



# C++ PROGRAMMING (43)

## Technical Task

# KEY

Regional – 2013

---

***Judges/Graders:***

Please double-check and verify all scores!

Property of Business Professionals of America.  
May be reproduced only for use in the Business Professionals of America  
*Workplace Skills Assessment Program* competition.



## C++ Programming

### Technical Task Scoring Sheet:

• Application reads input that is comma separated from command line .....	_____	15 pts
• Application loops until all zeros are read .....	_____	15 pts
• Application correctly determines the number of moves ...	_____	50 pts
• Application correctly determines KING .....	_____	50 pts
• Application displays number of moves and KING when applicable .....	_____	25 pts
Application exits program when all 0's entered with no output .....	_____	15 pts
Code uses a data structure to represent checker board .....	_____	20 pts
Code is well commented .....	_____	10 pts
Code uses proper variable naming conventions .....	_____	5 pts
Program runs from USB drive .....	_____	5 pts
<b>Total Points:</b>	_____	<b>210 pts</b>

Sample Solution code. Contestant code may vary.

```
/* Comments at top to tell what program does
   Contestant Number included */

#include <iostream>

using namespace std;

void input();
bool goRight();
bool goLeft();

int row, column;
int numChecker;
int matrix[8][8];

void main()
{
    int legalJump;
    bool king;

    do
    {
        for (int r=0;r<8;r++)
            for (int c=0;c<8;c++)
                matrix[r][c]=0;

        input();
        legalJump=0;
        king=false;
    }
}
```



```
        if (row!=0){
            for (int i=0;(i<numChecker && row!=8);i++){
                if ((matrix[row+1][column-1]==1) &&
goLeft())
                    {
                        if (matrix[row+2][column-2]!=1){
                            legalJump++;
                            row+=2;
                            column-=2;
                        }
                    }
                else if ((matrix[row+1][column+1]==1) &&
goRight())
                    {
                        if (matrix[row+2][column+2]!=1){
                            legalJump++;
                            row+=2;
                            column+=2;
                        }
                    }
                }
            if (row==8)
                king=true;
        }
        cout<<legalJump;
        if (king)
            cout<<" , KING\n";
        else
            cout<<endl;
    }
}while (row!=0);
}

//comments to tell what function does
void input()
{
    int temp1, temp2;
    cin>>row;
    cin.ignore(1, ',');
    if (row !=0){
        cin>>column;
        cin.ignore(1, ',');
        cin>>numChecker;
        cin.ignore(1, ',');
        for (int i=0;i<numChecker;i++){
            cin>>temp1;
            cin.ignore(1, ',');
            cin>>temp2;
            cin.ignore(1, ',');
```



```
                matrix[temp1][temp2]=1;
            }
        }
    }
    //comment to tell what function does
    bool goRight()
    {
        if ((row +2<= 8) && (column+2 <=8))
            return true;
        else
            return false;
    }
    //comment to tell what function does
    bool goLeft()
    {
        if ((row+2 <=8) && (column-2 >0))
            return true;
        else
            return false;
    }
}
```

Note to Grader: When running the application, use the following test data:

Input: 1,7,2,2,6,4,4

Output: 2

Input: 2,2,3,3,3,5,5,7,7

Output: 3, KING

Input: 1,5,4,2,4,2,6,3,7,3,3

Output: 0

Input: 1,3,5,2,4,3,5,2,2,4,2,5,3

Output: 1

Input: 2,2,5,3,3,5,5,7,7,3,1,4,2

Output: 3, KING

Input: 0,0,0

Output: <line should be blank>