]);
    }
    printf("Printing the Data from the file....!\n");
    display();
    return 0;
}
```

Output:



```
Gowtham
James
Mickel
Printing the Data from the file....!
Gowtham ::0
James ::1
Mickel ::2
```

2. Write a program to implement AVL tree by using Linked list ?

Code:

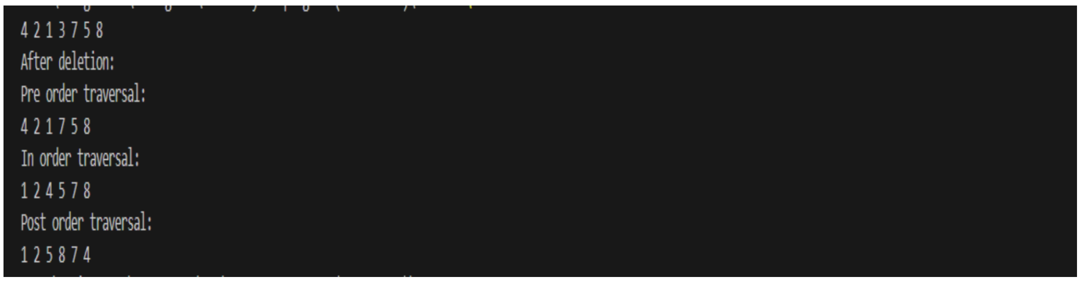
```
#include <stdio.h>
#include <stdlib.h>

struct Node {

  int key;

  struct Node *left;

  struct Node *right;

  int height;

};
```

```
int max(int a, int b);

int height(struct Node *N) {
  if (N == NULL)
    return 0;
  return N->height;
}
int max(int a, int b) {
  return (a > b) ? a : b;
}
struct Node *newNode(int key) {
  struct Node *node = (struct Node *)
    malloc(sizeof(struct Node));
  node->key = key;
  node->left = NULL;
  node->right = NULL;
  node->height = 1;
  return (node);
}
struct Node *rightRotate(struct Node *y) {
  struct Node *x = y->left;
  struct Node *T2 = x->right;
  x->right = y;
  y->left = T2;
  y->height = max(height(y->left), height(y->right)) + 1;
  x->height = max(height(x->left), height(x->right)) + 1;
  return x;
}
struct Node *leftRotate(struct Node *x) {
  struct Node *y = x->right;
  struct Node *T2 = y->left;
  y->left = x;
  x->right = T2;
  x->height = max(height(x->left), height(x->right)) + 1;
  y->height = max(height(y->left), height(y->right)) + 1;
  return y;
}
int getBalance(struct Node *N) {
  if (N == NULL)
    return 0;
  return height(N->left) - height(N->right);
}
struct Node *insertNode(struct Node *node, int key) {
  // Find the correct position to insertNode the node and insertNode it
```

```
if (node == NULL)
  return (newNode(key));
if (key < node->key)
  node->left = insertNode(node->left, key);
else if (key > node->key)
  node->right = insertNode(node->right, key);
else
  return node;
node->height = 1 + max(height(node->left),height(node->right));
int balance = getBalance(node);
if (balance > 1 && key < node->left->key)
  return rightRotate(node);
if (balance < -1 && key > node->right->key)
  return leftRotate(node);
if (balance > 1 && key > node->left->key) {
  node->left = leftRotate(node->left);
  return rightRotate(node);
}
if (balance < -1 && key < node->right->key) {
  node->right = rightRotate(node->right);
  return leftRotate(node);
  }
  return node;
}
struct Node *minValueNode(struct Node *node) {
  struct Node *current = node;
  while (current->left != NULL)
    current = current->left;
  return current;
}

struct Node *deleteNode(struct Node *root, int key) {
  if (root == NULL)
    return root;
  if (key < root->key)
    root->left = deleteNode(root->left, key);
  else if (key > root->key)
    root->right = deleteNode(root->right, key);
  else {
    if ((root->left == NULL) || (root->right == NULL)) {
      struct Node *temp = root->left ? root->left : root->right;
      if (temp == NULL) {
        temp = root;
      root = NULL;
    } else
      *root = *temp;
    free(temp);
  } else {
    struct Node *temp = minValueNode(root->right);
    root->key = temp->key;
    root->right = deleteNode(root->right, temp->key);
  }
}
if (root == NULL)
  return root;
root->height = 1 + max(height(root->left),
             height(root->right));
int balance = getBalance(root);
if (balance > 1 && getBalance(root->left) >= 0)
  return rightRotate(root);
if (balance > 1 && getBalance(root->left) < 0) {
  root->left = leftRotate(root->left);
  return rightRotate(root);
}
if (balance < -1 && getBalance(root->right) <= 0)
```

```
    return leftRotate(root);

  if (balance < -1 && getBalance(root->right) > 0) {

    root->right = rightRotate(root->right);

    return leftRotate(root);

  }

  return root;

}

void printPreOrder(struct Node *root) {

  if (root != NULL) {

    printf("%d ", root->key);

    printPreOrder(root->left);

    printPreOrder(root->right);

  }

}

void inorder(struct Node *root){
    if(root==NULL){
        // printf("The tree is empty\n");
        return ;
    }
    inorder(root->left);
    printf("%d ", root->key);
    inorder(root->right);
}
void postorder(struct Node *root){
    if(root==NULL){
        // printf("The tree is empty\n");
        return ;
    }
    postorder(root->left);
    postorder( root->right);
    printf("%d ",root->key);
}

int main() {

  struct Node *root = NULL;

  root = insertNode(root, 2);

  root = insertNode(root, 1);

  root = insertNode(root, 7);

  root = insertNode(root, 4);

  root = insertNode(root, 5);

  root = insertNode(root, 3);

  root = insertNode(root, 8);

  printPreOrder(root);

  root = deleteNode(root, 3);

  printf("\nAfter deletion: ");

struct Node *r=root;
printf("\nPre order traversal:\n");
  printPreOrder(root);
  printf("\nIn order traversal:\n");
  inorder(root);
  printf("\nPost order traversal:\n");
  postorder(root);
  return 0;

}
```

Output:



```
4 2 1 3 7 5 8
After deletion:
Pre order traversal:
4 2 1 7 5 8
In order traversal:
1 2 4 5 7 8
Post order traversal:
1 2 5 8 7 4
```