```
//Dynamic programming code.....
//sample graph: C->A->N->L->D->B
//             2  1  6  12 1
//               2  8  12 12
#include<stdio.h>
int main() {
 int n = 6;
 int g[n][n]; //graph in matrix form...
 int c[n];
 int p[n];
 int i, j;

 for (i = 0; i < n; i++) {
  for (j = 0; j < n; j++) {
   scanf("%d", &g[i][j]);
  }
 }

 int c1;   //cost from i-1 city/node
 int c2;  //cost from i-2 city/node;
 int min, minI;

 c[0] = 0;
 p[0] = -1;
 for (i = 1; i < n; i++) {
  c1 = c[i-1] + g[i-1][i];
  min = c1;
  minI = i-1;

  if (i - 2 >= 0) {
   c2 = c[i-2] + g[i-2][i];
   if (min > c2) {
    min = c2;
    minI = i-2;
   }
  }
  c[i] = min;
  p[i] = minI;
 }

 printf("Cost array: ");
 for (i = 0; i < n; i++) {
```

```
    printf("%d ", c[i]);
  }
  printf("\n");

  printf("Cost from 0th city to %dth city is %d\n", n-1, c[n-1]);

  int prev = p[n-1];
  printf("Path %d %d ", n-1, prev);
  while (prev != -1) {
    prev = p[prev];
    printf("%d ",prev);
  }
  printf("\n");

}
```