**ASSIGNMENT - 10**

**EXPERIMENT – 18: Write a program to find sum of subset by using backtracking.**

#include<stdio.h>

#include<stdlib.h>

void sumOfSub(int,int,int);

static int m=0;

int\*w;

int\*x;

void main()

{ int i=0,sum=0,n=0;

printf("Enter size of array:");

scanf("%d",&n);

w=(int\*)malloc(sizeof(int)\*n+1);

x=(int\*)malloc(sizeof(int)\*n+1);

printf("\nEnter %d elements:-\n",n);

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

{

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

sum+=w[i];

x[i]=0;

}

printf("\nEnter the sum to be obtained:");

scanf("%d",&m);

if(sum < m)

{ printf("\nNot possible to obtain any subset !!! ");

exit(1); }

printf("\nPossible Subsets are( 0 indicates exclusion and 1 indicates inclusion):-");

sumOfSub(0,1,sum);

}

void sumOfSub(int s,int k,int r)

{ int i=0;

x[k]=1;

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

{ printf("\n");

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

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

}

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

{ sumOfSub(s+w[k],k+1,r-w[k]); }

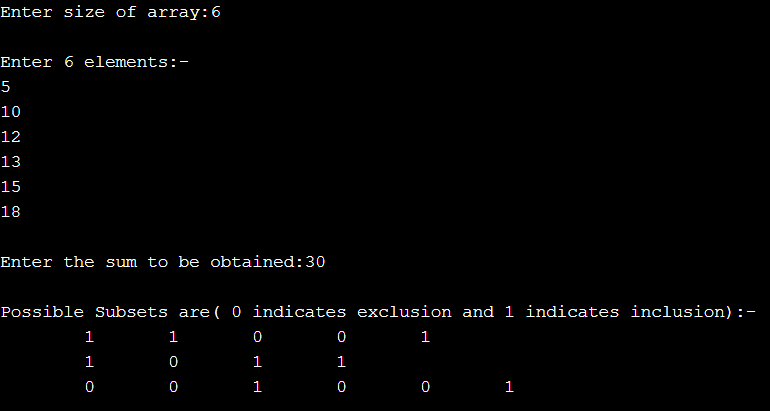
if((s+r-w[k])>=m && (s+w[k+1])<=m)

{ x[k]=0;

sumOfSub(s,k+1,r-w[k]);

}}

**OUTPUT-**



**EXPERIMENT – 19: Write a program to implement N Queens Problem Using Backtracking.**

#include <stdio.h>

int N;

int board[100][100];

int is\_attack(int i,int j)

{

int k,l;

for(k=0;k<N;k++)

{

if((board[i][k] == 1) || (board[k][j] == 1))

return 1;

}

for(k=0;k<N;k++)

{

for(l=0;l<N;l++)

{

if(((k+l) == (i+j)) || ((k-l) == (i-j)))

{

if(board[k][l] == 1)

return 1;

}

}

}

return 0;

}

int N\_queen(int n)

{

int i,j;

if(n==0)

return 1;

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

{

for(j=0;j<N;j++)

{

if((!is\_attack(i,j)) && (board[i][j]!=1))

{

board[i][j] = 1;

if(N\_queen(n-1)==1)

{

return 1;

}

board[i][j] = 0;

}

}

}

return 0;

}

int main()

{

printf("Enter the value of N for NxN chessboard:");

scanf("%d",&N);

int i,j;

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

{

for(j=0;j<N;j++)

{

board[i][j]=0;

}

}

N\_queen(N);

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

{

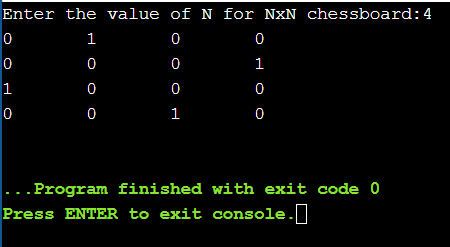
for(j=0;j<N;j++)

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

printf("\n");

}

**OUTPUT-**

****