

///problem 01

///ID: 011201031 Hasibul Hasan Nirob

#include<bits/stdc++.h>

using namespace std;

```
int subSetSum(int price[],int T,int N){
    bool subset[N+1][T+1];
    int cnt[N+1][T+1];
    for (int i=0; i<=T; i++){
        subset[0][i]=true;
        cnt[0][i] = 0;
    }
    for (int i=1; i<=N; i++){
        subset[i][0] = false;
        cnt[i][0] = -1;
    }
    for (int i = 1; i <= N; i++)
    {
        for (int j = 1; j <=T; j++)
        {
            subset[i][j] = subset[i][j - 1];
            cnt[i][j] = cnt[i][j - 1];
            if (i >= price[j - 1])
            {
                subset[i][j] = subset[i][j] ||
                    subset[i - price[j - 1]][j - 1];

                if (subset[i][j])
                    cnt[i][j] = max(cnt[i][j - 1],
                        cnt[i - price[j - 1]][j - 1] + 1);
            }
        }
    }
}
```

```

    }
}
}

int knapsack(int T, int price[], int value[],int N){
    int i,w;
    vector<vector<int>> K(N+1,vector<int>(T+1));
    for(i = 0; i <= N; i++){
        for(w = 0; w <= T; w++){
            if (i == 0 || w == 0)
                K[i][w] = 0;
            else if (price[i - 1] <= w)
                K[i][w] = max(value[i - 1] +
                               K[i - 1][w - price[i - 1]],
                               K[i - 1][w]);
            else
                K[i][w] = K[i - 1][w];
        }
    }
    return K[N][T];
}

```

```

int main(){

    int N,T,i,j;
    cout<<"Give the value of N: ";
    cin>>N;
    cout<<"Give the value of T: ";
    cin>>T;
}

```

```

int value[N],price[N];

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

    cout<<"Value "<<i+1<<": ";

    cin>>value[i];

}

cout<<endl;

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

    cout<<"Price "<<j+1<<": ";

    cin>>price[j];

}

N=subSetSum(price,T,N);

cout<<"\nMinimum value of the sum of prices is : "<<knapsack(T,price,value,N)<<endl;

return 0;

}

```