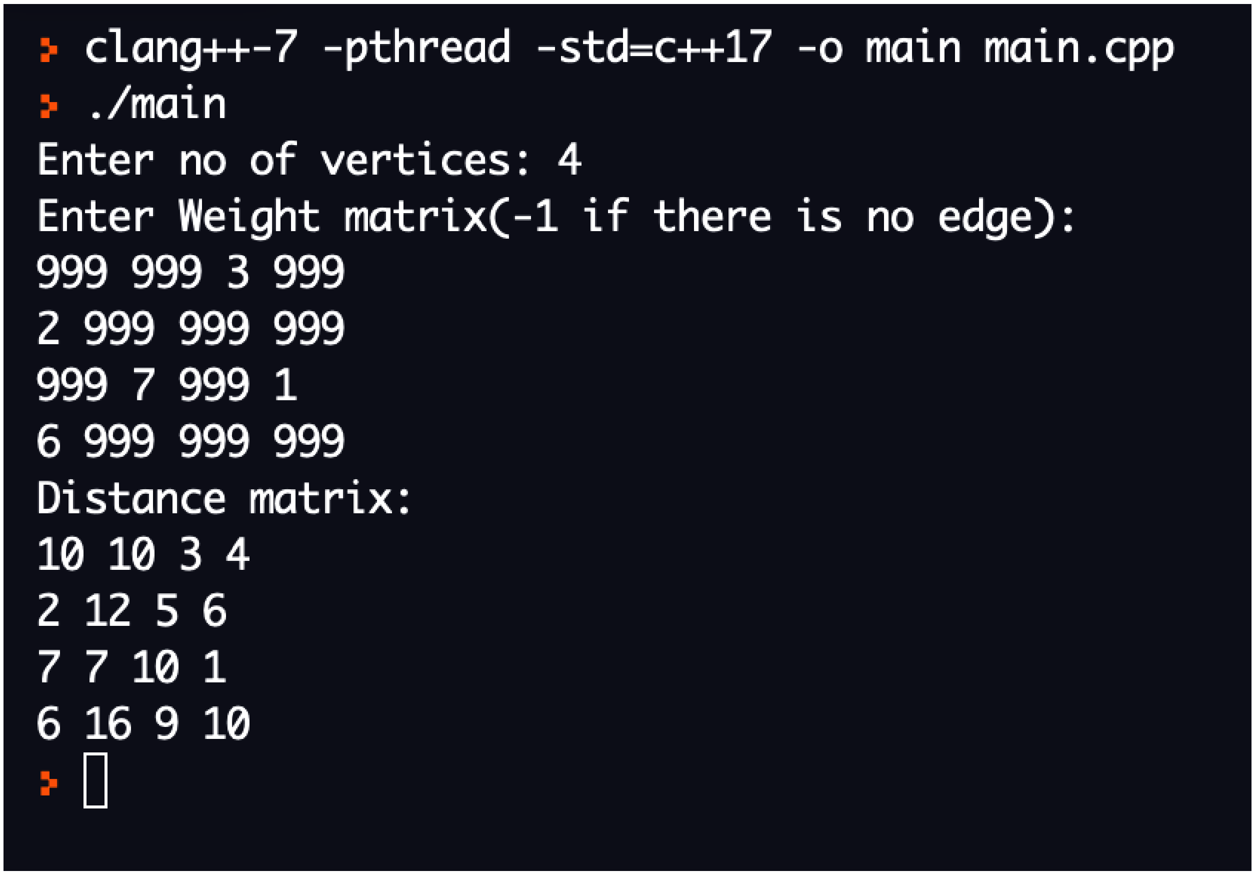
# PROGRAM 13

## Implement All Pair Shortest paths problem using Floyd’s algorithm.

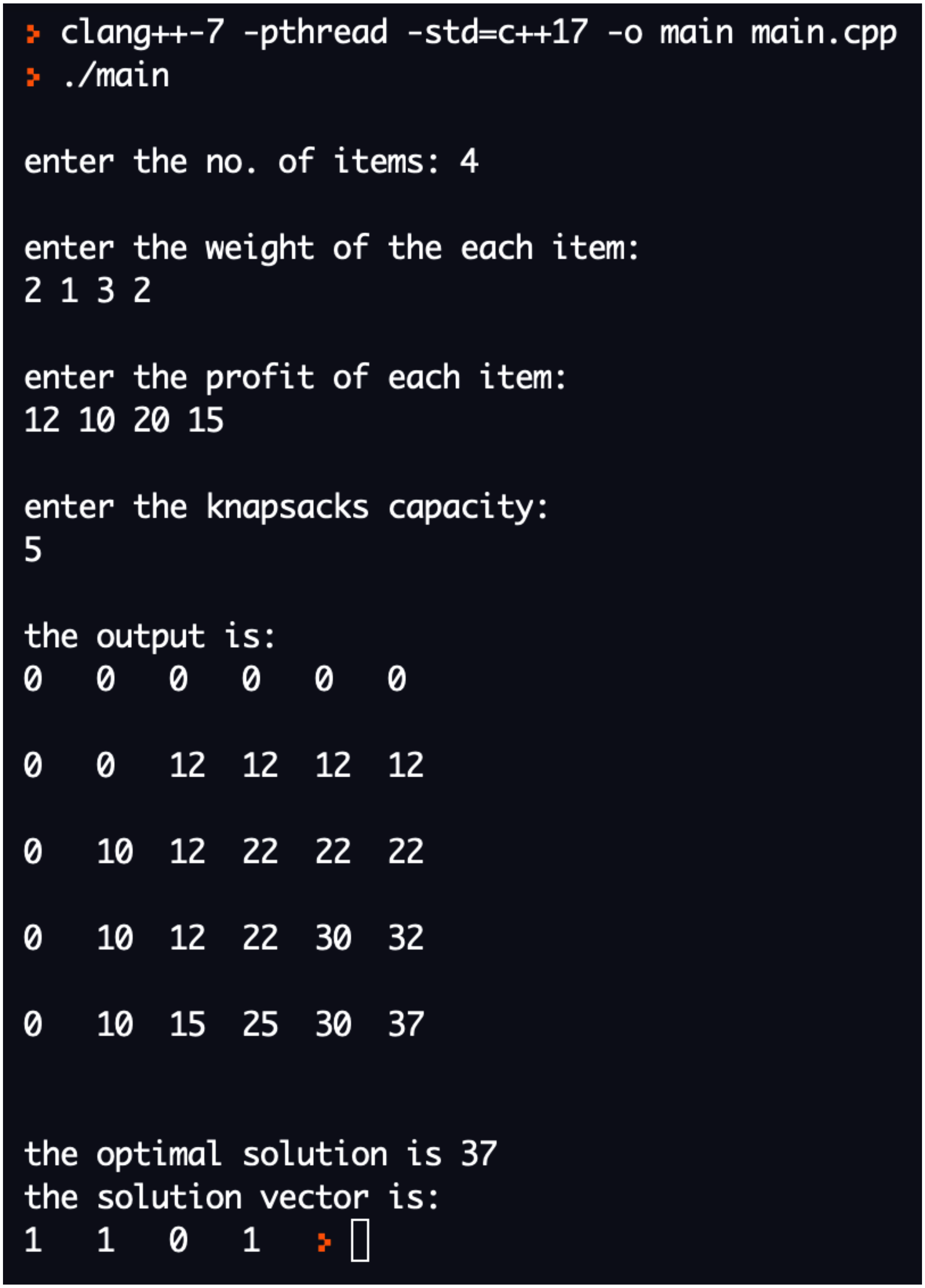
//Code

```
#include<iostream>
#define inf 9999
using namespace std;
int a[10][10],d[10][10][10];

void floyds(int n){
  int k=0;
  for(int i=1;i<=n;i++)
      for(int j=1;j<=n;j++)
          d[k][i][j]=a[i][j];
  for(k=1;k<=n;k++)
      for(int i=1;i<=n;i++)
          for(int j=1;j<=n;j++)
              d[k][i][j]=min(d[k-1][i][j],(d[k-1][i][k]+d[k-1][k][j]));
}

int main(){
  int n;
  cout<<"Enter no of vertices: ";
  cin>>n;
  cout<<"Enter Weight matrix(-1 if there is no edge): ";
  for(int i=1;i<=n;i++)
      for(int j=1;j<=n;j++){
          cin>>a[i][j];
          if(a[i][j]==-1)
              a[i][j]=inf;
      }
  floyds(n);
  cout<<"Distance matrix: "<<endl;
  for(int i=1;i<=n;i++){
    for(int j=1;j<=n;j++){
      if(d[n][i][j]>=inf)
        cout<<(-1)<<" ";
      else
        cout<<d[n][i][j]<<" ";
    }
    cout<<endl;
  }
}
```

//Output



```
› clang++-7 -pthread -std=c++17 -o main main.cpp
› ./main
Enter no of vertices: 4
Enter Weight matrix(-1 if there is no edge):
999 999 3 999
2 999 999 999
999 7 999 1
6 999 999 999
Distance matrix:
10 10 3 4
2 12 5 6
7 7 10 1
6 16 9 10
› 
```



```
› clang++-7 -pthread -std=c++17 -o main main.cpp
› ./main

enter the no. of items: 4

enter the weight of the each item:
2 1 3 2

enter the profit of each item:
12 10 20 15

enter the knapsacks capacity:
5

the output is:
0   0   0   0   0   0

0   0   12  12  12  12

0   10  12  22  22  22

0   10  12  22  30  32

0   10  15  25  30  37


the optimal solution is 37
the solution vector is:
1   1   0   1   › 
```