EXERCISE-2 ELECTION ALGORITHM

BULLY ALGORITHM

CODE

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define MAX 10
int list[MAX],n,c;
void display()
{
        int i;
        printf("\nProcesses-->");
        for(i=1;i<=n;i++)
                printf("\t %d",i);
        printf("\nAlive-->");
        for(i=1;i<=n;i++)
                printf("\t %d",list[i]);
        printf("\ncoordinator is::%d",c);
}
void bully()
{
        int ch,crash,activate,i,gid,flag,subcdr;
        do
        {
                printf("\n1.Crash\n2.Activate\n3.Display\n4.Exit\nEnter You
choice::");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:
                                printf("\nEnter Process no. to Crash::");
                                scanf("%d",&crash);
                                if(list[crash])
                                        list[crash]=0;
                                else
                                {
                                        printf("\nProcess is alreaady dead!!");
                                        break;
                                }
                                do
                                {
                                        printf("\nEnter election generator id::");
                                        scanf("%d",&gid);
                                        if(gid==c)
                                        {
                                                printf("\nenter a valid generator
id::");
                                        }
                                }while(gid==crash);
                                flag=0;
```

```
                        if(crash==c)
                        {
                                for(i=gid+1;i<=n;i++)
                                {
                                        printf("\nmessage is sent from %d to
%d",gid,i);
                                        if(list[i])
                                        {
                                                subcdr=i;
                                                printf("Response is sent
from %d to %d",i,gid);
                                                flag=1;
                                        }
                                }
                                if(flag==1)
                                {
                                        c=subcdr;
                                }
                                else
                                {
                                        c=gid;
                                }
                        }
                        display();
                        break;
                case 2:
                        //activate
                        printf("\nEnter Process no. to Activated::");
                        scanf("%d",&activate);
                        if(!list[activate])
                                list[activate]=1;
                        else
                        {
                                printf("\nProcess is alreaady alive!!");
                                break;
                        }
                        if(activate==n)
                        {
                                c=n;
                                break;
                        }
                        for(i=activate+1;i<=n;i++)
                        {
                                printf("\nmessage is sent from %d to
%d",activate,i);
                                if(list[i])
                                {
                                        subcdr=i;
                                        printf("Response is sent from %d to
%d",i,activate);
                                        flag=1;
                                }
                        }
                        if(flag==1)
                        {
```

```
                                        c=subcdr;
                                    }
                                    else
                                    {
                                        c=activate;
                                    }
                                display();
                                break;
                        case 3:
                                display();
                        break;
                        case 4:
                                break;
                        }
                }while(ch!=4);
        }
        int main()
        {
                int i,j;
                printf("\nEnter no. of process::");
                scanf("%d",&n);
                for(i=1;i<=n;i++)
                {
                        printf("\nEnter Process %d is Alive or not(0/1)::",i);
                        scanf("%d",&list[i]);
                        if(list[i])
                                c=i;
                }
                display();
                printf("\nBULLY ALGORITHM\n");
                bully();
                return 0;
        }
```

OUTPUT

```
Enter no. of process::5

Enter Process 1 is Alive or not(0/1)::0

Enter Process 2 is Alive or not(0/1)::0

Enter Process 3 is Alive or not(0/1)::1

Enter Process 4 is Alive or not(0/1)::0

Enter Process 5 is Alive or not(0/1)::0

Processes-->     1       2       3       4       5
Alive-->         0       0       1       0       0
coordinator is::3
BULLY ALGORITHM

1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::1

Enter Process no. to Crash::2
```

```
Process is alreaady dead!!
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::3

Processes-->      1        2        3        4        5
Alive-->          0        0        1        0        0
coordinator is::3
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::2

Enter Process no. to Activated::2

message is sent from 2 to 3Response is sent from 3 to 2
message is sent from 2 to 4
message is sent from 2 to 5
Processes-->      1        2        3        4        5
Alive-->          0        1        1        0        0
coordinator is::3
1.Crash
2.Activate
3.Display
4.Exit
```

RING ALGORITHM

CODE

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define MAX 10
int list[MAX],n,c;
void display()
{
        int i;
        printf("\nProcesses-->");
        for(i=1;i<=n;i++)
                printf("\t %d",i);
        printf("\nAlive-->");
        for(i=1;i<=n;i++)
                printf("\t %d",list[i]);
        printf("\ncoordinator is::%d",c);
}
void ring()
{
        int msg[20],ring_n,k,i;
        int ch,crash,activate,gid,flag,subcdr;
        do
        {
                        printf("\n1.Crash\n2.Activate\n3.Display\n4.Exit\nEnter        You
choice::");
                        scanf("%d",&ch);
                        switch(ch)
                        {
                                case 1:
                                        printf("\nEnter Process no. to Crash::");
```

```
                scanf("%d",&crash);
                if(list[crash])
                        list[crash]=0;
                else
                {
                        printf("\nProcess is alreaady dead!!");
                        break;
                }
                do
                {
                        printf("\nEnter election generator id::");
                        scanf("%d",&gid);
                        if(gid==c)
                        {
                                printf("\nenter a valid generator
id::");
                        }
                }while(gid==crash);
                flag=0;
                k=1;
                if(crash==c)
                {
                        msg[k++]=gid;
                        for(i=(gid+1)%n;i!=gid;i=(i+1)%n)
                        {
                                if(list[i])
                                {
                                        printf("\nmessage is sent to
%d k =%d",i,k);

                                        msg[k++]=i;
                                        printf("Response is sent
//
from %d to %d",i,gid);


                                }
                        }
                        subcdr=0;
                        for(i=1;i<k;i++)
                        {
                                printf("\nmsg::%d\n",msg[i]);
                                if(subcdr<msg[i])
                                {
                                        subcdr=msg[i];
                                }
                        }
                        c=subcdr;
                }
                display();
                break;
        case 2:
                //activate
                printf("\nEnter Process no. to Activated::");
                scanf("%d",&activate);
                if(!list[activate])
                        list[activate]=1;
                        else
```

```
                                        {
                                                printf("\nProcess is alreaady alive!!");
                                                break;
                                        }
                                        //
                                        if(activate==n)
                                        {
                                                c=n;
                                                break;
                                        }
                                        for(i=activate+1;i<=n;i++)
                                        {
                                                printf("\nmessage is sent from %d to
%d",activate,i);
                                                if(list[i])
                                                {
                                                        subcdr=i;
                                                        printf("\nResponse is sent from %d to
%d",i,activate);
                                                        flag=1;
                                                }
                                        }
                                        if(flag==1)
                                        {
                                                c=subcdr;
                                        }
                                        else
                                        {
                                                c=activate;
                                        }


                                                display();
                                                break;
                                        case 3:
                                                display();
                                        break;
                                        case 4:
                                                break;
                                        }
        }while(ch!=4);

}
int main()
{
        int i,j;
        printf("\nEnter no. of process::");
        scanf("%d",&n);
        for(i=1;i<=n;i++)
        {
                printf("\nEnter Process %d is Alive or not(0/1)::",i);
                scanf("%d",&list[i]);
                if(list[i])
                        c=i;
        }
```

```
        display();
        printf("\nRING ALGORITHM\n");
        ring();
        return 0;
    }
```

OUTPUT

```
Enter no. of process::3

Enter Process 1 is Alive or not(0/1)::0

Enter Process 2 is Alive or not(0/1)::1

Enter Process 3 is Alive or not(0/1)::0

Processes-->     1       2       3
Alive-->         0       1       0
coordinator is::2
RING ALGORITHM

1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::1

Enter Process no. to Crash::3

Process is alreaady dead!!
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::3
```

```
Processes-->      1         2         3
Alive-->          0         1         0
coordinator is::2
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::2

Enter Process no. to Activated::2

Process is alreaady alive!!
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::3
```