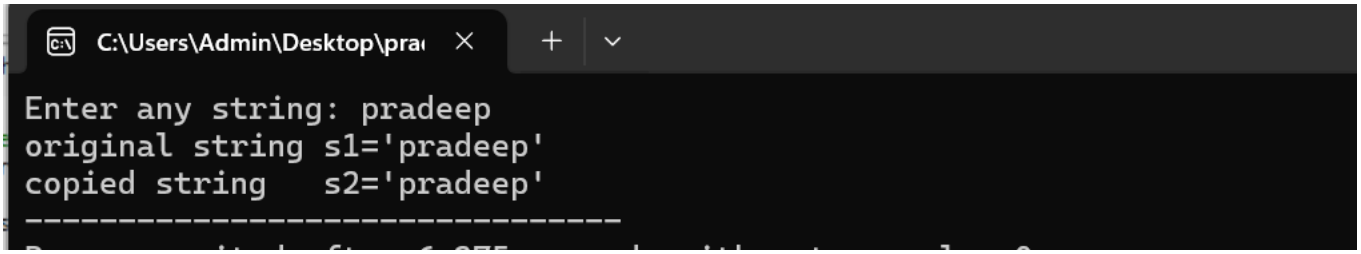
DAY2

1.Write C program for copy string to another string using recursive?

Program:

```
#include <stdio.h>
void stringcopy(char *s1,char *s2,int i)
{
   if(s1[i]=='\0')
   {
         s2[i]='\0';
         return;
         }
         else
         {
                  s2[i]=s1[i];
       stringcopy(s1,s2,++i);
   }
}
int main()
{
   char s1[1000],s2[1000];
   int i;
printf("Enter any string: ");
   gets(s1);
   stringcopy(s1,s2,0);
    printf("original string s1='%s'\n",s1);
   printf("copied string   s2='%s'",s2);
}
```

Output:



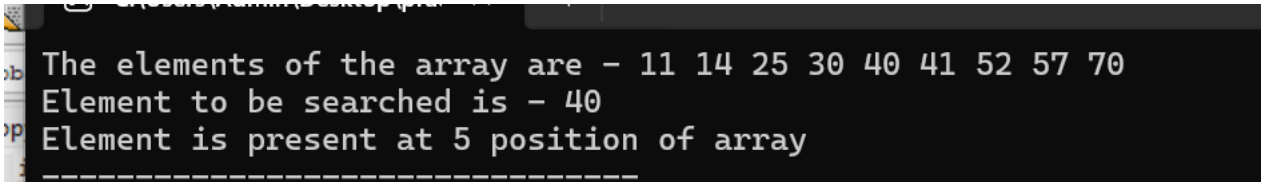
2.write C program for Binary search?

Program:

```
#include <stdio.h>
int binarySearch(int a[], int beg, int end, int val)
{
int mid;
if(end >= beg)
{ mid = (beg + end)/2;
if(a[mid] == val)
{
return mid+1;
}
else if(a[mid] < val)
{
return binarySearch(a, mid+1, end, val);
}

else
{
return binarySearch(a, beg, mid-1, val);
}
}
return -1;
}
int main() {
```

```
int a[] = {11, 14, 25, 30, 40, 41, 52, 57, 70};
int val = 40;
int n = sizeof(a) / sizeof(a[0]);
int res = binarySearch(a, 0, n-1, val);
printf("The elements of the array are - ");
for (int i = 0; i < n; i++)
printf("%d ", a[i]);
printf("\nElement to be searched is - %d", val);
if (res == -1)
printf("\nElement is not present in the array");
else
printf("\nElement is present at %d position of array", res);
return 0;
}
```

Output:

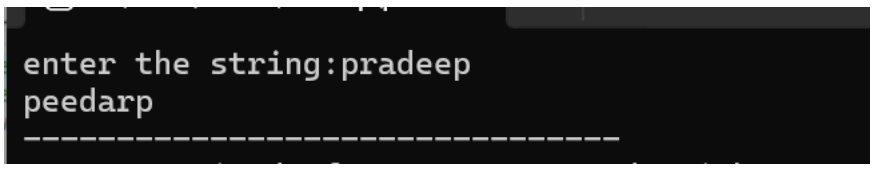


3.write C program for reverse a string using recursive?

Program:

```
#include<stdio.h>
#include<string.h>
void rev(char s[],char s1[],int i){
static   int j=0;
         if(i<=-1)
         {
                  return;
         }
         s1[j]=s[i];
         j++;
         return rev(s,s1,i-1);
}
int main(){
         char s[10];
         char s1[10];
         int n;
         printf("enter the string:");
         scanf("%s",s);
         n=strlen(s);
         rev(s,s1,n-1);
         printf(s1);
}
```

Output:



4. write C program for Strassen's matrix?

Program:

```
#include<stdio.h>
int main(){
  int a[2][2], b[2][2], c[2][2], i, j;
  int m1, m2, m3, m4 , m5, m6, m7;

  printf("Enter the 4 elements of first matrix: ");
  for(i = 0;i < 2; i++)
     for(j = 0;j < 2; j++)
          scanf("%d", &a[i][j]);

  printf("Enter the 4 elements of second matrix: ");
  for(i = 0; i < 2; i++)
     for(j = 0;j < 2; j++)
          scanf("%d", &b[i][j]);

  printf("\nThe first matrix is\n");
  for(i = 0; i < 2; i++){
     printf("\n");
     for(j = 0; j < 2; j++)
          printf("%d\t", a[i][j]);
  }

  printf("\nThe second matrix is\n");
  for(i = 0;i < 2; i++){
     printf("\n");
     for(j = 0;j < 2; j++)
          printf("%d\t", b[i][j]);
```

```
  }
  m1= (a[0][0] + a[1][1]) * (b[0][0] + b[1][1]);
  m2= (a[1][0] + a[1][1]) * b[0][0];
  m3= a[0][0] * (b[0][1] - b[1][1]);
  m4= a[1][1] * (b[1][0] - b[0][0]);
  m5= (a[0][0] + a[0][1]) * b[1][1];
  m6= (a[1][0] - a[0][0]) * (b[0][0]+b[0][1]);
  m7= (a[0][1] - a[1][1]) * (b[1][0]+b[1][1]);
  c[0][0] = m1 + m4- m5 + m7;
  c[0][1] = m3 + m5;
  c[1][0] = m2 + m4;
  c[1][1] = m1 - m2 + m3 + m6;

   printf("\nAfter multiplication using Strassen's algorithm \n");
   for(i = 0; i < 2 ; i++){
      printf("\n");
      for(j = 0;j < 2; j++)
          printf("%d\t", c[i][j]);
   }

   return 0;
}
```

Output:

```
Enter the 4 elements of first matrix: 1 2 3 4
Enter the 4 elements of second matrix: 1
1 1 1

The first matrix is

1       2
3       4
The second matrix is

1       1
1       1
After multiplication using Strassen's algorithm

3       3
7       7
```

5.write C program for merge sort?

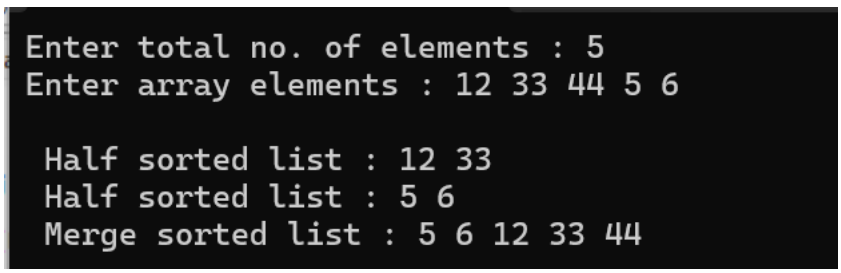
Program:

```
#include <stdio.h>
#include <conio.h>
void merge(int [],int ,int ,int );
void part(int [],int ,int );
int size;
int main()
{
int i, arr[30];
printf("Enter total no. of elements : ");
scanf("%d", &size);
printf("Enter array elements : ");
for(i=0; i<size; i++)
scanf("%d", &arr[i]);
part(arr, 0, size-1);
printf("\n Merge sorted list : ");
for(i=0; i<size; i++)
printf("%d ",arr[i]);
return 0;
}
void part(int arr[], int min, int max)
{
int mid,i;
if(min < max)
{
mid = (min + max) / 2;
part(arr, min, mid);
part(arr, mid+1, max);
```

```
merge(arr, min, mid, max);
}
if (max-min == (size/2)-1)
{
printf("\n Half sorted list : ");
for(i=min; i<=max; i++)
printf("%d ", arr[i]);
}
}
void merge(int arr[],int min,int mid,int max)
{
int tmp[30];
int i, j, k, m;
j = min;
m = mid + 1;
for(i=min; j<=mid && m<=max; i++)
{
if(arr[j] <= arr[m])
{
tmp[i] = arr[j];
j++;
}
else
{
tmp[i] = arr[m];
m++;
}
}
if(j > mid)
{
```

```
for(k=m; k<=max; k++)
{
tmp[i] = arr[k];
i++;
}
}
else
{
for(k=j; k<=mid; k++)
{
tmp[i] = arr[k];
i++;
}
}
for(k=min; k<=max; k++)
arr[k] = tmp[k];
}
```

Output:



6.write C program for Using Divide and Conquer strategy to find Max and Min value in the list.?

Program:

```
#include<stdio.h>
#include<stdio.h>
int max, min;
int a[100];
void maxmin(int i, int j)
{
 int max1, min1, mid;
 if(i==j)
 {
  max = min = a[i];
 }
 else
 {
  if(i == j-1)
  {
   if(a[i] <a[j])
   {
    max = a[j];
    min = a[i];
   }
   else
   {
    max = a[i];
    min = a[j];
   }
  }
 else
  mid = (i+j)/2;
```

```
  maxmin(i, mid);
  max1 = max; min1 = min;
  maxmin(mid+1, j);
  if(max <max1)
   max = max1;
  if(min > min1)
   min = min1;
 }
}
}
int main ()
{
 int i, num;
 printf ("\nEnter the total number of numbers : ");
 scanf ("%d",&num);
 printf ("Enter the numbers : \n");
 for (i=1;i<=num;i++)
  scanf ("%d",&a[i]);
 max = a[0];
 min = a[0];
 maxmin(1, num);
 printf ("Minimum element in an array : %d\n", min);
 printf ("Maximum element in an array : %d\n", max);
 return 0;
}
```

Output:

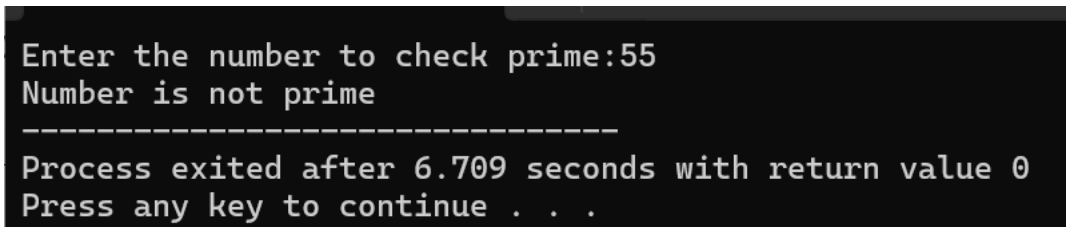
```
Enter the total number of numbers : 5
Enter the numbers :
12
2
0
1
33
Minimum element in an array : 0
Maximum element in an array : 33
```

7. write C program for prime number?

Program:

```
a#include<stdio.h>
int main(){
int n,i,m=0,flag=0;
printf("Enter the number to check prime:");
scanf("%d",&n);
m=n/2;
for(i=2;i<=m;i++)
{
if(n%i==0)
{
printf("Number is not prime");
flag=1;
break;
}
}
if(flag==0)
printf("Number is prime");
return 0;
 }
```

Output:



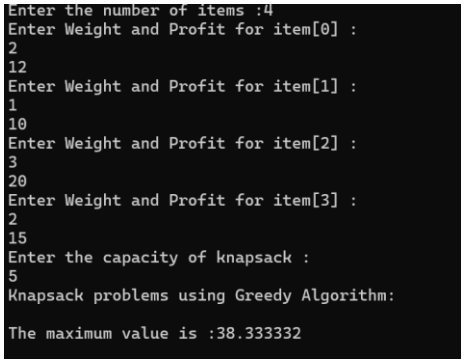
8. Write C a program to perform Knapsack problem using greedy techniques.

Program:

```
#include<stdio.h>
int main()
{
    float weight[50],profit[50],ratio[50],Totalvalue,temp,capacity,amount;
    int n,i,j;
    printf("Enter the number of items :");
    scanf("%d",&n);
   for (i = 0; i < n; i++)
   {
      printf("Enter Weight and Profit for item[%d] :\n",i);
      scanf("%f %f", &weight[i], &profit[i]);
   }
   printf("Enter the capacity of knapsack :\n");
   scanf("%f",&capacity);

    for(i=0;i<n;i++)
        ratio[i]=profit[i]/weight[i];

   for (i = 0; i < n; i++)
     for (j = i + 1; j < n; j++)
        if (ratio[i] < ratio[j])
        {
           temp = ratio[j];
           ratio[j] = ratio[i];
           ratio[i] = temp;

           temp = weight[j];
           weight[j] = weight[i];
```

```
            weight[i] = temp;

            temp = profit[j];
            profit[j] = profit[i];
            profit[i] = temp;
          }

    printf("Knapsack problems using Greedy Algorithm:\n");
    for (i = 0; i < n; i++)
    {
     if (weight[i] > capacity)
        break;
      else
     {
        Totalvalue = Totalvalue + profit[i];
        capacity = capacity - weight[i];
      }
    }
      if (i < n)
      Totalvalue = Totalvalue + (ratio[i]*capacity);
    printf("\nThe maximum value is :%f\n",Totalvalue);
    return 0;
}
```

Output:



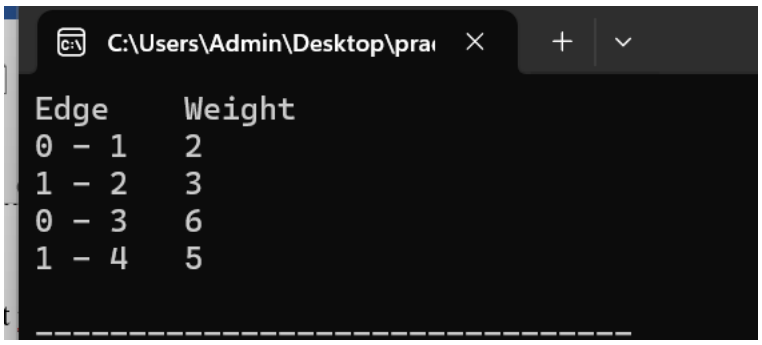
9.write C program for MST greedy techniques?

Program:

```
#include <limits.h>
#include <stdbool.h>
#include <stdio.h>
#define V 5
int minKey(int key[], bool mstSet[])
{
int min = INT_MAX, min_index;
for (int v = 0; v < V; v++)
if (mstSet[v] == false && key[v] < min)
min = key[v], min_index = v;
return min_index;
}
int printMST(int parent[], int graph[V][V])
{
printf("Edge \tWeight\n");
for (int i = 1; i < V; i++)
printf("%d - %d \t%d \n", parent[i], i,
graph[i][parent[i]]);
}
void primMST(int graph[V][V])
{
int parent[V];
int key[V];
bool mstSet[V];
for (int i = 0; i < V; i++)
key[i] = INT_MAX, mstSet[i] = false;
key[0] = 0;
parent[0] = -1;
```

```
for (int count = 0; count < V - 1; count++) {
int u = minKey(key, mstSet);
mstSet[u] = true;
for (int v = 0; v < V; v++)
if (graph[u][v] && mstSet[v] == false
&& graph[u][v] < key[v])
parent[v] = u, key[v] = graph[u][v];
}
printMST(parent, graph);
}
int main()
{
int graph[V][V] = { { 0, 2, 0, 6, 0 },
{ 2, 0, 3, 8, 5 },
{ 0, 3, 0, 0, 7 },
{ 6, 8, 0, 0, 9 },
{ 0, 5, 7, 9, 0 } };
primMST(graph);
return 0;
}
```

Output:

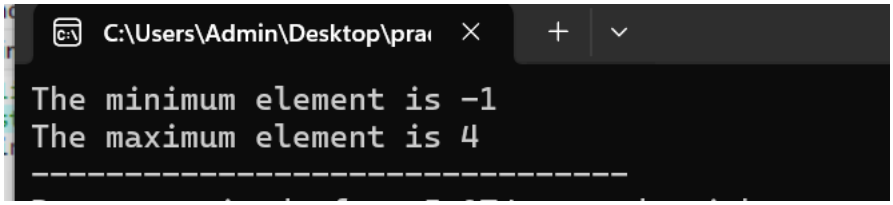


10. Write a program to print minimum and maximum value sequency for all the numbers in a list.

Program:

```
#include <limits.h>
#include <stdio.h>
void findMinimumMaximum(int arr[], int N)
{
    int i;
    int minE = INT_MAX, maxE = INT_MIN;
    for (i = 0; i < N; i++) {
        if (arr[i] < minE) {
            minE = arr[i];
        }
        if (arr[i] > maxE) {
            maxE = arr[i];
        }
    }
    printf("The minimum element is %d", minE);
    printf("\n");
    printf("The maximum element is %d", maxE);

    return;
}
int main()
{
    int arr[] = { 1, 2, 4, -1 };
    int N = sizeof(arr) / sizeof(arr[0]);
    findMinimumMaximum(arr, N);

    return 0;
}
```

Output:


C:\Users\Admin\Desktop\pra
The minimum element is -1
The maximum element is 4