```c
main.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  clock_t start, end;
6  double cpu_time;
7
8  void swap(int *xp, int *yp)
9  {
10     int temp = *xp;
11     *xp = *yp;
12     *yp = temp;
13  }
14
15  void selectionSort(int arr[], int n)
16  {
17     int i, j, min_idx;
18
19     for (i = 0; i < n-1; i++)
20     {
21         min_idx = i;
22         for (j = i+1; j < n; j++)
23         if (arr[j] < arr[min_idx])
24             min_idx = j;
25
26         swap(&arr[min_idx], &arr[i]);
27     }
28  }
29
30  int linear_search(int arr[],int key,int i,int n)
31  {
32     if(arr[i]==key)
33         return i+1;
34      else if(i>=n)
35         return -1;
36     else
37        return linear_search(arr,key,i+1,n);
38  }
```

```c
          return linear_search(arr,key,l+1,n);
38  }
39
40  void binary_search(int arr[],int l,int r,int key)
41  {
42      if(l>r)
43      {
44        printf("Element not present");
45        return;
46      }
47      int mid=l + (r - l) / 2;
48      if(arr[mid]==key)
49          printf("Element found at index:%d \n",mid +1);
50      else if(key<arr[mid])
51          return binary_search(arr,l,mid-1,key);
52      else if(key>arr[mid])
53          return binary_search(arr,mid+1,r,key);
54  }
55
56  int main()
57  {
58      int  r, k, n, ch, flag, key, l, mid, arr[10000];
59      printf("Enter the number of elements in array \n");
60      scanf("%d", &n);
61
62      for(k=0;k<n;k++)
63          arr[k]=rand()%n+1;
64
65      printf("Elements of the array are:\n");
66
67      for(k=0;k<n;k++){
68          printf("%d ",arr[k]);
69      }
70      printf("\n");
71
72      printf("RECURSIVE SEARCH functions:\n");
73      printf("1.LINEAR SEARCH:\n");
74      printf("2.BINARY SEARCH:\n");
75      printf("Enter your choice:");
```

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

printf("RECURSIVE SEARCH functions:\n");
printf("1.LINEAR SEARCH:\n");
printf("2.BINARY SEARCH:\n");
printf("Enter your choice:");
scanf("%d",&ch);

switch(ch)
{
case 1:
    printf("Enter the element to be searched:");
    scanf("%d",&key);

    start = clock();
    flag=linear_search(arr,key,0,n);
    for (int c = 1; c <= 5000; c++)
        for (int d = 1; d <= 5000; d++) { }
    end = clock();
    cpu_time = (double)(end-start)/CLOCKS_PER_SEC;

    if(flag==-1)
        printf("Element not found \n");
    else
        printf("Element found at position:%d \n",flag);

    printf("Execution time for linear_search = %f seconds\n", cpu_time);
    break;
case 2:
    printf("Enter the element to be searched:");
    scanf("%d",&key);

    start = clock();
    // sorting array for binary search
    selectionSort(arr, n);
    printf("Elements of the sorted array are:\n");
    for(k=0;k<n;k++){
        printf("%d ",arr[k]);
```

```c
        start = clock();
        flag=linear_search(arr,key,0,n);
        for (int c = 1; c <= 5000; c++)
            for (int d = 1; d <= 5000; d++) { }
        end = clock();
        cpu_time = (double)(end-start)/CLOCKS_PER_SEC;

        if(flag==-1)
            printf("Element not found \n");
        else
            printf("Element found at position:%d \n",flag);

        printf("Execution time for linear_search = %f seconds\n", cpu_time);
        break;
    case 2:
        printf("Enter the element to be searched:");
        scanf("%d",&key);

        start = clock();
        // sorting array for binary search
        selectionSort(arr, n);
        printf("Elements of the sorted array are:\n");
        for(k=0;k<n;k++){
            printf("%d ",arr[k]);
        }
        printf("\n");
        binary_search(arr,0,n,key);

        for (int c = 1; c <= 5000; c++)
            for (int d = 1; d <= 5000; d++) { }
        end = clock();
        cpu_time = (double)(end-start)/CLOCKS_PER_SEC;
        printf("Execution time for binary_search = %f seconds\n", cpu_time);

        break;
    }
    return 0;
}
```

```
Enter the number of elements in array
50
Elements of the array are:
34 37 28 16 44 36 37 43 50 22 13 28 41 10 14 27 41 27 23 37 12 19 18 30 33 31 13 24 18 36 30 3 23 9 20 18 44 7 12 43 30 24 22 20 35 38 49 25 16 21
RECURSIVE SEARCH functions:
1.LINEAR SEARCH:
2.BINARY SEARCH:
Enter your choice:1
Enter the element to be searched:43
Element found at position:8
Execution time for linear_search = 0.055775 seconds


...Program finished with exit code 0
Press ENTER to exit console.
```

```
Enter the number of elements in array
50
Elements of the array are:
34 37 28 16 44 36 37 43 50 22 13 28 41 10 14 27 41 27 23 37 12 19 18 30 33 31 13 24 18 36 30 3 23 9 20 18 44 7 12 43 30 24 22 20 35 38 49 25 16 21
RECURSIVE SEARCH functions:
1.LINEAR SEARCH:
2.BINARY SEARCH:
Enter your choice:2
Enter the element to be searched:37
Elements of the sorted array are:
3 7 9 10 12 12 13 13 14 16 16 18 18 18 19 20 20 21 22 22 23 23 24 24 25 27 27 28 28 30 30 30 31 33 34 35 36 36 37 37 37 38 41 41 43 43 44 44 49 50
Element found at index:39
Execution time for binary_search = 0.056555 seconds


...Program finished with exit code 0
Press ENTER to exit console.
```

```c
main.c

1  #include <stdio.h>
2  #include <stdlib.h>
3  int iterative(int a,int b)
4  {
5      int q,r;
6      while(b>0)
7      {
8          q=a/b;
9          r=a%b;
10         a=b;
11         b=r;
12     }
13     return a;
14 }
15 int recursive(int a,int b)
16 {
17     if(b==0)
18     {
19         return a;
20     }
21     int q=a/b;
22     int r=a%b;
23     return(b,r);
24 }
25 int main()
26 {
27     int n,a,b,gcd;
28     printf("enter any two positive numbers\n");
29     scanf("%d%d",&a,&b);
30     for(;;)
31     {
32         printf("Enter any choice:\n1.iterative\n2.recursive\n3.exit\n");
33         scanf("%d",&n);
34         switch(n)
35         {
36         case 1:gcd=iterative(a,b);
37         printf("gcd is %d\n",gcd);
38         break;
```

```c
        q=a/b;
        r=a%b;
        a=b;
        b=r;
    }
    return a;
}
int recursive(int a,int b)
{
    if(b==0)
    {
        return a;
    }
    int q=a/b;
    int r=a%b;
    return(b,r);
}
int main()
{
    int n,a,b,gcd;
    printf("enter any two positive numbers\n");
    scanf("%d%d",&a,&b);
    for(;;)
    {
        printf("Enter any choice:\n1.iterative\n2.recursive\n3.exit\n");
        scanf("%d",&n);
        switch(n)
        {
        case 1:gcd=iterative(a,b);
        printf("gcd is %d\n",gcd);
        break;
        case 2:gcd=recursive(a,b);
        printf("gcd is %d\n",gcd);
        break;
        default:exit(0);
        }
    }
}
```

```
enter any two positive numbers
60 24
Enter any choice:
1.iterative
2.recursive
3.exit
1
gcd is 12
Enter any choice:
1.iterative
2.recursive
3.exit
2
gcd is 12
Enter any choice:
1.iterative
2.recursive
3.exit
3


...Program finished with exit code 0
Press ENTER to exit console.
```