CHINMOYEE DASH

DEPT-IT

ROLL-NO-50

SEM-6TH

WEEK-3 CODING

Problem 1:
    C program to convert a binary number to octal number.

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

int main()
{
    int num,rem,i=1,dec=0,j;
    scanf("%d",&num);
    while(num>0)
    {
            rem = num%10;
            if(rem ==0 || rem==1)
            {
                    dec = dec+rem*i;
                    i = i*2;
                    num=num/10;
            }
            else{
                    printf("ERROR");
                    break;
```
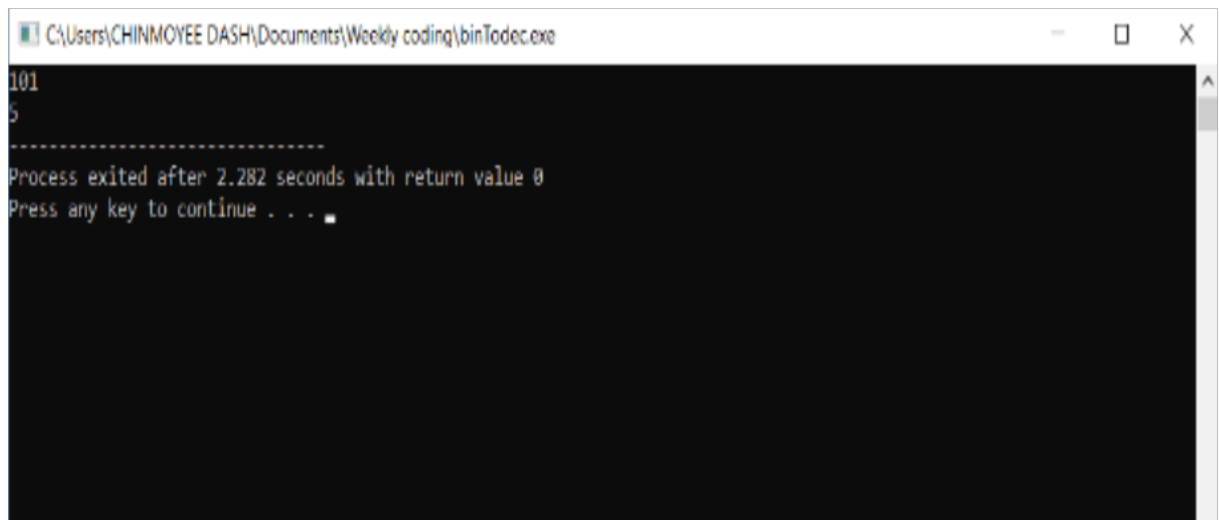
```c
            }
    }
    if(dec!=0)
    {
            printf("%d",dec);
    }
return 0;
}
```


Program 2:
Given an array of integers, sort the first half of the array in ascending order and second half in descending order. Take input from STDIN by scanf().

_____

```c
#include<stdio.h>
int sort(int a[],int n)  //Bubble sort
{
    int i,j,temp;
    for(i=0; i<n; i++)
    {
            for(j=0; j<n-i-1; j++)
```

```c
            {
                    if(a[j]>a[j+1])
                    {
                            temp = a[j];
                            a[j] = a[j+1];
                            a[j+1] = temp;
                    }
            }
        }
}
int main()
{
    int a[100],n,i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
            scanf("%d",&a[i]);
    }
    sort(a,n);
    for(i=0; i<n/2; i++)
    {
            printf("%d\t",a[i]);
    }
    for(i=n-1; i>=n/2; i--)
    {
            printf("%d\t",a[i]);
    }
```
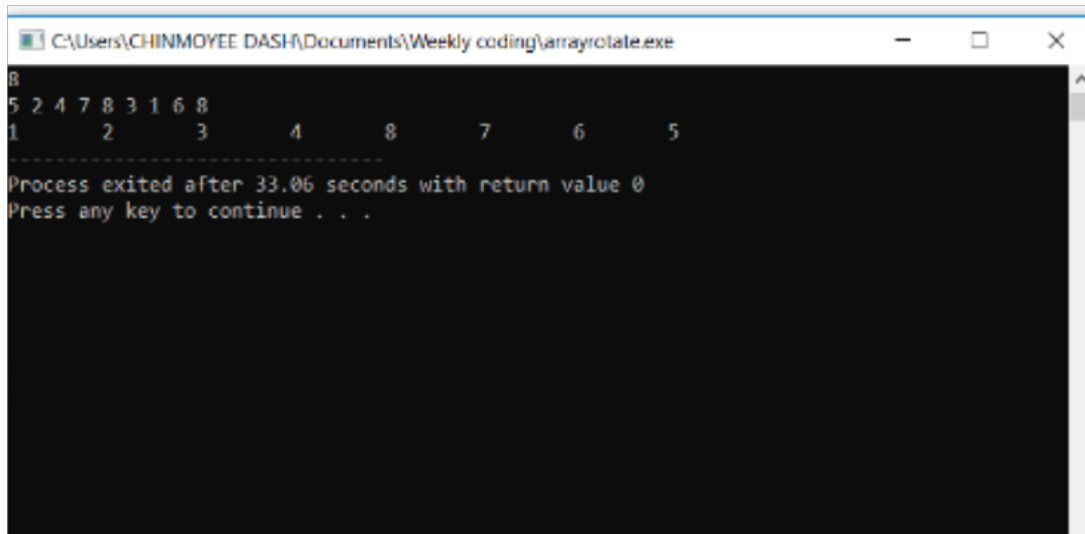
```
        return 0;

    }
```



```
C:\Users\CHINMOYEE DASH\Documents\Weekly coding\arrayrotate.exe          —    □    ×
8
5 2 4 7 8 3 1 6 8
1        2        3        4        8        7        6        5
------------------------------------
Process exited after 33.06 seconds with return value 0
Press any key to continue . . .
```

Program 3:
A prime number is a whole number greater than 1, whose only two whole-number factors are 1 and itself. The first few prime numbers are 2, 3, 5, 7, 11 and 13. Given two integers, print the sum of all prime numbers between n1 and n2 (both inclusive).

```c
#include<stdio.h>
int isprime(int j) {
    int count=0,i;
    for( i = 2 ; i <= j/2; i++) {
        if(j%i == 0) {
            count = 1;
        }
    }
    if(count == 0) {
        return 1;
    }
    else
```

```c
    return 0;
}

int main()
{
int n1,n2,i,j,sum=0,T;
scanf("%d",&T);
while(T-->0)
{
            printf("Enter the start interval: ");
            scanf("%d",&n1);
            printf("Enter the end interval: ");
            scanf("%d",&n2);
            if(n1>=0 && n2>=0 && n1<=100 && n2<=100)
            {
                if(n1==n2)
                {
                printf("0");
                }
                else{
                        for(i=n1; i<=n2; i++)
                        {
                                if(isprime(i)==1)
                                {
                                sum = sum + i;
                                }
                        }
```

```c
                    if(n1 == 1)
                            printf("Sum is: %d\t",sum-1);
                    else
                            printf("Sum is: %d\t",sum);
            }
            printf("\n");
        }
        else
            printf("ERROR");
}


return 0;
}
```
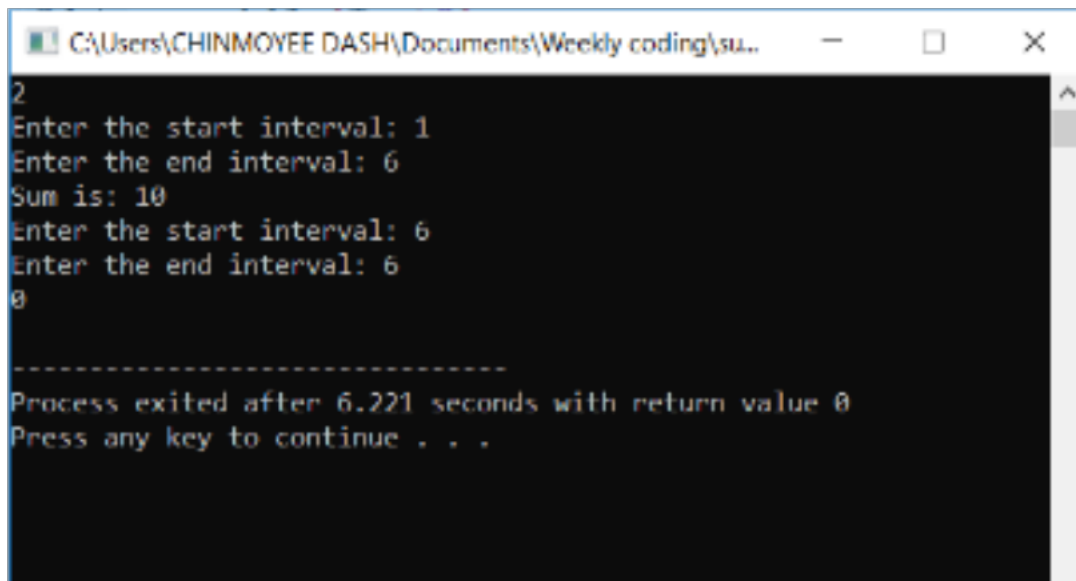


```
C:\Users\CHINMOYEE DASH\Documents\Weekly coding\su...        —     □     ×
2
Enter the start interval: 1
Enter the end interval: 6
Sum is: 10
Enter the start interval: 6
Enter the end interval: 6
0

--------------------------------
Process exited after 6.221 seconds with return value 0
Press any key to continue . . .
```
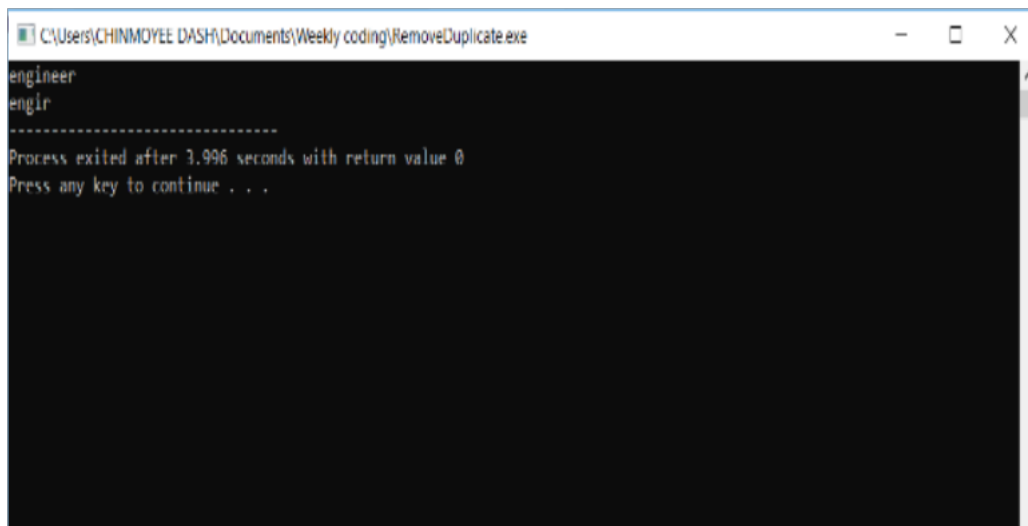
Program 4:
Remove duplicates character from a given string.

//Q.4) Remove the duplicate from string

```c
//------------------------------

#include<stdio.h>
#include<string.h>
int main()
{
        char str[100],c='-';
scanf("%s",str);
int i,j;
int length = strlen(str);
//printf("%d",length);
for(i=0; i<length; i++)
{
        for(j=i+1; j<length; j++)
        {
                if(str[i]==str[j])
                {
                str[j]=c;
                }
        }
        if(str[i]!='-')
            printf("%c",str[i]);
}
return 0;
}
```

Program 5

C Program to find sum of series 1 + 1/2 + 1/3 + 1/4 +... + 1/n. The value n is positive integer
passed to the program as the first command line parameter. Write the output to stdout formatted as a floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text. Scientific format (such as 1.00E+5) should NOT be used while printing the output. You may assume that the inputs will be such that the output will not exceed the largest possible real number that can be stored in a float type variable.

―――

```c
#include<stdio.h>

int main()
{
int n;
scanf("%d",&n);
float sum=0.00,i;
for(i=1; i<=n; i++)
{
	//n = float(n);
	sum = (sum+(1.0/i));
}
```
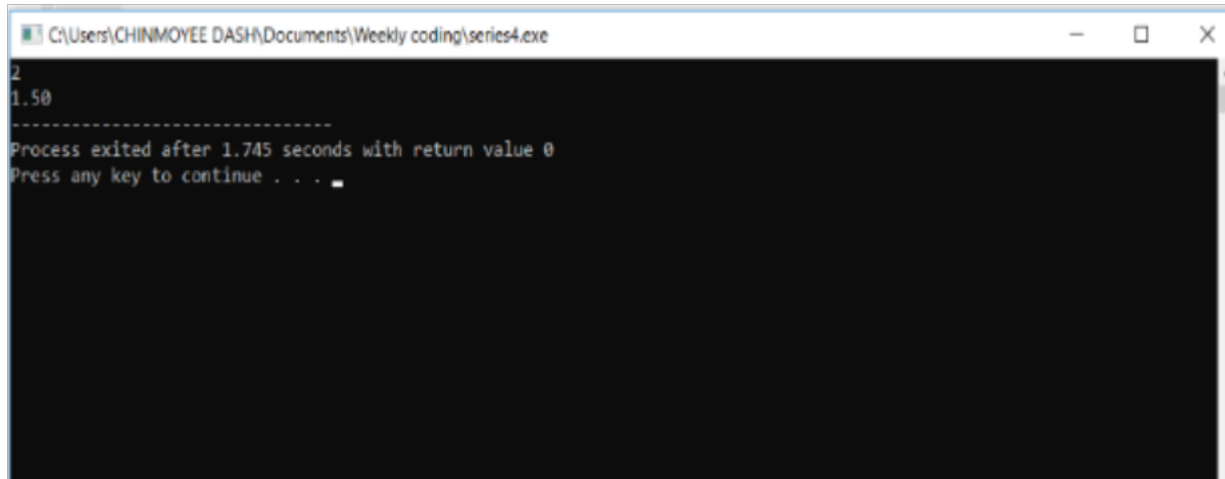
```
printf("%.2f",sum);

return 0;

}
```

```
2
1.50
---------------------------------
Process exited after 1.745 seconds with return value 0
Press any key to continue . . .
```