



```
1 #include <limits.h>
2 #include <stdio.h>
3
4 int min(int x, int y, int z);
5
6
7 int minCost(int cost[R][C], int m, int n)
8 {
9     if (n < 0 || m < 0)
10         return INT_MAX;
11     else if (m == 0 && n == 0)
12         return cost[m][n];
13     else
14         return cost[m][n]
15             + min(minCost(cost, m - 1, n - 1),
16                 minCost(cost, m - 1, n),
17                 minCost(cost, m, n - 1));
18 }
19 int min(int x, int y, int z)
20 {
21     if (x < y)
22         return (x < z) ? x : z;
23     else
24         return (y < z) ? y : z;
25 }
26
27 int main()
28 {
29     int cost[R][C]
30         = { { 1, 2, 3 }, { 4, 8, 2 }, { 1, 5, 3 } };
31     printf(" %d ", minCost(cost, 2, 2));
32     return 0;
33 }
```

/tmp/kwKA9RCxvS.o

8

 Introducing the most  
interactive C Course.  
Learn practically and get certified.

[Get Started!](#)



main.c



Run

Output

Clear

```
1  #include<stdio.h>
2  int main()
3  {
4  int n,sum=0,m;
5  printf("Enter a number:");
6  scanf("%d",&n);
7  while(n>0)
8  {
9  m=n%10;
10 sum=sum+m;
11 n=n/10;
12 }
13 printf("Sum is=%d",sum);
14 return 0;
15 }
```

/tmp/kwKA9RCXvS.o

Enter a number:693258

Sum is=33|

main.c

```
45     printf("[%d][%d]: ", i, j);
46     scanf("%d", &graph[i][j]);
47 }
48 }
49 printf("The original graph is:\n");
50 for (i = 0; i < n; i++)
51 {
52     for (j = 0; j < n; j++)
53     {
54         printf("%d ", graph[i][j]);
55     }
56     printf("\n");
57 }
58 floydWarshall(graph, n);
59 printf("The shortest path matrix is:\n");
60 for (i = 0; i < n; i++)
61 {
62     for (j = 0; j < n; j++)
63     {
64         printf("%d ", graph[i][j]);
65     }
66     printf("\n");
67 }
68 return 0;
69 }
```

Run

Output

```
/tmp/wgiNGu1rKm.o
Enter the number of vertices: 3
Enter the edges:
[0][0]: 2
[0][1]: 3
[0][2]: 1
[1][0]: 4
[1][1]: 5
[1][2]: 6
[2][0]: 7
[2][1]:
7
[2][2]: 8
The original graph is:
2 3 1
4 5 6
7 7 8
The shortest path matrix is:
2 3 1
4 5 5
7 7 8
|
```

Get Started!

main.c



Run

```
1 #include <stdio.h>
2
3 void main()
4 {
5     int no_row,c=1,blk,i,j;
6     printf("Input number of rows: ");
7     scanf("%d",&no_row);
8     for(i=0;i<no_row;i++)
9     {
10         for(blk=1;blk<=no_row-i;blk++)
11             printf(" ");
12         for(j=0;j<=i;j++)
13         {
14             if (j==0||i==0)
15                 c=1;
16             else
17                 c=c*(i-j+1)/j;
18             printf("% 4d",c);
19         }
20         printf("\n");
21     }
22 }
```

Output

/tmp/wgiNGu1rKm.o

Input number of rows: 6

1

```
      1  1
     1  2  1
    1  3  3  1
   1  4  6  4  1
  1  5 10 10  5  1
```

 **Introducing the most interactive C Course.**

Learn practically and get certified.

**Get Started!**

```
9  scanf("%d", &end);
10
11  printf("All prime numbers between 1 to %d are:\n", end);
12
13  /* Find all Prime numbers between 1 to end */
14  for(i=2; i<=end; i++)
15  {
16      /* Assume that the current number is Prime */
17      isPrime = 1;
18
19      /* Check if the current number i is prime or not */
20      for(j=2; j<=i/2; j++)
21      {
22          /*
23           * If i is divisible by any number other than 1 and self
24           * then it is not prime number
25           */
26          if(i%j==0)
27          {
28              isPrime = 0;
29              break;
30          }
31      }
32
33      /* If the number is prime then print */
34      if(isPrime==1)
35      {
36          printf("%d, ", i);
37      }
38  }
39
40  return 0;
41 }
```

/tmp/H4QR8v5SQB.o

Find prime numbers between 1 to : 50

All prime numbers between 1 to 50 are:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,