



C

sqroot.c



CODE

OUTPUT

```
1  #include<stdio.h>
2  int main()
3  {
4      int n ,i=0 ;
5      int root=1;
6      printf("enetr number ");
7
8      scanf("%d",&n);
9      while(1)
10     {
11         i=i+1;
12         root = (n/root+root)/2;
13         if(i == n+1)
14         {
15             break;
16         }
17     }
18     printf("%d",root);
19 }
20
```

TAB

{

}

(

)

"

&amp;

RUN



C

sqroot.c



CODE

OUTPUT

```
1 #include<stdio.h>
2 int main()
3 {
4
```

## INPUT

If your program needs any run time inputs, please add it here. Use new lines for more than one input.

36



Show Always



Save Input

CANCEL

RUN



C

sqroot.c

CODE

OUTPUT

```
enetr number 6
```



C

maxnum.c



CODE

OUTPUT

```
1 //3. MAXIMUM HIGHT OF STAIRS
2 #include<stdio.h>
3 int main()
4 {
5     int blocks=20;
6     int stairs=0;
7     while(blocks!=0)
8     {
9         if(stairs+1<= blocks)
10         {
11             stairs++;
12             blocks=blocks-stairs;
13         }
14         else
15         {
16             break;
17         }
18     }
19     printf("no of stairs are
20 %d",stairs);
21 }
```

TAB

{

}

(

)

"

&amp;

RUN



C

maxnum.c

CODE

OUTPUT

no of stairs are 5



C

numappear1time.c



CODE

OUTPUT

```
1
2 //1. number which appears one time
3 #include<stdio.h>
4 int main()
5 {
6     int a[]={5,3,3,4,5,2,4};
7
8     int res=a[0];
9     for(int i=1; i<7; i++)
10    {
11        res=res^a[i];
12    }
13    printf("single repeated element
14 is %d",res);
15    }
16
17
```

TAB

{

}

(

)

"

&amp;

RUN



C

numappear1time.c

CODE

OUTPUT

single repeated element is 2



C

pathcounting.c



CODE

OUTPUT

```
1  #include <stdio.h>
2
3  int PathCounting(int m, int n)
4  {
5      int ctr[m][n];
6      for (int i = 0; i < m; i++)
7      {
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]
19 [j-1];
20         }
21     }
22     return ctr[m-1][n-1];
23 }
24
25 int main()
26 {
27     int p,q;
28     printf("enter no. of rows of matrix
29 ");
30     scanf("%d",&p);
31     printf("\nenter no. of column of
32 matrix ");
```

TAB

{

}

(

)

"

&amp;

RUN





C

pathcounting.c



CODE

OUTPUT

```
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]
19 [j-1];
20         }
21     }
22     return ctr[m-1][n-1];
23 }
24 int main()
25 {
26     int p,q;
27     printf("enter no. of rows of matrix
28 ");
29     scanf("%d",&p);
30     printf("\nenter no. of column of
31 matrix ");
32     scanf("%d",&q);
33     printf("\nThe size of matrix is : %d,
34 %d\n",p,q);
35     printf("\nThe all possible paths from
36 top left to bottom right is: %d
37 \n",PathCounting(p,q));
38 }
```

TAB

{

}

(

)

"

&amp;

RUN



C

pathcounting.c



CODE

OUTPUT

```
1 #include <stdio.h>
```

```
2  
3 int PathCounting(int m, int n)
```

## INPUT

If your program needs any run time inputs, please add it here. Use new lines for more than one input.

3

4



Show Always



Save Input

CANCEL

RUN

```
24 int main()
```

```
25 {
```

```
26     int p,q;
```

```
27     printf("enter no. of rows of matrix  
28 ");
```

```
29     scanf("%d",&p);
```

```
printf("\nenter no. of column of  
matrix ");
```

TAB

{

}

(

)

"

&amp;

RUN



# C

pathcounting.c

CODE

OUTPUT

```
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
```