DINING PHILOSOPHERS PROBLEM

PROGRAM

```
int tph, philname[20], status[20], howhung, hu[20], cho; main()

{

int i; clrscr();

printf("\n\nDINING PHILOSOPHER PROBLEM");

printf("\nEnter the total no. of philosophers: ");

scanf("%d",&tph);

for(i=0;i<tph;i++)

{

philname[i]=(i+1); status[i]=1;

}

printf("How many are hungry : ");
```

```
scanf("%d", &howhung);

if(howhung==tph)

{

printf("\n All are hungry..\nDead lock stage will occur");

printf(\n" Exiting\n");

else{

for(i=0;i<howhung;i++){

printf("Enterphilosopher%dposition:",(i+1));

scanf("%d",&hu[i]);

status[hu[i]]=2;

}

do

{
```

```
printf("1.One can eat at a time\t2.Two can eat at a time

\t3.Exit\nEnter your choice:");

scanf("%d", &cho);

switch(cho)

{

case 1: one();

break;

case 2: two();

break;

case 3: exit(0);

default: printf("\nInvalid option..");

}
```

```
}while(1);

}

}

one()

{

int pos=0, x, i;

printf("\nAllow one philosopher to eat at any time\n");

for(i=0;i<howhung; i++, pos++)

{

printf("\nP %d is granted to eat", philname[hu[pos]]);

for(x=pos;x<howhung;x++)

printf("\nP %d is waiting", philname[hu[x]]);

}
```

```
}

two()

{

int i, j, s=0, t, r, x;

printf("\n Allow two philosophers to eat at same

time\n"); for(i=0;i<howhung;i++)

{

for(j=i+1;j<howhung;j++)

{

if(abs(hu[i]-hu[j])>=1&& abs(hu[i]-hu[j])!=4)

{

printf("\n\ncombination %d \n", (s+1));
```

```
t=hu[i];

r=hu[j]; s++;

printf("\nP %d and P %d are granted to eat", philname[hu[i]],

philname[hu[j]]);

for(x=0;x<howhung;x++)

{

if((hu[x]!=t)&&(hu[x]!=r))

printf("\nP %d is waiting", philname[hu[x]]);

}

}

}

}

}
```

INPUT

DINING PHILOSOPHER PROBLEM

Enter the total no. of philosophers: 5

How many are hungry : 3

Enter philosopher 1 position: 2

Enter philosopher 2 position: 4

Enter philosopher 3 position: 5

OUTPUT

1.One can eat at a time 2.Two can

eat at a time 3.Exit Enter your choice: 1

Allow one philosopher to eat at any time

P 3 is granted to eat

P 3 is waiting

P 5 is waiting

P 0 is waiting

P 5 is granted to eat

P 5 is waiting

P 0 is waiting

P 0 is granted to eat

P 0 is waiting