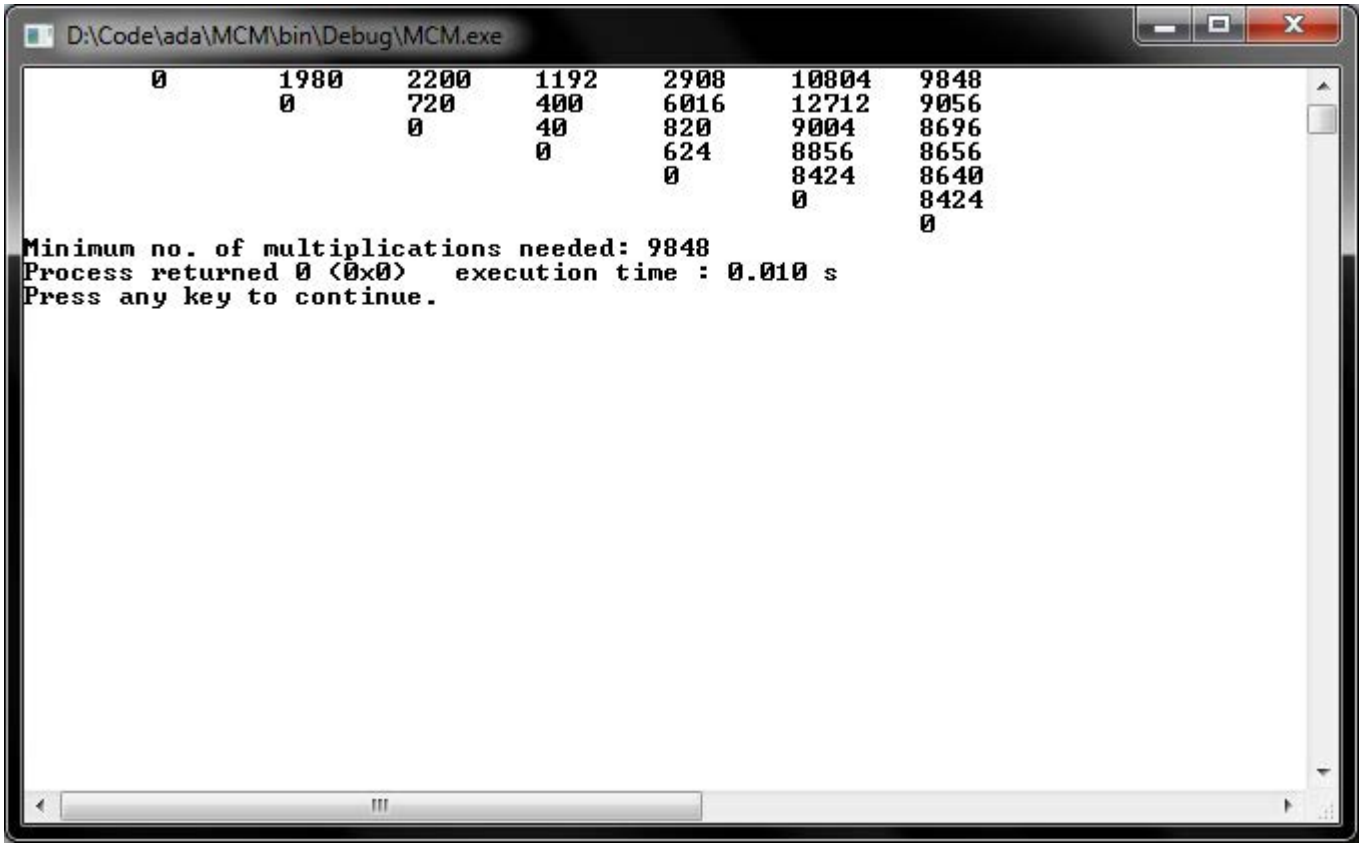
Matrix Chain Multiplication

```
#include <iostream>
#include <vector>
using namespace std;
typedef vector<long long> vi;
const LL inf = 1LL << 60;

void print(vii M) {
   for(LL i=1; i<M.size(); i++) {
      for(LL j=0; j<M[0].size(); j++) {
         if(j < i) {
            cout<<"\t";
         } else {
            cout<<M[i][j]<<"\t";
         }
      }
      cout<<endl;
   }
}

LL MCM(LL n, vi V) {
   vii M(n+1,vi(n+1,inf));
   fromeq(0,n,i) {
      M[i][i] = 0;
   }

   for(LL l = 1; l<=n; l++) {  //No of matrices to be multiplied, i.e range of j
      LL depth = n-l; //Range till which i will go
      for(LL i=1; i<=depth; i++) {
         LL j=i+l;
         for(LL k=i; k<j; k++) {
            M[i][j] = min(M[i][j], M[i][k] + M[k+1][j] + V[i-1]*V[k]*V[j]);
         }
      }
   }
   print(M);
   return M[1][n];
}

int main() {
   vi V = {11,36,5,4,2,78,54,2};
   LL n = V.size()-1; //No. of matrices
   cout<<"Minimum no. of multiplications needed: "<<MCM(n,V);
   return 0;
}
```

Output:


D:\Code\ada\MCM\bin\Debug\MCM.exe
0 1980 2200 1192 2908 10804 9848
0 720 400 6016 12712 9056
0 40 820 9004 8696
0 624 8856 8656
0 8424 8640
0 8424
0
Minimum no. of multiplications needed: 9848
Process returned 0 (0x0) execution time : 0.010 s
Press any key to continue.