**EXP-8**

**Design and implement C program to find a subset of a given set S={s1,s2,….sn} of n positive integers whose whose sum is equal to a given positive integer d.**

#include<stdio.h>

int s[10] , x[10],d ;

void sumofsub ( int , int , int ) ;

void main ()

{

int n , sum = 0 ;

int i ;

printf ( " \n Enter the size of the set : " ) ;

scanf ( "%d" , &n ) ;

printf ( " \n Enter the set in increasing order:\n" ) ;

for ( i = 1 ; i <= n ; i++ )

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

printf ( " \n Enter the value of d : \n " ) ;

scanf ( "%d" , &d ) ;

for ( i = 1 ; i <= n ; i++ )

sum = sum + s[i] ;

if ( sum < d || s[1] > d )

printf ( " \n No subset possible : " ) ;

else

sumofsub ( 0 , 1 , sum ) ;

}

void sumofsub ( int m , int k , int r )

{

int i=1 ; x[k] = 1 ;

if ( ( m + s[k] ) == d )

{

printf("Subset:");

for ( i = 1 ; i <= k ; i++ )

if ( x[i] == 1 )

printf ( "\t%d" , s[i] ) ;

printf ( "\n" ) ;

}

else

if ( m + s[k] + s[k+1] <= d )

sumofsub ( m + s[k] , k + 1 , r - s[k] ) ;

if ( ( m + r - s[k] >= d ) && ( m + s[k+1] <=d))

{

x[k] = 0;

sumofsub ( m , k + 1 , r - s[k] ) ;

}

}

EXP-12

Design and implement C program for N Queen’s problem using Backtracking

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

int a[30],count=0;

int place(int pos)

{

int i;

for(i=1;i<pos;i++)

{

if((a[i]==a[pos])||((abs(a[i]-a[pos])==abs(i-pos))))

return 0;

}

return 1;

}

void print\_sol(int n)

{

int i,j; count++;

printf("\n\nSolution #%d:\n",count);

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

{

if(a[i]==j)

printf("Q\t");

else

printf(“\*\t”);

}

printf("\n");

}

}

void queen(int n)

{

int k=1; a[k]=0;

while(k!=0)

{

a[k]=a[k]+1; while((a[k]<=n)&&!place(k))

a[k]++;

if(a[k]<=n)

{

if(k==n)

print\_sol(n);

else

{

k++;

a[k]=0;

}

}

else

k--;

}

}

void main()

{

int i,n;

printf("Enter the number of Queens\n");

scanf("%d",&n);

queen(n);

printf("\nTotal solutions=%d",count);

}

}