Week 11

Q1) Given a sequence of matrices, write an algorithm to find most efficient way to multiply these matrices together. To find the optimal solution, you need to find the order in which these matrices should be multiplied.

Input Format:

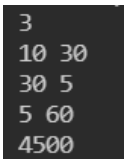
First line of input will take number of matrices n that you need to multiply. For each line i in n, take two inputs which will represent dimensions aXb of matrix i.

Output Format:

Output will be the minimum number of operations that are required to multiply the list of matrices.

```
#include<bits/stdc++.h>
using namespace std;
long matChainOrder(int *p,int n)     {
   int m[n][n];
   int i,j,k,l,q;
   for(i=1;i<n;i++)
   m[i][i]=0;
   for(l=2;l<n;l++)
   {
      for(i=1;i<n-l+1;i++)
      {
         j=i+l-1;
         m[i][j]=INT_MAX;
         for(k=i;k<=j-1;k++)
         {
            q=m[i][k]+m[k+1][j]+p[i-1]*p[k]*p[j];
            if(q<m[i][j])
            m[i][j]=q;
         }
      }
   }
   return m[1][n-1];
}
int main()
{
   int n;
   cin>>n;
   int p[n+1];
   for(int i=0;i<n;i++)
   {
      cin>>p[i]>>p[i+1];
   }
   cout<<matChainOrder(p,n+1);
   return 0;}
```

OUTPUT



Q2) Given a set of available types of coins. Let suppose you have infinite supply of each type of coin. For a given value N, you have to Design an algorithm and implement it using a program to find number of ways in which these coins can be added to make sum value equals to N.

Input Format:

First line of input will take number of coins that are available.
Second line of input will take the value of each coin.
Third line of input will take the value N for which you need to find sum.

Output Format:

Output will be the number of ways.

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   int n,amt;
   cin>>n;
   int i,j,a[n];
   for(i=0;i<n;i++)
   cin>>a[i];
   cin>>amt;
   int ans[amt+1];
   for(i=1;i<=amt;i++)
   ans[i]=0;
   ans[0]=1;
   for(j=0;j<n;j++)
   {
      for(i=1;i<=amt;i++)
      {
         if(a[j]<=i)
         ans[i]+=(ans[i-a[j]]);
      }
   }
   cout<<ans[amt];
   return 0;
}
```

OUTPUT

```
4
2 5 6 3
10
5
```

Q3) Given a set of elements, you have to partition the set into two subsets such that the sum of elements in both subsets is same. Design an algorithm and implement it using a program to solve this problem.

Input Format:

First line of input will take number of elements n present in the set.
Second line of input will take n space-separated elements of the set.

Output Format:

Output will be 'yes' if two such subsets found otherwise print 'no'.

```
#include<bits/stdc++.h>
using namespace std;
int main()  {
   int n;
   cin>>n;
   int i,j,a[n];
   for(i=0;i<n;i++)
   cin>>a[i];
   int sum=0;
   for(i=0;i<n;i++)
   sum+=a[i];
   if(sum%2!=0)
   {
      cout<<"no";
      return 0;
   }
   sum=sum/2;
   bool s[n+1][sum+1];
   for(i=0;i<=n;i++)
   {
      for(j=0;j<=sum;j++)
      {
         if(j==0)
         s[i][j]=1;
         else if(i==0)
         s[i][j]=0;
         else
         {
            if(a[i-1]>j)
            s[i][j]=s[i-1][j];
            else
            s[i][j]=(s[i-1][j] || s[i-1][j-a[i-1]]);
         }
      }
```

```
    }
    if(s[n][sum])
    cout<<"yes";
    else
    cout<<"no";
    return 0;
}
```

OUTPUT