Practical 1

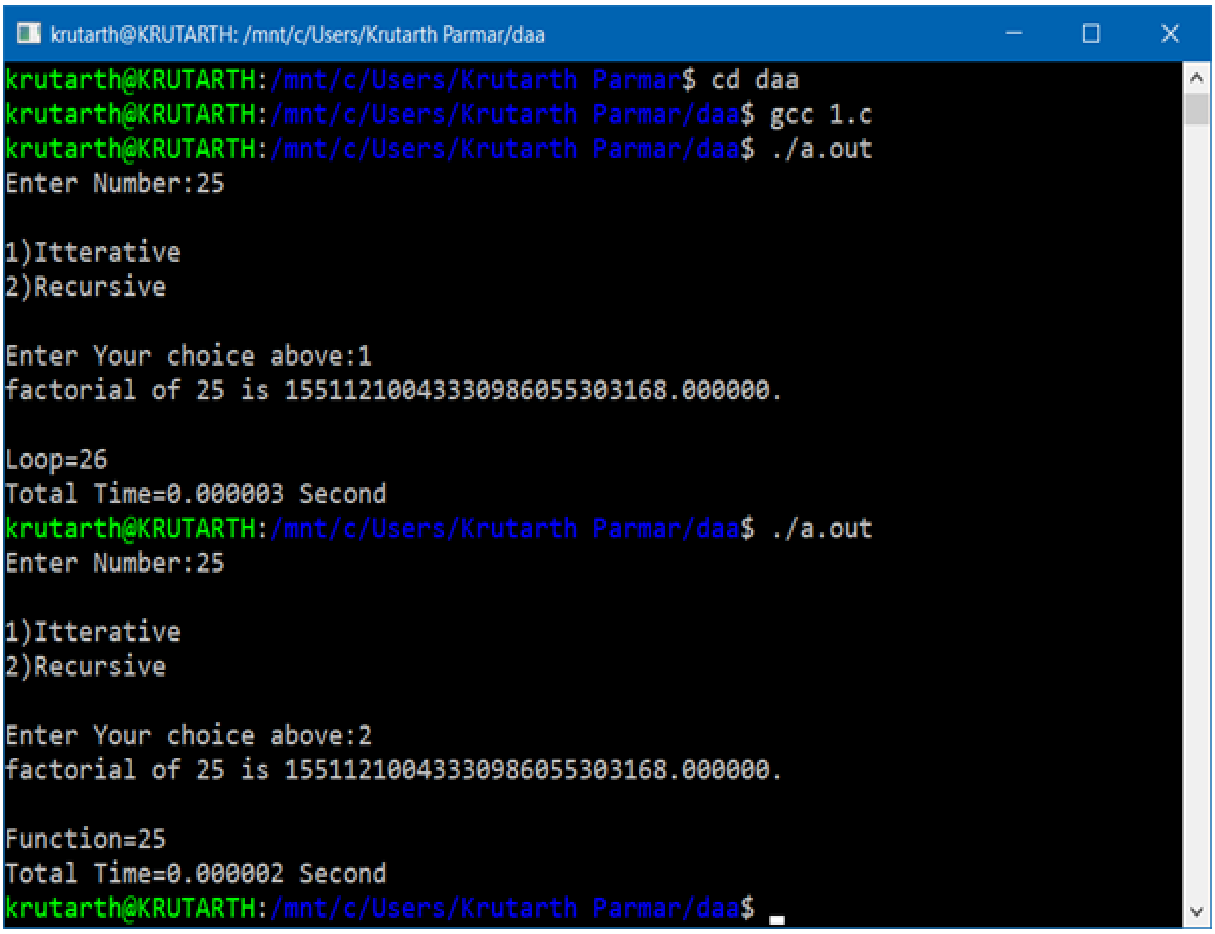
Aim: Implement factorial algorithm using iterative and recursive manner.

Content:

```
#include<stdio.h>
#include<time.h>
int lc=0,fc=0;
double fact(int n)                    //Recursive
{
        double f;
        if(n==1)
        f=1;
        else
        f=n*fact(n-1);
fc++;                                 //Function Count
return f;
}
void main()
{
        clock_t st,et;
        int i,n,c;
        double f=1,tt;
        printf("Enter Number:");
        scanf("%d",&n);
        printf("\n1)Itterative");
        printf("\n2)Recursive");
        printf("\n\nEnter Your choice above:");
        scanf("%d",&c);
        switch(c)
        {
                case 1:
                        st=clock();                 //Starting Time
                        for(i=1;lc++,i<n+1;i++)     //Itterative
                        f=f*i;
                        et=clock();                 //Ending Time
                        break;

                case 2:
                        st=clock();                 //Starting Time
                        f=fact(n);
                        et=clock();                 //Ending Time
                        break;

                default:
                        printf("Wrong Choice");
                        break;
```

```
            }
            tt=(double) (et-st)/CLOCKS_PER_SEC;                //Total Time
            printf("factorial of %d is %lf.\n",n,f);
            if(c==1)
            printf("\nLoop=%d",lc);
            else
            printf("\nFunction=%d",fc);
            printf("\nTotal Time=%lf Second\n",tt);
}
```

Output:



```
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar$ cd daa
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ gcc 1.c
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out
Enter Number:25

1)Itterative
2)Recursive

Enter Your choice above:1
factorial of 25 is 15511210043330986055303168.000000.

Loop=26
Total Time=0.000003 Second
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out
Enter Number:25

1)Itterative
2)Recursive

Enter Your choice above:2
factorial of 25 is 15511210043330986055303168.000000.

Function=25
Total Time=0.000002 Second
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ _
```

Practical 2

Aim: Implement bubble sort algorithm and measure its execution time.

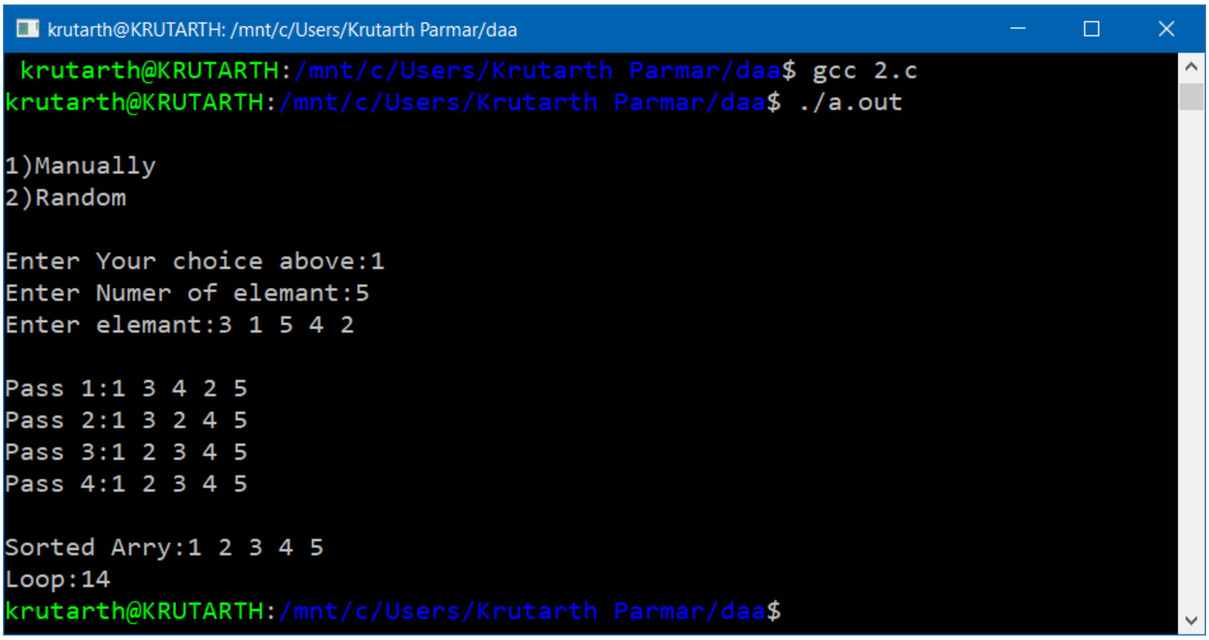
Content:

```
#include<stdio.h>
#include<time.h>
clock_t st,et;
int a[50000],n,i,j,k,c,cc,lc=0;
double tt;
void bubble()
{
	for(i=0;i<n-1;i++)
	for(j=0;lc++,j<n-i-1;j++)
	if(a[j]>a[j+1])
	{
		a[j]=a[j]+a[j+1];
		a[j+1]=a[j]-a[j+1];
		a[j]=a[j]-a[j+1];
	}
}
void main()
{
	printf("\n1)Manually");
	printf("\n2)Random");
	printf("\n\nEnter Your choice above:");
	scanf("%d",&c);
	if(c==1)
	{
		printf("Enter Numer of elemant:");
		scanf("%d",&n);
		printf("Enter elemant:");
		for(i=0;i<n;i++)
		scanf("%d",&a[i]);
		st=clock();
		for(i=0;i<n-1;i++)
		{
			for(j=0;lc++,j<n-i-1;j++)
			if(a[j]>a[j+1])
			{
				a[j]=a[j]+a[j+1];
				a[j+1]=a[j]-a[j+1];
				a[j]=a[j]-a[j+1];
			}
```

```
                printf("\nPass %d:",i+1);
                for(k=0;k<n;k++)
                printf("%d ",a[k]);
        }
        et=clock();
    }
    else if(c==2)
    {
        n=50000;
        printf("\n\n1)Best Case");
        printf("\n2)Average Case");
        printf("\n3)Worst Case");
        printf("\n\nEnter Your choice above:");
        scanf("%d",&cc);
        switch(cc)
        {
                case 1:
                        for(i=0;i<n;i++)
                        a[i]=i;
                        st=clock();
                        bubble();
                        et=clock();
                        break;

                case 2:
                        for(i=0;i<n/2;i++)
                        a[i]=i;
                        for(i=n/2,j=49999;i<n;i++,j--)
                        a[i]=j;
                        st=clock();
                        bubble();
                        et=clock();
                        break;

                case 3:
                        for(i=0,j=49999;i<n;i++,j--)
                        a[i]=j;
                        st=clock();
                        bubble();
                        et=clock();
                        break;

                default:
                        printf("Wrong Choice");
                        break;
        }
```

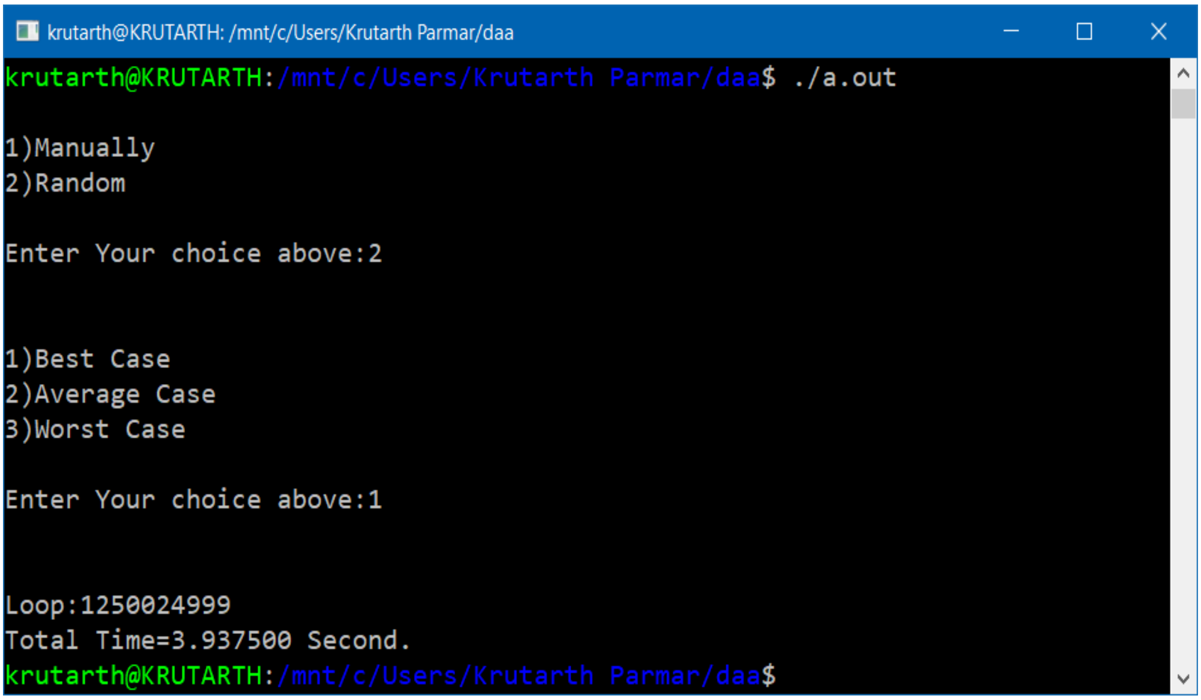
```
            }
            else
            printf("Wrong Choice");
            tt=(double) (et-st)/CLOCKS_PER_SEC;
            if(c==1)
            {
                printf("\n\nSorted Arry:");
                for(i=0;i<n;i++)
                printf("%d ",a[i]);
            }
            printf("\n\nLoop:%d",lc);
            printf("\nTotal Time=%lf Second.\n",tt);
}
```

Output:



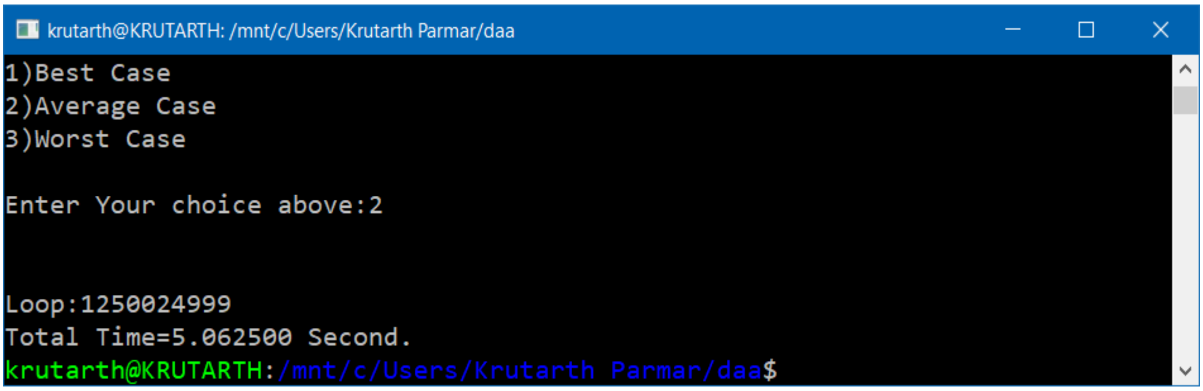
```
 krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ gcc 2.c
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out

1)Manually
2)Random

Enter Your choice above:1
Enter Numer of elemant:5
Enter elemant:3 1 5 4 2

Pass 1:1 3 4 2 5
Pass 2:1 3 2 4 5
Pass 3:1 2 3 4 5
Pass 4:1 2 3 4 5

Sorted Arry:1 2 3 4 5
Loop:14
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Best Case:



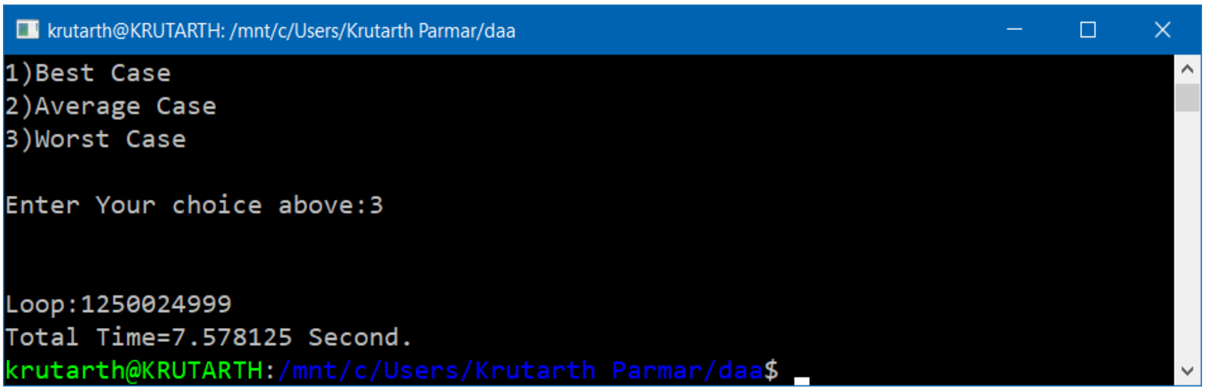
```
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out

1)Manually
2)Random

Enter Your choice above:2


1)Best Case
2)Average Case
3)Worst Case

Enter Your choice above:1


Loop:1250024999
Total Time=3.937500 Second.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Average Case:



```
1)Best Case
2)Average Case
3)Worst Case

Enter Your choice above:2


Loop:1250024999
Total Time=5.062500 Second.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Worst Case:



```
1)Best Case
2)Average Case
3)Worst Case

Enter Your choice above:3


Loop:1250024999
Total Time=7.578125 Second.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

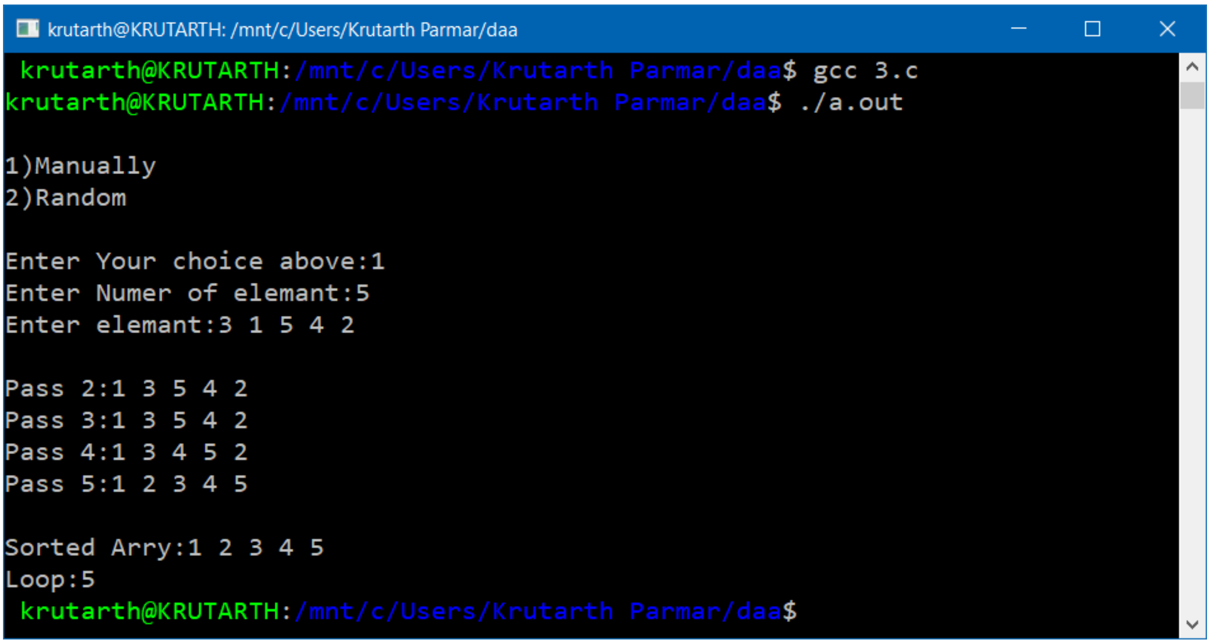
Practical 3

Aim: Implement insertion sort algorithm and perform its best case, average case and worst case analysis.

Content:

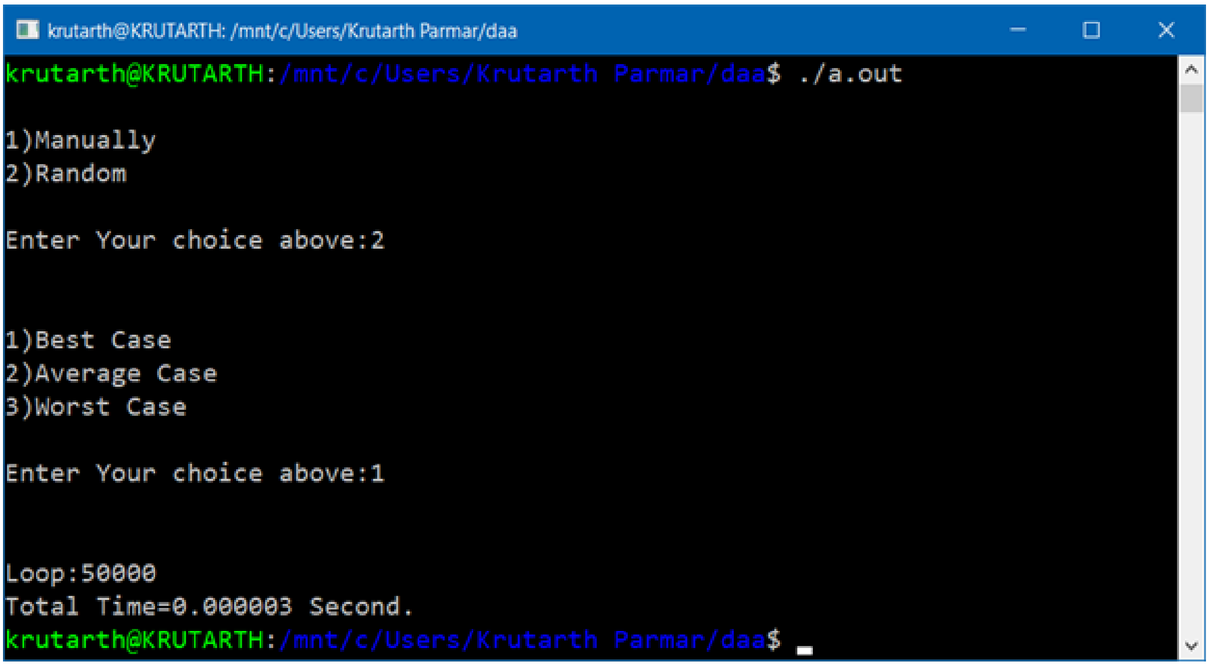
```
#include<stdio.h>
#include<time.h>
clock_t st,et;
int a[50000],n,i,j,k,key,c,cc,lc=0;
double tt;
void insertion()
{
        for(i=1;lc++,i<n;i++)
        {
                key=a[i];
                j=i-1;
                while(a[j]>key && j>=0)
                {
                        a[j+1]=a[j];
                        j--;
                }
                a[j+1]=key;
        }
}
void main()
{
        printf("\n1)Manually");
        printf("\n2)Random");
        printf("\n\nEnter Your choice above:");
        scanf("%d",&c);
        if(c==1)
        {
                printf("Enter Numer of elemant:");
                scanf("%d",&n);
                printf("Enter elemant:");
                for(i=0;i<n;i++)
                scanf("%d",&a[i]);
                st=clock();
                for(i=1;lc++,i<n;i++)
                {
                        key=a[i];
                        j=i-1;
                        while(a[j]>key && j>=0)
                        {
                                a[j+1]=a[j];
                                j--;
                        }
                        a[j+1]=key;
                        printf("\nPass %d:",i+1);
                        for(k=0;k<n;k++)
                        printf("%d ",a[k]);
                }
                et=clock();
        }
        else if(c==2)
        {
                n=50000;
                printf("\n\n1)Best Case");
                printf("\n2)Average Case");
                printf("\n3)Worst Case");
                printf("\n\nEnter Your choice above:");
                scanf("%d",&cc);
                switch(cc)
                {
                        case 1:
                                for(i=0;i<n;i++)
                                a[i]=i;
                                st=clock();
                                insertion();
                                et=clock();
                                break;

                        case 2:
                                for(i=0;i<n/2;i++)
                                a[i]=i;
                                for(i=n/2,j=49999;i<n;i++,j--)
                                a[i]=j;
                                st=clock();
                                insertion();
                                et=clock();
                                break;

                        case 3:
                                for(i=0,j=49999;i<n;i++,j--)
                                a[i]=j;
                                st=clock();
                                insertion();
                                et=clock();
                                break;
                        default:
                                printf("Wrong Choice");
                                break;
                }
        }
        else
        printf("Wrong Choice");
        tt=(double) (et-st)/CLOCKS_PER_SEC;
        if(c==1)
        {
           printf("\n\nSorted Arry:");
           for(i=0;i<n;i++)
           printf("%d ",a[i]);
        }
        printf("\n\nLoop:%d",lc);
        printf("\nTotal Time=%lf Second.\n",tt);
}
```

Output:



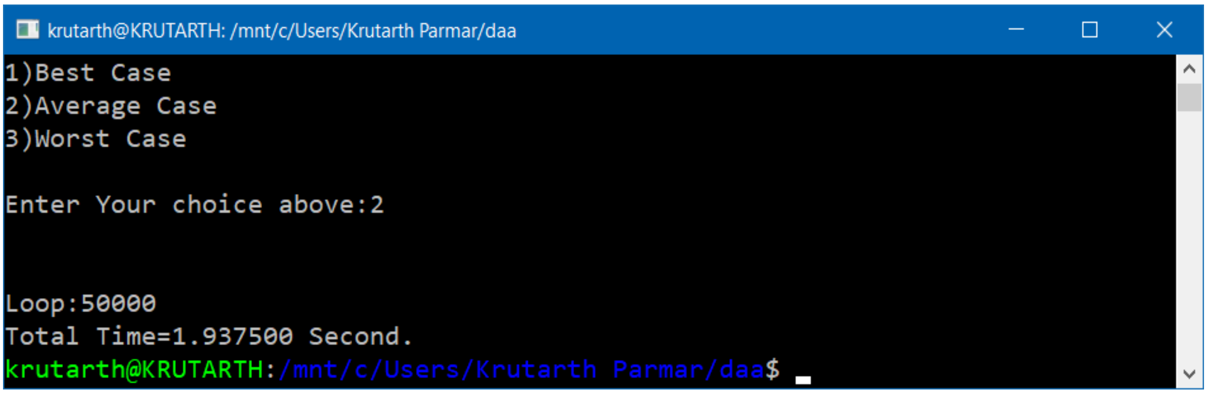
```
 krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ gcc 3.c
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out

1)Manually
2)Random

Enter Your choice above:1
Enter Numer of elemant:5
Enter elemant:3 1 5 4 2

Pass 2:1 3 5 4 2
Pass 3:1 3 5 4 2
Pass 4:1 3 4 5 2
Pass 5:1 2 3 4 5

Sorted Arry:1 2 3 4 5
Loop:5
 krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Best Case:



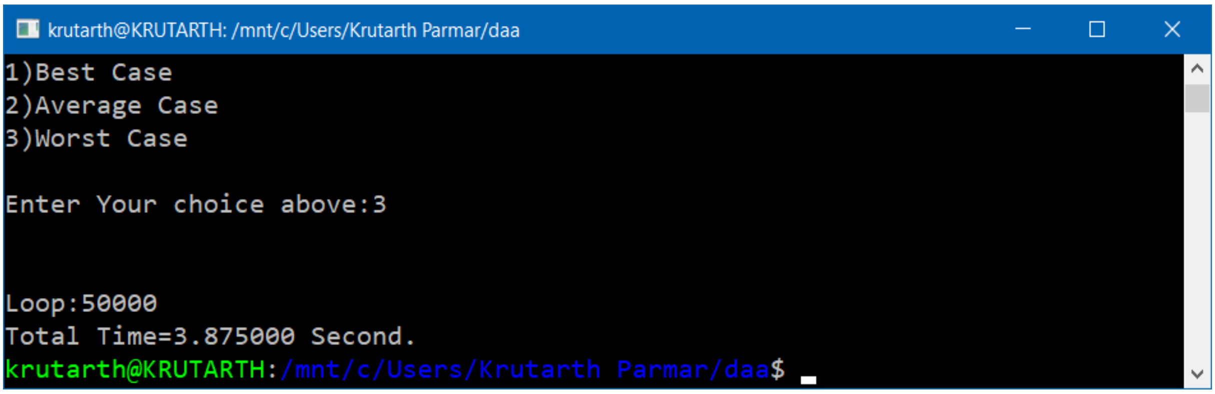
```
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out

1)Manually
2)Random

Enter Your choice above:2


1)Best Case
2)Average Case
3)Worst Case

Enter Your choice above:1


Loop:50000
Total Time=0.000003 Second.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Average Case:



```
1)Best Case
2)Average Case
3)Worst Case

Enter Your choice above:2


Loop:50000
Total Time=1.937500 Second.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Worst Case:



```
1)Best Case
2)Average Case
3)Worst Case

Enter Your choice above:3


Loop:50000
Total Time=3.875000 Second.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Practical 4

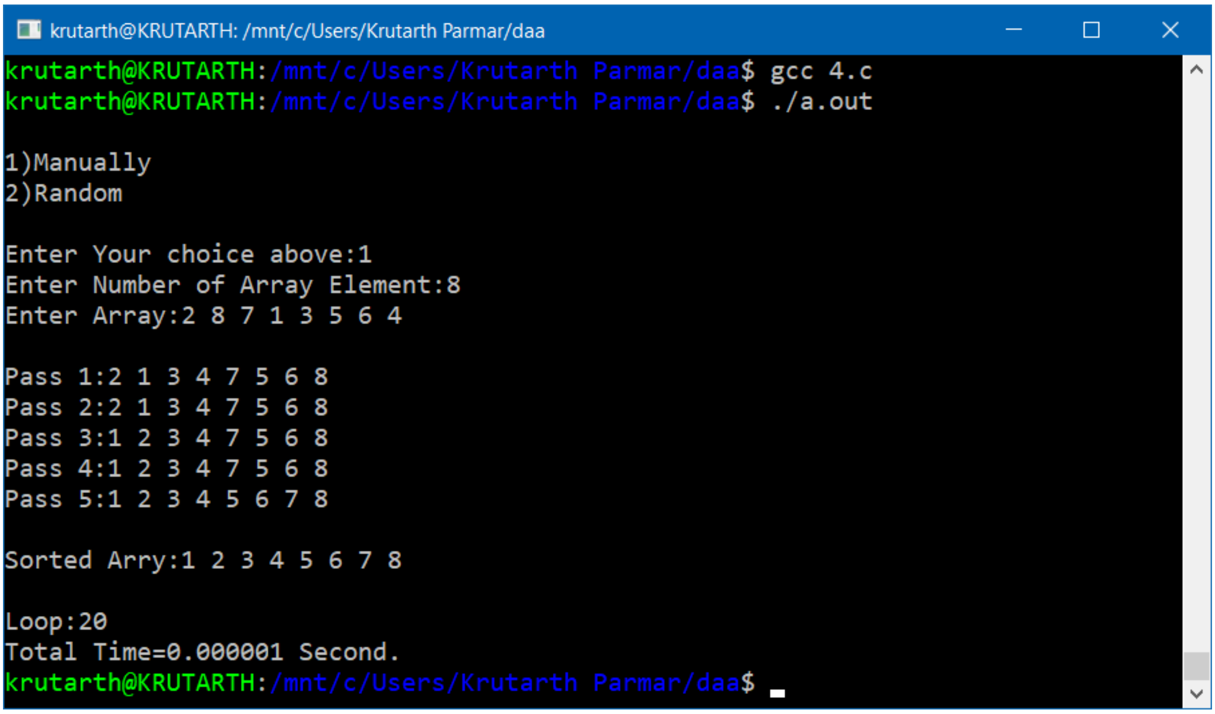
Aim: Implement quick sort algorithm and perform its best case, average case and worst case analysis.

Content:

```
#include<stdio.h>
#include<time.h>
int a[50000],c,n,m=0,lc=0;
int partition(int p,int r)
{
        int i,j,x,temp;
        i=p-1;
        x=a[r];
        for(j=p;lc++,j<r;j++)
        if(a[j]<=x)
        {
                i++;
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
        }
        temp=a[i+1];
        a[i+1]=a[r];
        a[r]=temp;
        if(c==1)
        {
                printf("\nPass %d:",++m);
                for(x=0;x<n;x++)
                printf("%d ",a[x]);
        }
return i+1;
}
void quicksort(int p,int r)
{
        if(p<r)
        {
                int q=partition(p,r);
                quicksort(p,q-1);
                quicksort(q+1,r);
        }
}
void main()
{
        clock_t st,et;
        int i,j,cc;
        double tt;
        printf("\n1)Manually\n2)Random");
        printf("\n\nEnter Your choice above:");
        scanf("%d",&c);
        if(c==1)
        {
                printf("Enter Number of Array Element:");
                scanf("%d",&n);
                printf("Enter Array:");
                for(i=0;i<n;i++)
                scanf("%d",&a[i]);
                st=clock();
                quicksort(0,n-1);
                et=clock();
        }
        else if(c==2)
        {
                n=50000;
                printf("\n\n1)Best Case");
                printf("\n2)Average Case");
                printf("\n3)Worst Case");
                printf("\n\nEnter Your choice above:");
                scanf("%d",&cc);
                switch(cc)
                {
                        case 1:
                                for(i=0,j=49999;i<n;i++,j--)
                                a[i]=j;
                                st=clock();
                                quicksort(0,49999);
                                et=clock();
                                break;

                        case 2:
                           for(i=0;i<n/2;i++)
                                a[i]=i;
                                for(i=n/2,j=49999;i<n;i++,j--)
                                a[i]=j;
                                st=clock();
                                quicksort(0,49999);
                                et=clock();
                                break;

                        case 3:
```

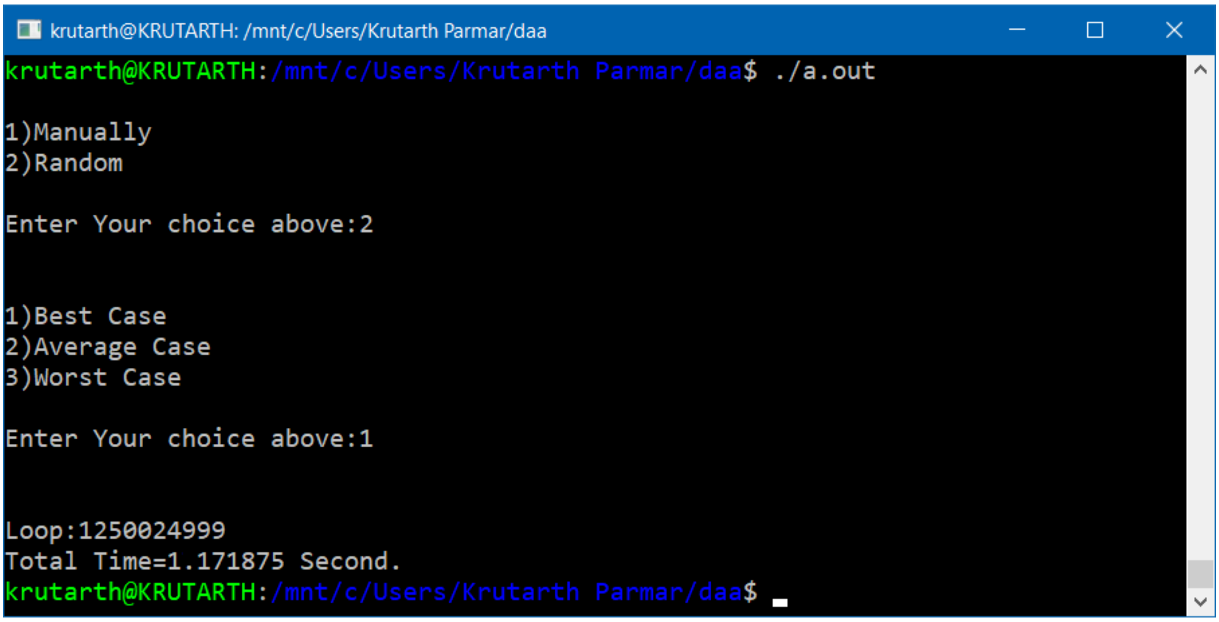
```
                for(i=0;i<n;i++)
                a[i]=i;
                st=clock();
                quicksort(0,49999);
                et=clock();
                break;

            default:
                printf("\nWrong Choice\n");
                break;
        }
    }
    else
    printf("\nWrong Choice\n");
    tt=(double) (et-st)/CLOCKS_PER_SEC;
    if(c==1)
    {
      printf("\n\nSorted Arry:");
      for(i=0;i<n;i++)
      printf("%d ",a[i]);
    }
    printf("\n\nLoop:%d",lc);
    printf("\nTotal Time=%lf Second.\n",tt);
}
```

Output:



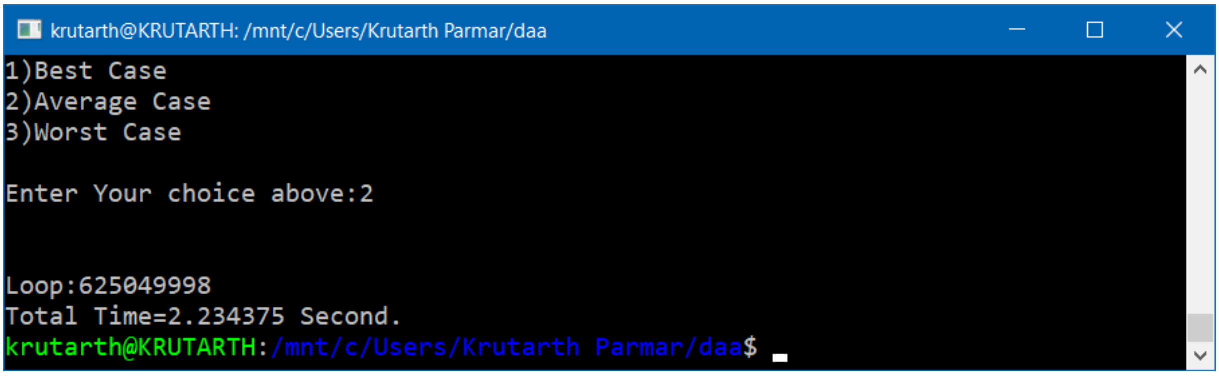
```
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ gcc 4.c
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out

1)Manually
2)Random

Enter Your choice above:1
Enter Number of Array Element:8
Enter Array:2 8 7 1 3 5 6 4

Pass 1:2 1 3 4 7 5 6 8
Pass 2:2 1 3 4 7 5 6 8
Pass 3:1 2 3 4 7 5 6 8
Pass 4:1 2 3 4 7 5 6 8
Pass 5:1 2 3 4 5 6 7 8

Sorted Arry:1 2 3 4 5 6 7 8

Loop:20
Total Time=0.000001 Second.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Best Case:



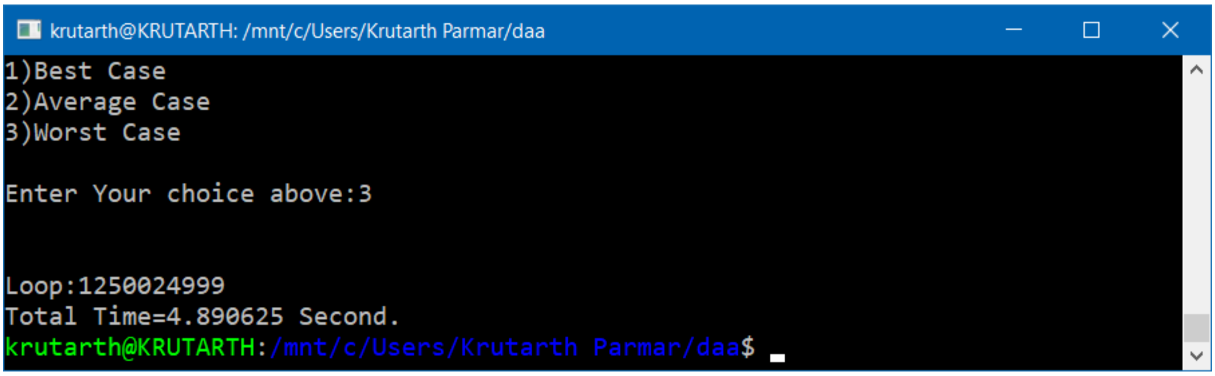
```
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out

1)Manually
2)Random

Enter Your choice above:2


1)Best Case
2)Average Case
3)Worst Case

Enter Your choice above:1


Loop:1250024999
Total Time=1.171875 Second.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Average Case:



```
1)Best Case
2)Average Case
3)Worst Case

Enter Your choice above:2


Loop:625049998
Total Time=2.234375 Second.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Worst Case:



```
1)Best Case
2)Average Case
3)Worst Case

Enter Your choice above:3


Loop:1250024999
Total Time=4.890625 Second.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Practical 5

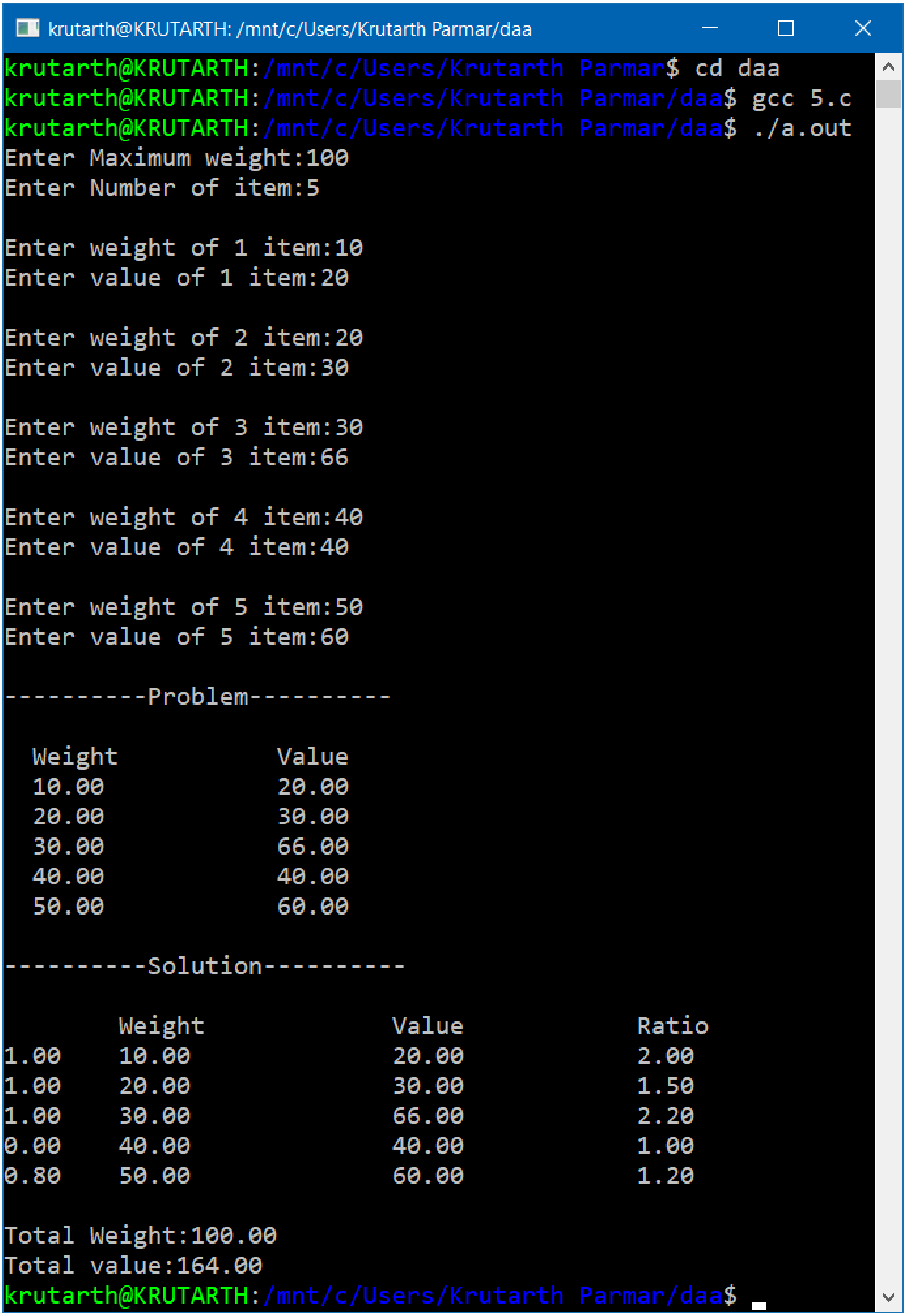
Aim: Implement knapsack problem using greedy approach.

Content:

```
#include<stdio.h>
int n,i,j;
double tw=0,tv=0,max;
struct d
{
        int f;
        double v,w,k,r;
}d[100];
void knpsacp()
{
        double a[100],temp,key;
        for(i=0;i<n;i++)
        a[i]=d[i].r;
        for(i=1;i<n;i++)
        {
                key=a[i];
                j=i-1;
                while(a[j]<key && j>=0)
                {
                        a[j+1]=a[j];
                        j--;
                }
                a[j+1]=key;
        }
        for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        if(a[i]==d[j].r)
        if(d[j].f!=28)
        if(tw+d[j].w<=max)
        {
                tw=tw+d[j].w;
                tv=tv+d[j].v;
                d[j].k=1;
                d[j].f=28;
                j=n;
        }
        else
        {
                temp=(max-tw)/d[j].w;
```

```
                        tw=tw+d[j].w*temp;
                        tv=tv+d[j].v*temp;
                        d[j].k=temp;
                        d[j].f=28;
                        i=j=n;
            }
}
void main()
{
            printf("Enter Maximum weight:");
            scanf("%lf",&max);
            printf("Enter Number of item:");
            scanf("%d",&n);
            for(i=0;i<n;i++)
            {
                        printf("\n");
                        printf("Enter weight of %d item:",i+1);
                        scanf("%lf",&d[i].w);
                        printf("Enter value of %d item:",i+1);
                        scanf("%lf",&d[i].v);
                        d[i].r=d[i].v/d[i].w;
                        d[i].f=0;
            }
            printf("\n----------Problem----------\n");
            printf("\n  Weight  \t   Value");
            for(i=0;i<n;i++)
            printf("\n  %.2lf   \t   %.2lf",d[i].w,d[i].v);
            printf("\n");
            knpsacp();
            printf("\n----------Solution----------\n");
            printf("\n\tWeight  \t   Value   \t    Ratio");
            for(i=0;i<n;i++)
            printf("\n%.2lf    %.2lf    \t   %.2lf   \t    %.2lf",d[i].k,d[i].w,d[i].v,d[i].r);
            printf("\n\nTotal Weight:%.2lf",tw);
            printf("\nTotal value:%.2lf\n",tv);
}
```

Output:



```
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar$ cd daa
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ gcc 5.c
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out
Enter Maximum weight:100
Enter Number of item:5

Enter weight of 1 item:10
Enter value of 1 item:20

Enter weight of 2 item:20
Enter value of 2 item:30

Enter weight of 3 item:30
Enter value of 3 item:66

Enter weight of 4 item:40
Enter value of 4 item:40

Enter weight of 5 item:50
Enter value of 5 item:60

----------Problem----------

  Weight           Value
  10.00            20.00
  20.00            30.00
  30.00            66.00
  40.00            40.00
  50.00            60.00

----------Solution----------

        Weight             Value            Ratio
1.00    10.00              20.00            2.00
1.00    20.00              30.00            1.50
1.00    30.00              66.00            2.20
0.00    40.00              40.00            1.00
0.80    50.00              60.00            1.20

Total Weight:100.00
Total value:164.00
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$
```

Practical 6

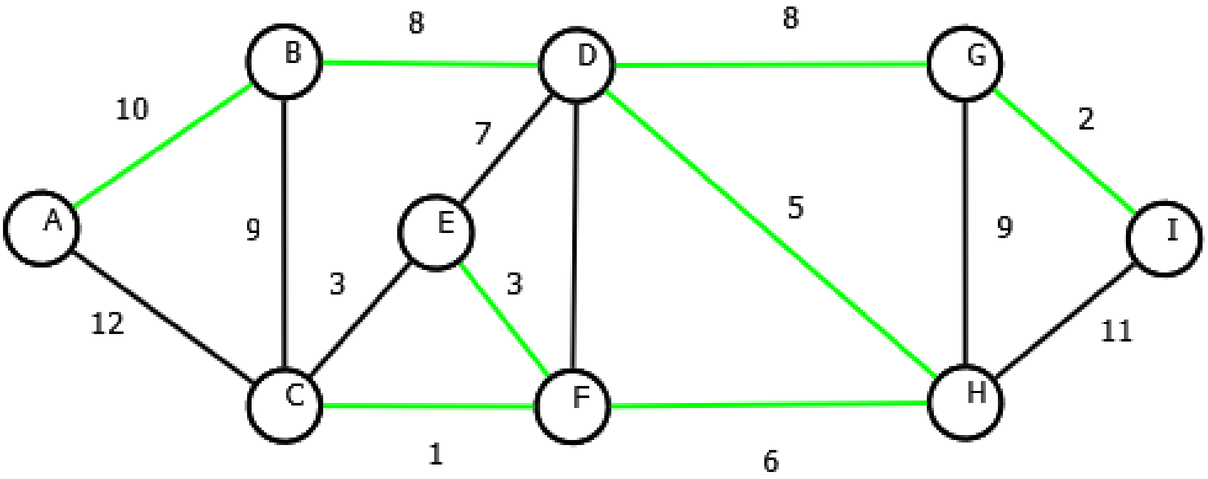
Aim: Implement Prim’s algorithm for finding shortest path.

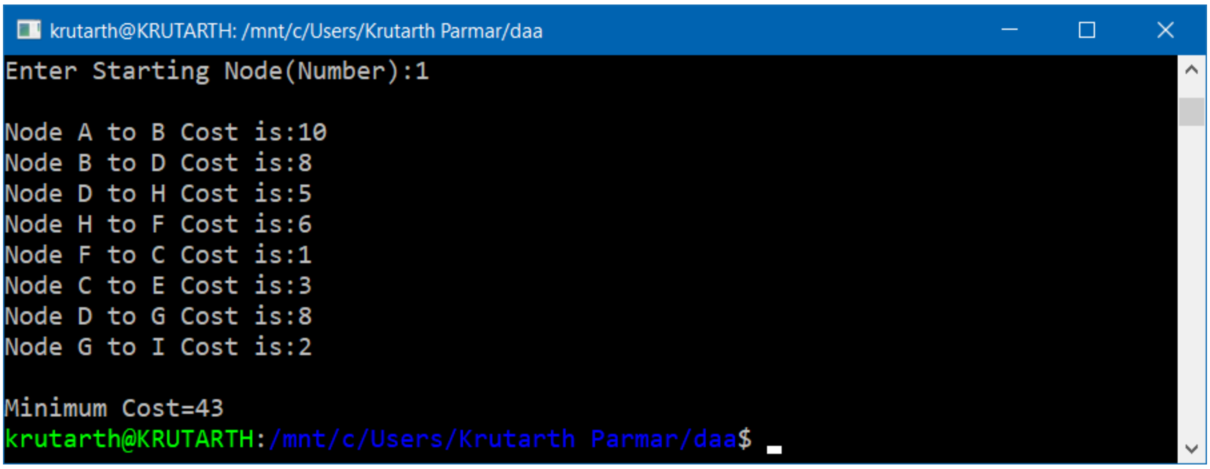
Content:

```
#include<stdio.h>
void main()
{
	int i,j,a,b,f,n,min,temp,vnode=1,tcost=0;
	int c[100][100],root[100];
	printf("1)Number\n2)Alphabet");
	printf("\n\nEnter Your choice:");
	scanf("%d",&temp);
	if(temp==1)
	f=1;
	if(temp==2)
	f=65;
	printf("\nEnter Number of Node:");
	scanf("%d",&n);
	printf("\n");
	for(i=0;i<n;i++)
	for(j=i+1;j<n;j++)
	{
		if(f==1)
		printf("Enter cost of edje[%d][%d]:",i+f,j+f);
		if(f==65)
		printf("Enter cost of edje[%c][%c]:",i+f,j+f);
		scanf("%d",&c[i][j]);
		c[j][i]=c[i][j];
		if(c[i][j]==0)
		c[i][j]=c[j][i]=999;
	}
	printf("\nEnter Starting Node(Number):");
	scanf("%d",&temp);
	root[temp-1]=1;
	while(vnode<n)
	{
		min=999;
		for(i=0;i<n;i++)
		for(j=0;j<n;j++)
		if(c[i][j]<min)
		if(root[i]!=0)
		{
			min=c[i][j];
```

```
                        a=i;
                        b=j;
                }
                if(root[a]==0 || root[b]==0)
                {
                        if(f==1)
                        printf("\nNode %d to %d Cost is:%d",a+f,b+f,min);
                        if(f==65)
                        printf("\nNode %c to %c Cost is:%d",a+f,b+f,min);
                        tcost+=min;
                        root[b]=1;
                        vnode++;
                }
                c[a][b]=c[b][a]=999;
        }
        printf("\n\nMinimum Cost=%d\n",tcost);
}
```

Output:





```
Enter Starting Node(Number):1

Node A to B Cost is:10
Node B to D Cost is:8
Node D to H Cost is:5
Node H to F Cost is:6
Node F to C Cost is:1
Node C to E Cost is:3
Node D to G Cost is:8
Node G to I Cost is:2

Minimum Cost=43
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ 
```

Practical 7

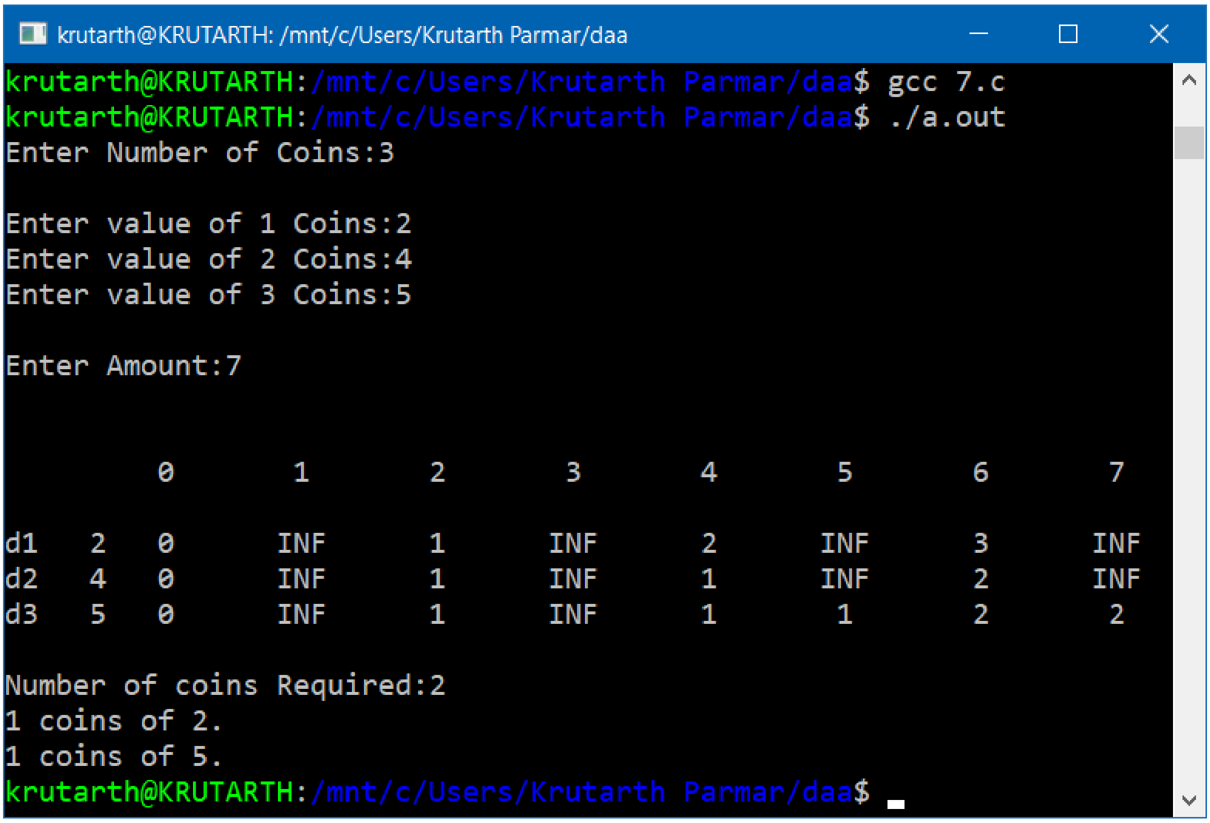
Aim: Implement making change problem using dynamic programming.

Content:

```
#include<stdio.h>
void main()
{
        int i,j,n,N,x,y;
        int c[100][100],d[100],m[100];
        printf("Enter Number of Coins:");
        scanf("%d",&n);
        printf("\n");
        for(i=1;i<=n;i++)
        {
                printf("Enter value of %d Coins:",i);
                scanf("%d",&d[i]);
        }
        printf("\nEnter Amount:");
        scanf("%d",&N);
        printf("\n\n     ");
        for(j=0;j<=N;j++)
        printf("\t %d",j);
        printf("\n");
        for(i=1;i<=n;i++)
        {
                printf("\nd%d   %d",i,d[i]);
                for(j=0;j<=N;j++)
                if(j==0)
                {
                        printf("\t 0");
                        c[i][j]=0;
                }
                else if(i==1 && j<d[i])
                {
                        printf("\tINF");
                        c[i][j]=999;
                }
                else if(i==1)
                {
                        c[i][j]=1+c[1][j-d[i]];
                        if(c[i][j]<999)
                        printf("\t %d",c[i][j]);
                else
```

```
                    printf("\tINF");
            }
            else if(j<d[i])
            {
                    c[i][j]=c[i-1][j];
                    if(c[i][j]<999)
                    printf("\t %d",c[i][j]);
                    else
                    printf("\tINF");
            }
            else
            {
                    if(c[i-1][j]<(1+c[i][j-d[i]]))
                    c[i][j]=c[i-1][j];
                    else
                    c[i][j]=1+c[i][j-d[i]];
                    if(c[i][j]<999)
                    printf("\t %d",c[i][j]);
                    else
                    printf("\tINF");
            }
    }
    printf("\n\nNumber of coins Required:%d",c[n][N]);
    x=n;y=N;
    while(c[x][y]!=0)
    {
            while(c[x][y]==c[x-1][y])
            x--;
            y=y-d[x];
            m[x]+=1;
    }
    for(i=1;i<=n;i++)
    if(m[i]!=0)
    printf("\n%d coins of %d.",m[i],d[i]);
    printf("\n");
}
```

Output:



krutarth@KRUTARTH: /mnt/c/Users/Krutarth Parmar/daa

```
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ gcc 7.c
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out
Enter Number of Coins:3

Enter value of 1 Coins:2
Enter value of 2 Coins:4
Enter value of 3 Coins:5

Enter Amount:7


         0       1       2       3       4       5       6       7

d1    2  0       INF     1       INF     2       INF     3       INF
d2    4  0       INF     1       INF     1       INF     2       INF
d3    5  0       INF     1       INF     1       1       2       2

Number of coins Required:2
1 coins of 2.
1 coins of 5.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ _
```

Practical 8

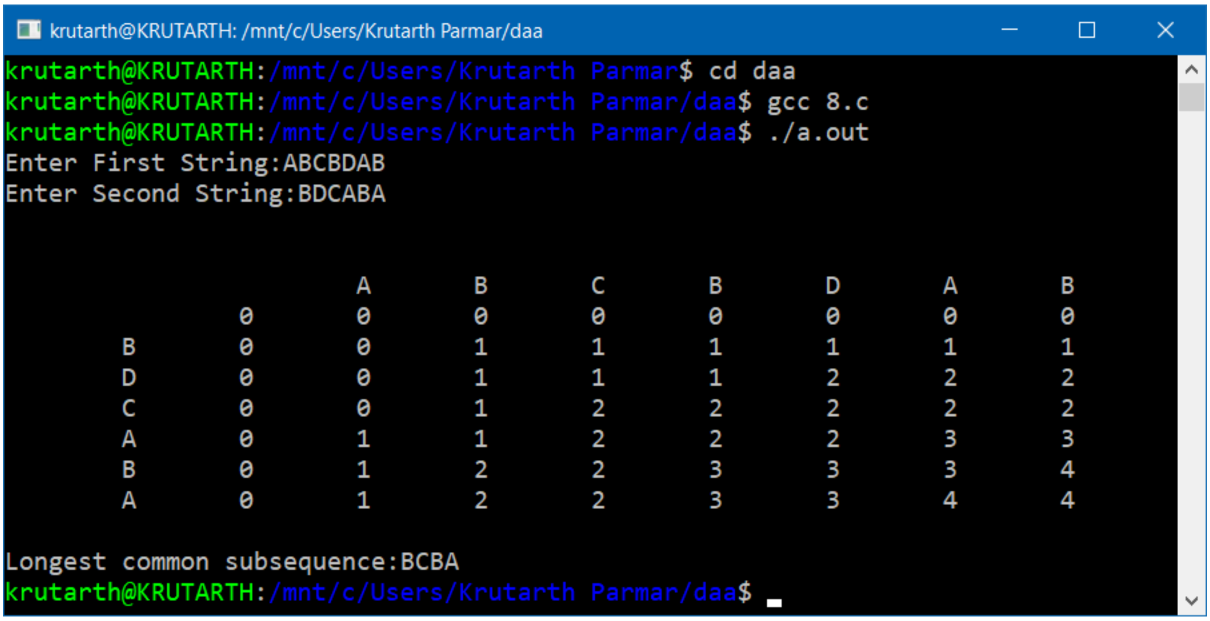
Aim: Implement Longest common subsequence using dynamic programming.

Content:

```
#include<stdio.h>
#include<string.h>
void main()
{
	int i,j,k=0;
	int c[100][100];
	char a[100],b[100],lcs[100];
	printf("Enter First String:");
	scanf("%s",b);
	printf("Enter Second String:");
	scanf("%s",a);
	printf("\n\n\t \t");
	for(i=0;i<strlen(b);i++)
	printf("\t%c",b[i]);
	for(i=0;i<=strlen(a);i++)
	{
		if(i==0)
		printf("\n\t ");
		else
		printf("\n\t%c",a[i-1]);
		for(j=0;j<=strlen(b);j++)
		{
			if(i==0 || j==0)
			{
				c[i][j]=0;
				printf("\t%d",c[i][j]);
			}
			else if(a[i-1]==b[j-1])
			{
				c[i][j]=1+c[i-1][j-1];
				printf("\t%d",c[i][j]);
			}
			else
			{
				if(c[i][j-1]>c[i-1][j])
				c[i][j]=c[i][j-1];
				else
				c[i][j]=c[i-1][j];
				printf("\t%d",c[i][j]);
			}
```

```
                        }
                }
                printf("\n\nLongest common subsequence:");
                i=strlen(a);
                j=strlen(b);
                while(i>0 && j>0)
                if(a[i-1]==b[j-1])
                {
                        lcs[k]=a[i-1];
                        i--;j--;k++;
                }
                else if(c[i-1][j]>c[i][j-1])
                {       i--;    }
                else
                {       j--;    }
                for(i=strlen(lcs)-1;i>=0;i--)
                printf("%c",lcs[i]);
                printf("\n");
}
```

Output:



```
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar$ cd daa
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ gcc 8.c
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ ./a.out
Enter First String:ABCBDAB
Enter Second String:BDCABA


                        A       B       C       B       D       A       B
                0       0       0       0       0       0       0       0
        B       0       0       1       1       1       1       1       1
        D       0       0       1       1       1       2       2       2
        C       0       0       1       2       2       2       2       2
        A       0       1       1       2       2       2       3       3
        B       0       1       2       2       3       3       3       4
        A       0       1       2       2       3       3       4       4

Longest common subsequence:BCBA
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/daa$ _
```

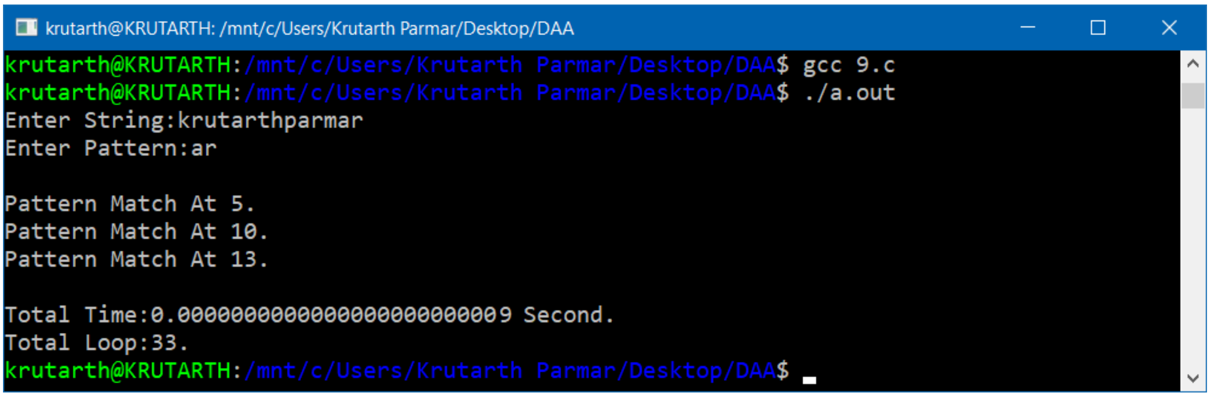
Practical 9

Aim: Implement topological sorting algorithm and measure its execution time.

Content:

```
#include<stdio.h>
#include<string.h>
#include<time.h>
void main()
{
        clock_t st,et;
        int i,j,k,f,n,m,x=1,lc=0;
        char s[100],p[100];
        double tt;
        printf("Enter String:");
        scanf("%s",s);
        printf("Enter Pattern:");
        scanf("%s",p);
        st=clock();
        n=strlen(s);
        m=strlen(p);
        for(i=0;lc++,i<=n-m;i++)
        {
                f=0;
                k=i;
                for(j=0;lc++,j<m;j++,k++)
                if(s[k]==p[j])
                f++;
                else
                break;
                if(f==m)
                {
                        printf("\nPattern Match At %d.",i+1);
                        x=0;
                }
        }
        if(x)
        printf("\nNo Match Found.");
        et=clock();
        tt=(double) (et-st)/CLOCKS_PER_SEC;
        printf("\n\nTotal Time:%.25lf Second.",tt);
        printf("\nTotal Loop:%d.\n",lc);
}
```

Output:



```
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/Desktop/DAA$ gcc 9.c
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/Desktop/DAA$ ./a.out
Enter String:krutarthparmar
Enter Pattern:ar

Pattern Match At 5.
Pattern Match At 10.
Pattern Match At 13.

Total Time:0.0000000000000000000000009 Second.
Total Loop:33.
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/Desktop/DAA$ _
```

Practical 10

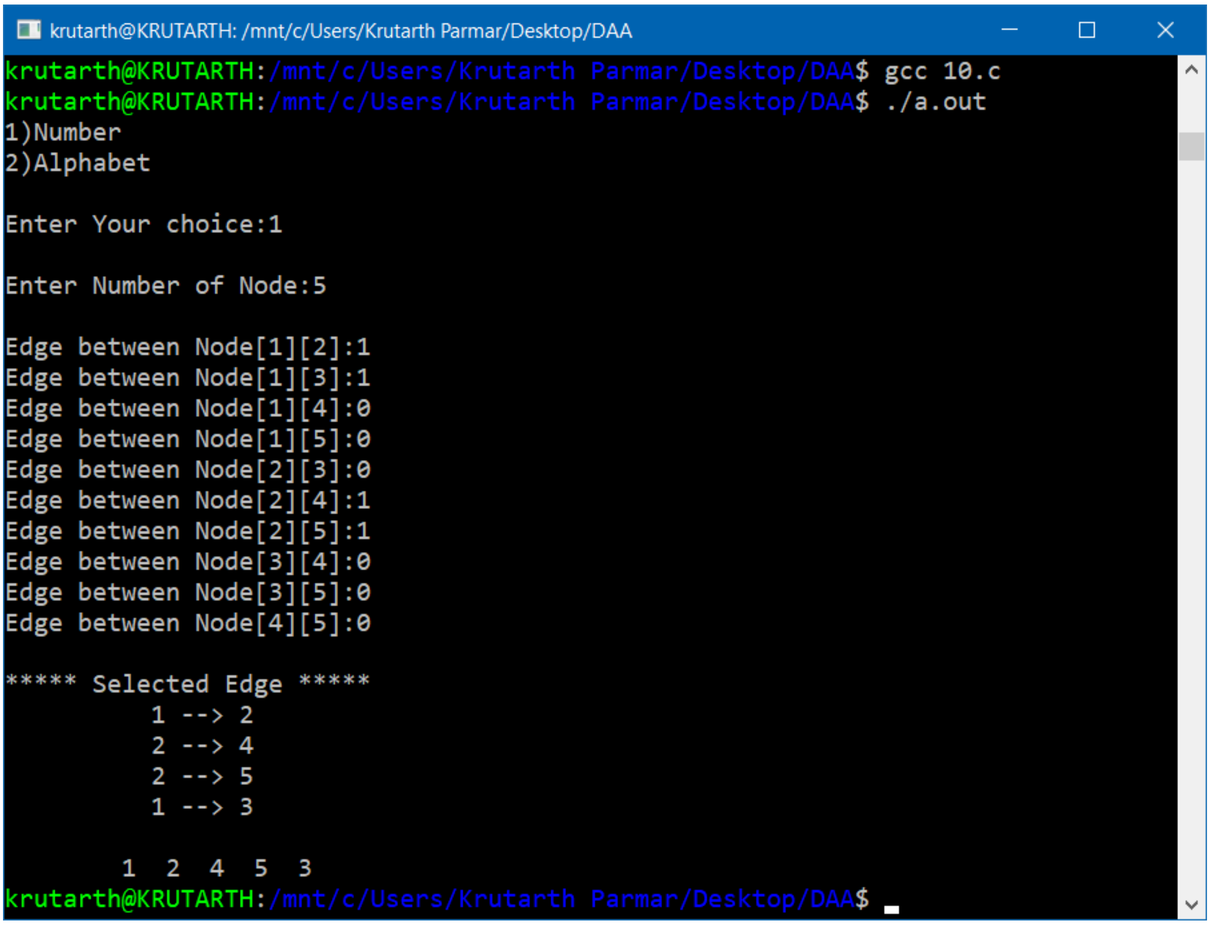
Aim: Implement a program that can traverse a path using depth first search algorithm.

Content:

```
#include<stdio.h>
int a[100][100],r[100],s[100],n,f,k=0;
int check(int x)
{
        int i;
        for(i=0;i<k;i++)
        if(s[i]==x)
        {
                x=0;
                i=k+1;
        }
return x;
}
void dfs(int node)
{
        int i;
        r[node]=1;
        for(i=0;i<n;i++)
        if(a[node][i] && !(r[i]))
        {
                if(f==1)
                printf("\n\t  %d --> %d",node+f,i+f);
                else
                printf("\n\t  %c --> %c",node+f,i+f);
                s[k]=check(node+f);
                k++;
                s[k]=check(i+f);
                k++;
                dfs(i);
        }
}
void main()
{
        int i,j,c=0,temp;
        printf("1)Number\n2)Alphabet");
        printf("\n\nEnter Your choice:");
        scanf("%d",&temp);
        if(temp==1)
        f=1;
```

```
      if(temp==2)
      f=65;
      printf("\nEnter Number of Node:");
      scanf("%d",&n);
      printf("\n");
      for(i=0;i<n;i++)
      for(j=i+1;j<n;j++)
      {
            if(f==1)
    printf("Edge between Node[%d][%d]:",i+f,j+f);
    else
    printf("Edge between Node[%c][%c]:",i+f,j+f);
            scanf("%d",&a[i][j]);
      }
      printf("\n***** Selected Edge *****");
      dfs(0);
      for(i=0;i<n;i++)
      if(r[i])
      c++;
      printf("\n\n\t");
      for(i=0;i<k;i++)
      if(s[i]!=0)
      if(f==1)
      printf("%d  ",s[i]);
      else
      printf("%c  ",s[i]);
      if(c!=n)
      printf("\nGrap is Not Connected.");
      printf("\n");
}
```

Output:



```
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/Desktop/DAA$ gcc 10.c
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/Desktop/DAA$ ./a.out
1)Number
2)Alphabet

Enter Your choice:1

Enter Number of Node:5

Edge between Node[1][2]:1
Edge between Node[1][3]:1
Edge between Node[1][4]:0
Edge between Node[1][5]:0
Edge between Node[2][3]:0
Edge between Node[2][4]:1
Edge between Node[2][5]:1
Edge between Node[3][4]:0
Edge between Node[3][5]:0
Edge between Node[4][5]:0

***** Selected Edge *****
          1 --> 2
          2 --> 4
          2 --> 5
          1 --> 3

        1  2  4  5  3
krutarth@KRUTARTH:/mnt/c/Users/Krutarth Parmar/Desktop/DAA$ _
```