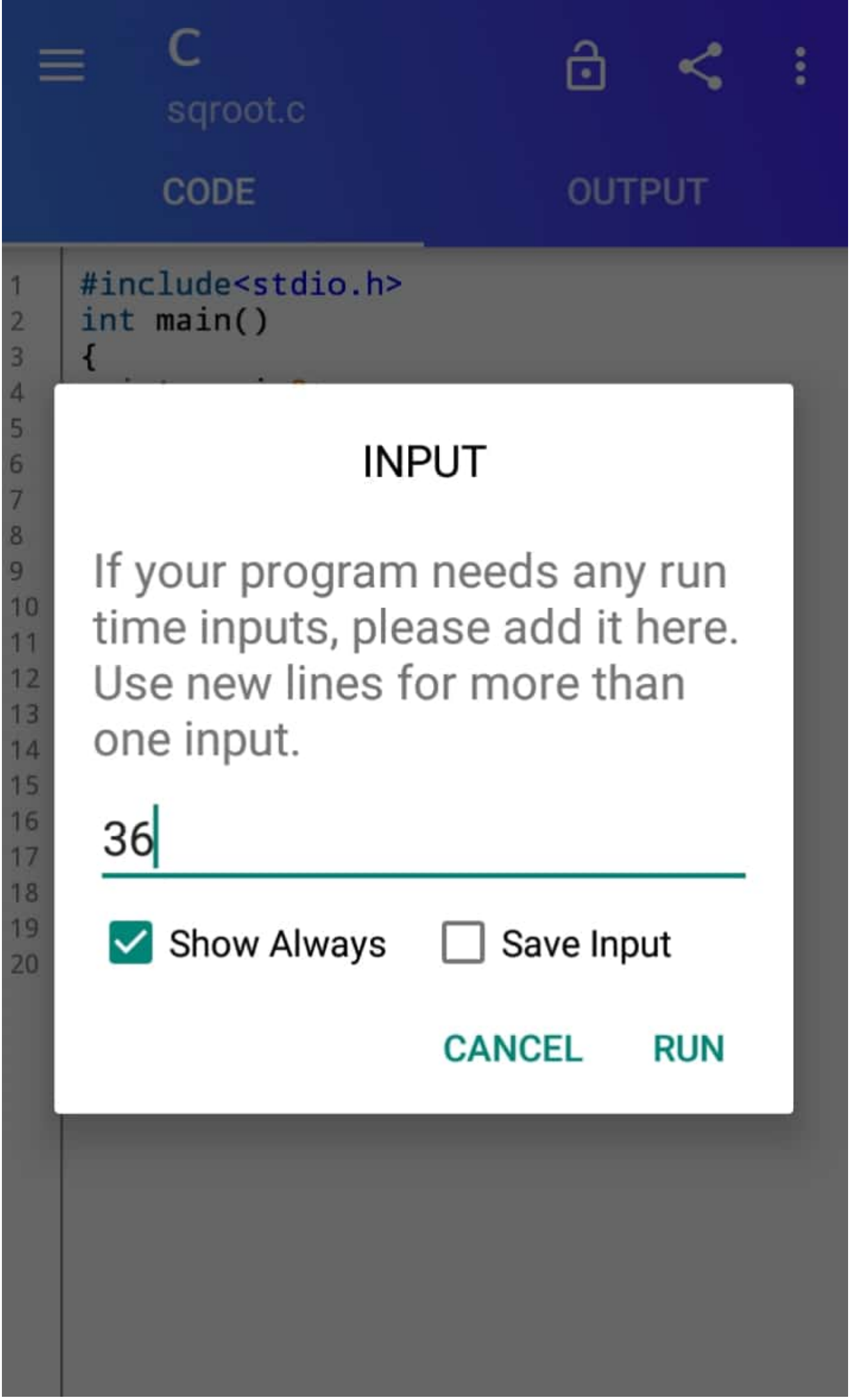
C

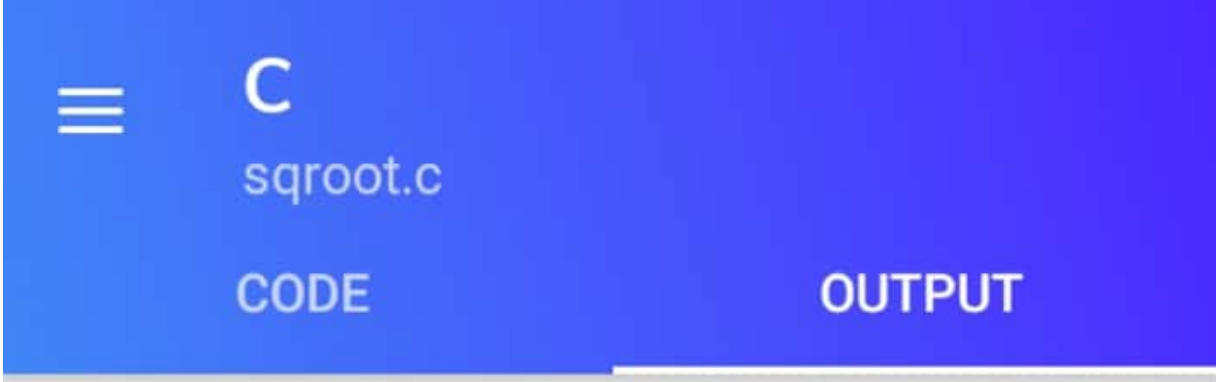
sqroot.c

CODE | OUTPUT

```c
#include<stdio.h>
int main()
{
  int n ,i=0 ;
  int root=1;
  printf("enetr number ");

  scanf("%d",&n);
  while(1)
  {
    i=i+1;
    root = (n/root+root)/2;
    if(i == n+1)
    {
      break;
    }

  }
  printf("%d",root);
  }
```

TAB { } ( ) " & RUN






```
enetr number 6
```

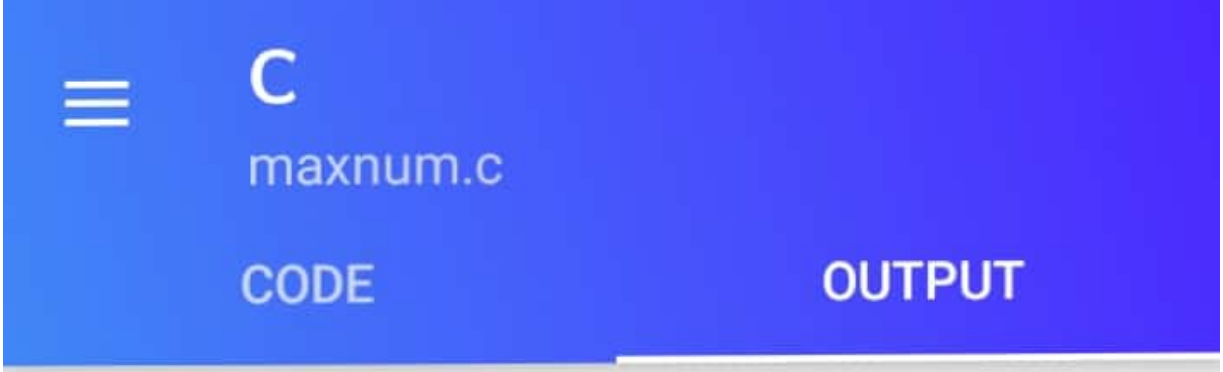
C

maxnum.c

CODE OUTPUT

```c
//3. MAXIMUM HIGHT OF STAIRS
#include<stdio.h>
int main()
{
    int blocks=20;
    int stairs=0;
    while(blocks!=0)
    {
        if(stairs+1<= blocks)
        {
            stairs++;
            blocks=blocks-stairs;
        }
        else
        {
            break;
        }
    }
    printf("no of stairs are %d",stairs);
}
```

TAB { } ( ) " & RUN




```
no of stairs are 5
```

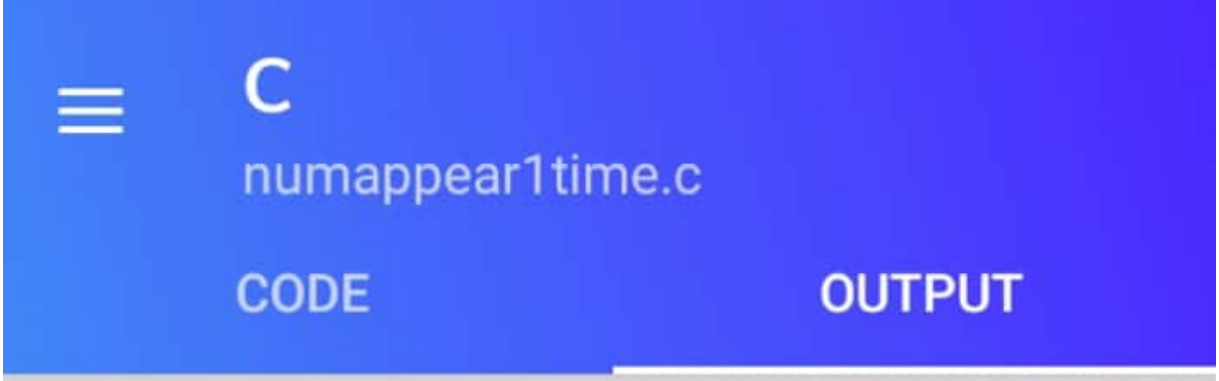
C

numappear1time.c

CODE OUTPUT

```c
 //1. number which appears one time
#include<stdio.h>
int main()
    {
        int a[]={5,3,3,4,5,2,4};

        int res=a[0];
        for(int i=1; i<7; i++)
        {
            res=res^a[i];
        }
        printf("single repeated element is %d",res);
    }
```

TAB { } ( ) " & RUN




```
single repeated element is 2
```

C

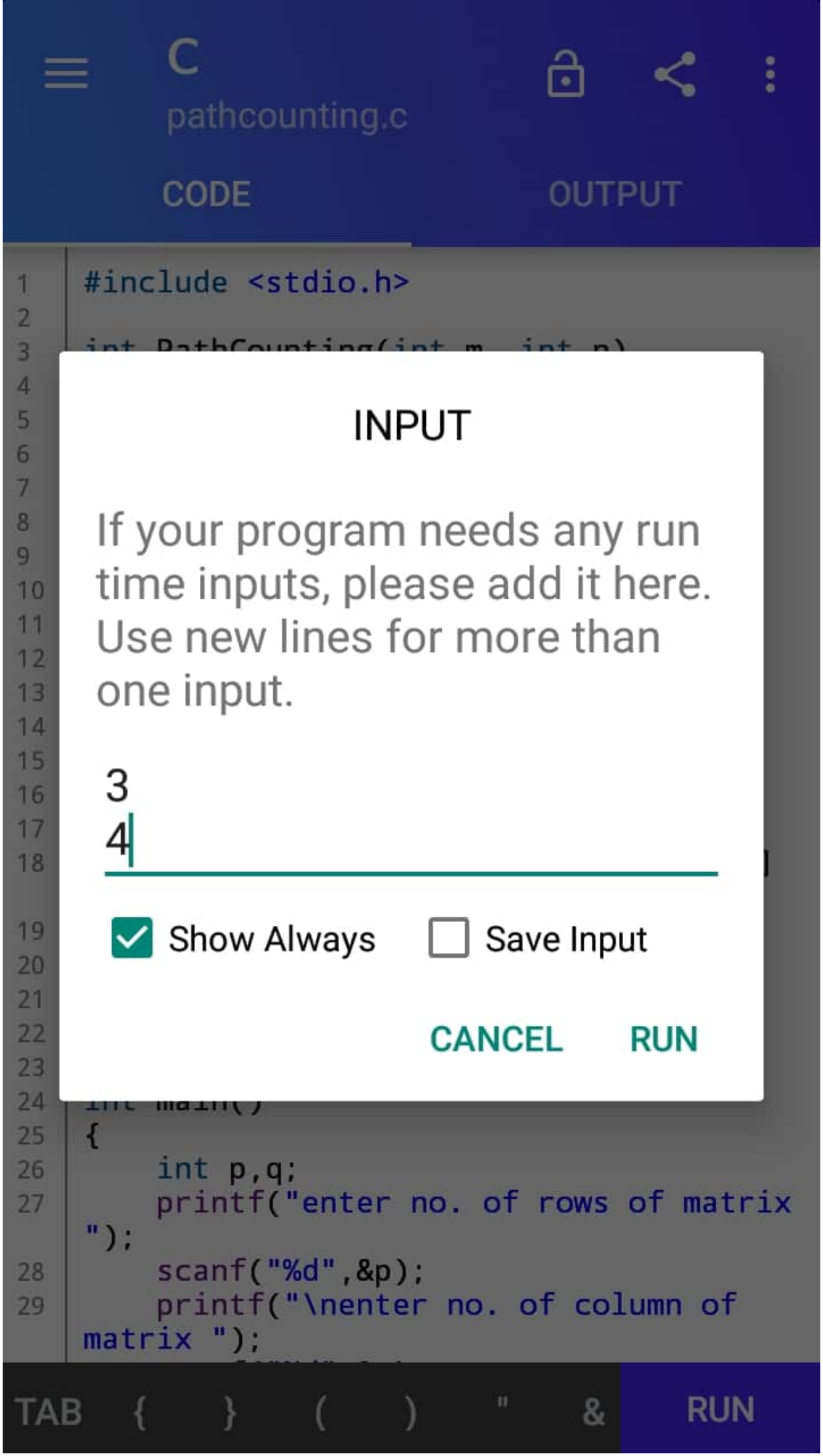
pathcounting.c

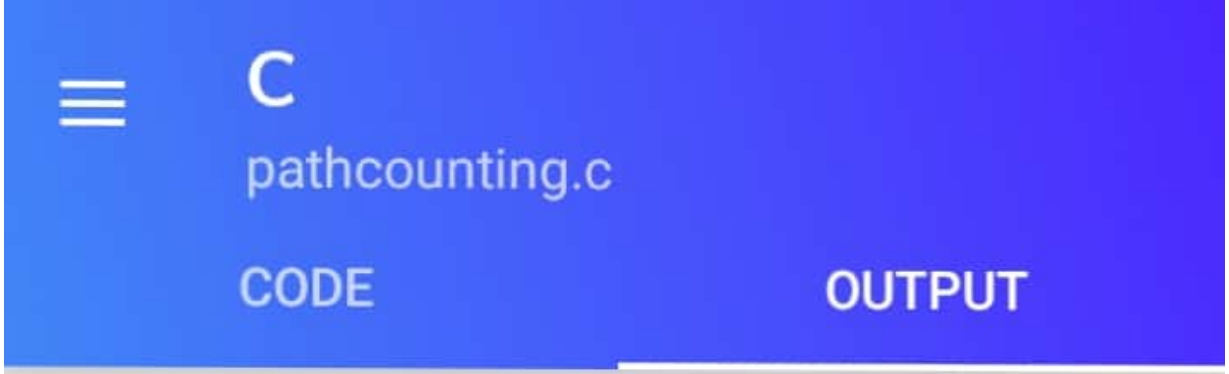
CODE OUTPUT

```c
#include <stdio.h>

int PathCounting(int m, int n)
{
    int ctr[m][n];
    for (int i = 0; i < m; i++)
        {
    ctr[i][0] = 1;
    }
    for (int j = 0; j < n; j++)
        {
    ctr[0][j] = 1;
    }
    for (int i = 1; i < m; i++)
    {
        for (int j = 1; j < n; j++)
            {
      ctr[i][j] = ctr[i-1][j] + ctr[i]
[j-1];
       }
    }
    return ctr[m-1][n-1];
}

int main()
{
    int p,q;
    printf("enter no. of rows of matrix
");
    scanf("%d",&p);
    printf("\nenter no. of column of
matrix ");
```

TAB { } ( ) " & RUN

```
8       ctr[i][0] = 1;
9       }
10      for (int j = 0; j < n; j++)
11          {
12      ctr[0][j] = 1;
13      }
14      for (int i = 1; i < m; i++)
15      {
16          for (int j = 1; j < n; j++)
17              {
18        ctr[i][j] = ctr[i-1][j] + ctr[i]
[j-1];
19        }
20      }
21      return ctr[m-1][n-1];
22  }
23
24  int main()
25  {
26      int p,q;
27      printf("enter no. of rows of matrix
");
28      scanf("%d",&p);
29      printf("\nenter no. of column of
matrix ");
30      scanf("%d",&q);
31    printf("\nThe size of matrix is : %d,
%d\n",p,q);
32    printf("\nThe all possible paths from
top left to bottom right is: %d
\n",PathCounting(p,q));
33  }
```






```
enter no. of rows of matrix
enter no. of column of matrix
The size of matrix is : 3, 4

The all possible paths from top
left to bottom right is: 10
```