Mastermind

Introduction

The mastermind game originated as a wooden board game where one player set the code hidden on one side of the board and the other had to attempt to crack the code and solve the puzzle and the players competed to complete the game in the least number of turns possible. The other player, after each round, tells the player attempting to crack the code how many are in the right position and how many are the correct color but in the wrong position. This game is a computer rendition of the classic board game and emulates the computer version of the game which can be found online by the name of code breakers. This game includes a game board made of ascii characters that resembles the gameboard on both the classic and computer game and shows the colors the user entered on the board. I don't know how to make graphics or windows sadly.

Game Rules

1.) There are 6 possible colors

2.) There are 4 spaces per row that the colors can be assigned to

3.) The player must guess and try to match the correct sequence of colors to the hidden answer key

4.) Check the game board after each round to see how you are doing

5.) The player gets a maximum of eight turns to solve the mastermind code puzzle

C++ Code

```
/*
 * File:   main.cpp
 * Author: Sebastian Hall
 * Created on April 25, 2018, 6:48 PM
 * Purpose: Mastermind Computer Game Replica With 6 Colors,
 *          4 Columns, And 8 Rows
 */

//System Libraries
#include <iostream> //For Output
#include <cstdlib>  //For Random Functions
#include <limits>   //For numeric_limits

//User Libraries
#include "functions.h"



int main(){
   //Declare And Initialize Variables
   short const COLS=4;//The Number Of Columns In The Game, Constant
   std::string r1 [COLS]={"   "," "   "," "   "},//First Row In The Game
            r2 [COLS]={"   "," "   "," "   "},//Second "    "    "
            r3 [COLS]={"   "," "   "," "   "},//Third  "    "    "
            r4 [COLS]={"   "," "   "," "   "},//Fourth "    "    "
            r5 [COLS]={"   "," "   "," "   "},//Fifth  "    "    "
            r6 [COLS]={"   "," "   "," "   "},//Sixth  "    "    "
            r7 [COLS]={"   "," "   "," "   "},//Seventh "    "    "
            r8 [COLS]={"   "," "   "," "   "},//Eighth "    "    "
            key [COLS]={"   "," "   "," "   "};//Game Winning Answer Key
   short rghtClr=0,//Number Of Correct Color, But Not Position
        rghtPos=0,//Number In Correct Position
        index=0,//Index For Looping Through The Rounds
        choice=0,//Menu Choice Variable
        rounds=8;//Number Of Rounds In The Game
   bool gameWin=false,//Boolean Value If The Game Is Won Or Not
       error=false;//Boolean Value If Error In User Input
   setWin(key);//Initialize Key Array
```

```
//Display Game Title
std::cout<<"\tWelcome To Mastermind\n"
         "\tPress Enter To Begin"<<std::endl;//Prompts User To Enter
std::cin.ignore();//Pauses Until Enter Is Hit


//Loop Game Rounds
do{//Do Until User Chooses To Or Must Leave Game
   do{//Do Until The Current Round Is Completed
      try{
         //Reset Error Flag
         error=false;

         //Output Main Play Menu
         std::cout<<"1.) View The Game Board"<<std::endl<<
                  "2.) Play Round "<<index+1<<std::endl<<
                  "3.) Quit The Game Early"<<std::endl<<
                  "\nChoice: ";

         //Get User Choice
         std::cin>>choice;//User Inputs Menu Choice

         if(std::cin.fail()||choice<1||choice>3){
            std::cin.clear();//Remove Any Buffer Error States
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(),'\n');
            throw 0;//Throw Exception
         }
      }
      catch(int x){//Catch Exception
         std::cout<<"\n\nThat's Not An Option Here\n\n";//Tell User
         error=true;//Set Error Flag To True
      }

      //Evaluate User Choice
      switch(choice){
         case 1:gmeScrn(r1,r2,r3,r4,r5,r6,r7,r8,
                        key,rghtPos,rghtClr);break;//Output The Board
         case 2:wchRow(r1,r2,r3,r4,r5,r6,r7,r8,
                        key,rghtPos,rghtClr,index);break;//Play Round
         case 3:std::cout<<"Farewell, Quitter\n\n\n";break;//Say Goodbye
      }
   }while(choice==1||error==true);//While Just Choosing To See Game Board
```

```
        //Check If User Won The Game
        if(rghtPos==4)//If All Are In The Right Position
            gameWin=true;//Set gameWin Boolean To True

        //Increment Index
        index++;//Index Incremented By 1
    }while(index<rounds&&!gameWin&&choice!=3);//While Not Any Ending Conditions



    //Acknowledge Users Win/Loss
    std::cout<<"Round "<<index<<std::endl;
    if(gameWin)//If The Game Was Won
        std::cout<<"Congratulations, You Won!\n";//Congratulate Winner
    else//Else The Game Was Lost
        std::cout<<"Congratulations, You Lost!\n";//Mock Winner


    //Output Correct Answer To The Game
    std::cout<<"\nCorrect Answer\n";
    circle(key);//Output Correct Answer To The Game


    //A Sebastian Production
    return 0;
}



//Function Definitions
void gmeScrn(std::string r1 [],std::string r2 [],std::string r3 [],
             std::string r4 [],std::string r5 [],std::string r6 [],
             std::string r7 [],std::string r8 [],std::string key [],
             short rghtPos,short rghtClr){

    //Output All The Rows In The Game
    std::cout<<"\n\n\n\tMastermind\n";//Game Name Above The Table
    std::cout<<"Row 8 \n";//Row 8
    circle(r8);
    std::cout<<"Row 7 \n";//Row 7
    circle(r7);
```

```
    std::cout<<"Row 6 \n";//Row 6
    circle(r6);
    std::cout<<"Row 5 \n";//Row 5
    circle(r5);
    std::cout<<"Row 4 \n";//Row 4
    circle(r4);
    std::cout<<"Row 3 \n";//Row 3
    circle(r3);
    std::cout<<"Row 2 \n";//Row 2
    circle(r2);
    std::cout<<"Row 1 \n";//Row 1
    circle(r1);

    //Outputs The Number Of Correct Guesses
    std::cout<<"Correct Color, But Wrong Position: "<<rghtClr<<std::endl
            <<"Correct Color In Correct Position: "<<rghtPos<<std::endl;
    std::cout<<"Press Enter To Continue\n\n\n";//Prompts User To Press Enter
    std::cin.ignore();//Pauses Until Enter Is Pressed
    std::cin.ignore();//Pauses Until Enter Is Pressed
}



void circle(std::string clr []){
    std::cout<<"   ____     ____     ____     ____   \n"
             "  /    \\   /    \\   /    \\   /    \\ \n"
             " /      \\ /      \\ /      \\ /      \\\n"
             " | "<<clr[0]<<"  | | "<<clr[1]<<"  | | "<<
                  clr[2]<<"  | | "<<clr[3]<<"  | \n"
             " \\      / \\      / \\      / \\      / \n"
             "  \\____/   \\____/   \\____/   \\____/  \n";
}



void round(std::string row [],std::string key [],short &rghtClr,short &rghtPos){
    //Declare And Initialize Variables
    short choice=0;//Menu Choice
    const short COLS=4;//The Number Of Columns In The Game, Constant
    bool error=false;
```

```
//Loop For All The Color Choices
for(int i=0;i<COLS;i++){

   //Get Color Choice From User With Exception Loop
   do{
      try{
         //Reset Error Flag
         error=false;

         //Output Color Choice Menu
         std::cout<<"\n\n\nColor Choices:\n\n"
                  "1.) Red\n"
                  "2.) Blue\n"
                  "3.) Yellow\n"
                  "4.) Green\n"
                  "5.) Purple\n"
                  "6.) Orange"<<std::endl<<
                  "Enter Your Choice For Circle #"<<i+1<<": ";

         //Get User Input
         std::cin>>choice;//Get Color Choice From User

         //Attempt To Catch And Fix Buffer Error
         if(std::cin.fail()||choice<1||choice>6){//If Char Or Bad Range
            std::cin.clear();//Clear Any Error Flags From cin
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(),'\n');
            error=true;//Set Error Flag To Continue Looping
            throw 0;//Throw Exception With Dummy Variable
         }
      }
      catch(int x){//Catch Exception
         std::cout<<std::endl<<std::endl<<"It Appears You"//Output Error
               " Entered Something That You Should Not Have";//Message
      }
   }while(error==true);//Continue Looping While Error State Is True

   //Enter Value Into Array For Row
   std::cout<<std::endl<<std::endl<<std::endl<<std::endl;//Move Down 4 Lines
   switch(choice){//Evaluates Color Choice
      case 1:row[i]="Red";break;//Adds Chosen Color To Array
      case 2:row[i]="Blu";break;//Adds Chosen Color To Array
      case 3:row[i]="Yel";break;//Adds Chosen Color To Array
```

```
            case 4:row[i]="Grn";break;//Adds Chosen Color To Array
            case 5:row[i]="Prp";break;//Adds Chosen Color To Array
            case 6:row[i]="Org";break;//Adds Chosen Color To Array
            default :std::cout<<"Something Happened\n";//Lets Me Know
        }
    }
    //Update Correctness Of Color And Position After Each Turn
    chckAns(key,row,rghtClr,rghtPos);
}




void setWin(std::string key []){
    //Set Random Seed
    srand(static_cast<unsigned>(time(0)));

    //Declare And Initialize Variables
    short color=0;//Color To Be Added To Row Array
    short const COLS=4;//The Number Of Columns In The Game, Constant

    //Fill Each Element
    for(int i=0;i<COLS;i++){
        color=rand()%6;//Random Value Range [0-5] For Each Color Option

        //Use Switch Case To Choose Color For Key
        switch(color){//Evaluates Random Color Choice With Switch
            case 0:key[i]="Red";break;//Adds Random Color To Key
            case 1:key[i]="Blu";break;//Adds Random Color To Key
            case 2:key[i]="Yel";break;//Adds Random Color To Key
            case 3:key[i]="Grn";break;//Adds Random Color To Key
            case 4:key[i]="Prp";break;//Adds Random Color To Key
            case 5:key[i]="Org";break;//Adds Random Color To Key
        }
    }
}


void chckAns(std::string key [],std::string row [],short &rghtClr,short &rghtPos)   {
    //Declare And Initialize Variables
    enum Colors {RED,BLU,YEL,GRN,PRP,ORG};//Enumerator For Colors
    short kColor [6]={0,0,0,0,0,0},//Holds The Active Number Of Colors In Key
```

```
    rColor [6]={0,0,0,0,0,0};//Holds The Active Number Of Colors In Current Row
short const COLS=4;//The Number Of Columns In The Game, Constant
rghtClr=0;//Reset Value For Each Round
rghtPos=0;//Reset Value For Each Round

//Set All Color Values Into Arrays And Right Color/Pos Variables
for(int i=0;i<COLS;i++){
   //Fill Right Pos Variable
   if(row[i]==key[i])//If The Value In Key Is Equivalent To Row
      rghtPos++;//Right Position Is Incremented

   //Count The Number Of Color Each In The Key Array
   if(key[i]=="Red")//If The Color Is Red
      kColor[RED]++;//Increment The Count Of That Color
   else if(key[i]=="Blu")//If The Color Is Blue
      kColor[BLU]++;//Increment The Count Of That Color
   else if(key[i]=="Yel")//If The Color Is Yellow
      kColor[YEL]++;//Increment The Count Of That Color
   else if(key[i]=="Grn")//If The Color Is Green
      kColor[GRN]++;//Increment The Count Of That Color
   else if(key[i]=="Prp")//If The Color Is Purple
      kColor[PRP]++;//Increment The Count Of That Color
   else if(key[i]=="Org")//If The Color Is Orange
      kColor[ORG]++;//Increment The Count Of That Color

   //Count The Number Of Color Each In The Row Array
   if(row[i]=="Red")//If The Color Is Red
      rColor[RED]++;//Increment The Count Of That Color
   else if(row[i]=="Blu")//If The Color Is Blue
      rColor[BLU]++;//Increment The Count Of That Color
   else if(row[i]=="Yel")//If The Color Is Yellow
      rColor[YEL]++;//Increment The Count Of That Color
   else if(row[i]=="Grn")//If The Color Is Green
      rColor[GRN]++;//Increment The Count Of That Color
   else if(row[i]=="Prp")//If The Color Is Purple
      rColor[PRP]++;//Increment The Count Of That Color
   else if(row[i]=="Org")//If The Color Is Orange
      rColor[ORG]++;//Increment The Count Of That Color
}

//Calculate The Number Of Correct Colors
for(int i=0;i<6;i++)//For Each Of The Six Color Possibilities
```

```
      if(kColor[i]>0&&rColor[i]>0){//If Both The Row & Key Have Common Color
         if(kColor[i]>rColor[i])//If Key Has More Color Than Row
            rghtClr+=rColor[i];//Add Rows Color Value To Right Color
         else                  //If Row Has More Than The Key
            rghtClr+=kColor[i];//Add The Keys Value On
   }

   //Adjust For Correct Color, But Wrong Position
   rghtClr-=rghtPos;//Subtract The Number In Right Position From Right Color
}

void wchRow(std::string r1 [],std::string r2 [],std::string r3 [],
         std::string r4 [],std::string r5 [],std::string r6 [],
         std::string r7 [],std::string r8 [],std::string key [],
         short &rghtPos,short &rghtClr,short index){
   //Use The Right Row For The Right Rounds
   switch(index){
      case 0 :round(r1,key,rghtClr,rghtPos);break;//Use r1 For Round 1
      case 1 :round(r2,key,rghtClr,rghtPos);break;//Use r2 For Round 2
      case 2 :round(r3,key,rghtClr,rghtPos);break;//Use r3 For Round 3
      case 3 :round(r4,key,rghtClr,rghtPos);break;//Use r4 For Round 4
      case 4 :round(r5,key,rghtClr,rghtPos);break;//Use r5 For Round 5
      case 5 :round(r6,key,rghtClr,rghtPos);break;//Use r6 For Round 6
      case 6 :round(r7,key,rghtClr,rghtPos);break;//Use r7 For Round 7
      case 7 :round(r8,key,rghtClr,rghtPos);break;//Use r8 For Round 8
      default : std::cout<<"Something Went Wrong In wchRow\n";
   }
}
```

End C++ Code

Pseudocode

```
int main(){
        declare constant COLS and assign it 4
        declare string r1
        declare string r2
        declare string r3
        declare string r4
        declare string r5
        declare string r6
        declare string r7
        declare string r8
        declare short rghtClr and set to 0
        declare short rghtPos and set to 0
        declare short choice and set to 0
        declare short index and set to 0
        declare short rounds and set to 8
        declare bool gameWin and set to false
        declare bool error and set to false
        call setWin function

        output title
        allow user to press enter to start the game
        do
                do
                        try
                                set error flag to false
                                output the menu choices
                                get user input
                                if user enters bad data
                                        clear keyboard error flags
                                        empty keyboard buffer
                                        throw an exception
```

```
                    catch
                        output the error message
                        set error flag to true
                    switch
                        case 1 call gmeScrn function
                        case 2 call wchRow function
                        case 3 tell user goodbye and set condition to exit
                while choice is the first one or the error flag is set
                if rghtPos value equals 4 and indicates the game is won
                increment the index value
        while the game is not over or user did not choose to quit
        output the current round
        if the game was won
                congratulate the winner
        else
                congratulate the loser
        output correct answer
        call circle function with key string array
}

void gmeScrn( string [] x 9, short x 2){
        output game title mastermind
        output row 8
        call function circle with r8
        output row 7
        call function circle with r7
        output row 6
        call function circle with r6
        output row 5
        call function circle with r5
        output row 4
        call function circle with r4
        output row 3
```

```
        call function circle with r3
        output row 2
        call function circle with r2
        output row 1
        call function circle with r1
        output correct number of colors guessed
        output correct number in right position
        allow user to press enter to continue
        pause until user presses enter to continue
}

void circle(string []){
        output all elements in the string inside of string literal circles
}

void round (string [] x 2, short & x 2){
        declare short choice and set to 0
        declare constant COLS and set to 4
        delcare bool error and set to false
        loop for the index i=0 until I is greater than COLS
                do
                        try
                                set error flag to false
                                output all of the menu options
                                get user input
                                if the user messed up the input
                                        clear input messup flag
                                        clear keyboard buffer
                                        set error flag to true
                                        throw an exception
                        catch
                                output an error message
                while the error flag is set to true
```

```
            output 4 newlines to move screen down
            switch for the choice option
                case 1 set row[i] to Red and break from switch
                case 2 set row[i] to Blu and break from switch
                case 3 set row[i] to Yel and break from switch
                case 4 set row[i] to Grn and break from switch
                case 5 set row[i] to Prp and break from switch
                case 6 set row[i] to Org and break from switch
        call function chckAns to check the validity of the users answer
}

void setWin (string []){
        set random seed
        declare short color and set to 0
        declare constant COLS and set to 4
        for int i=0 until I is greater than COLS
            assign the color a random value from range 0-5
            switch case of color
                case 0 set key[i] to Red and break from switch
                case 1 set key[i] to Blu and break from switch
                case 2 set key[i] to Yel and break from switch
                case 3 set key[i] to Grn and break from switch
                case 4 set key[i] to Prp and break from switch
                case 5 set row[i] to Org and break from switch
}

void chckAns(string [] x 2, short & x 2){
        declare enum colors with RED,BLU,YEL,GRN,PRP,ORG
        declare short array kColor and assign it to 0
        declare short array rColor and assign it to 0
        declare const COLS and assign it to 0
        set rghtPos to 0
        set rghtClr to 0
```

```
for int i=0 until I is greater than COLS
        if the value in the row array is the same as in the answer array
                increment the rghtPos value
        if key[i] is equivalent to Red
                increment kColor[RED]
        else if key[i] is equivalent to Blu
                increment kColor[BLU]
        else if key[i] is equivalent to Yel
                increment kColor[YEL]
        else if key[i] is equivalent to GRN
                increment kColor[GRN]
        else if key[i] is equivalent to Prp
                increment kColor[PRP]
        else if key[i] is equivalent to Org
                increment kColor[ORG]
        if row[i] is equivalent to Red
                increment rColor[RED]
        else if row[i] is equivalent to Blu
                increment rColor[BLU]
        else if row[i] is equivalent to Yel
                increment rColor[YEL]
        else if row[i] is equivalent to GRN
                increment rColor[GRN]
        else if row[i] is equivalent to Prp
                increment rColor[PRP]
        else if row[i] is equivalent to Org
                increment rColor[ORG]
for int i=0 until I is greater than 6
        if the value in kColor[i] and rColor[i] are both not 0
                if the value in kColor[i] is greater than rColor[i]
                        assign rghtClr the value in rColor[i]
                else
                        assign rghtClr the value in kColor[i]
```

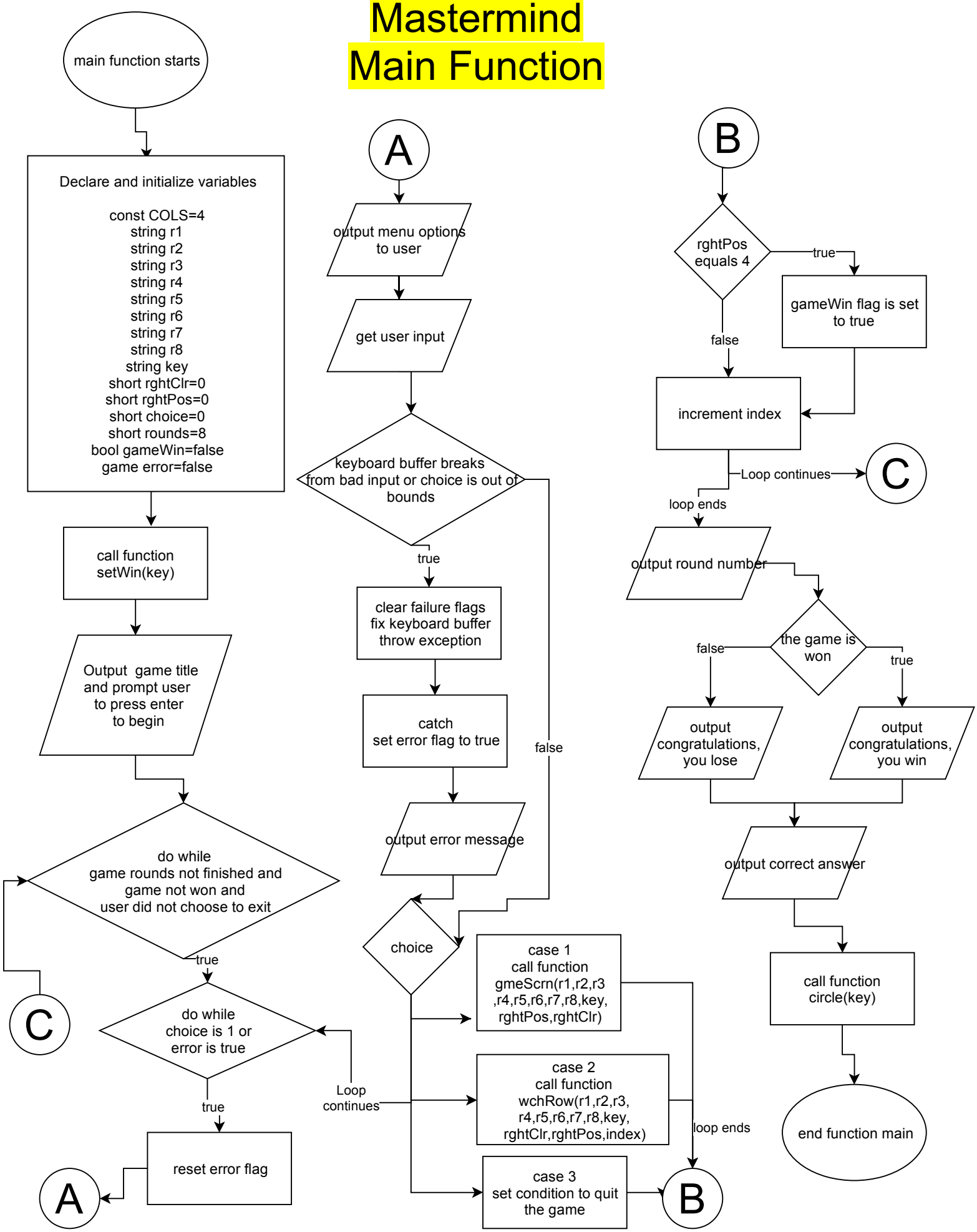
```
        subtract rghtPos from rghtClr
}

void wchRow(string [] x 9, short & x 2, short){
        switch for index short
                case 0 call function round with r1, key, rghtClr and rghtPos
                case 1 call function round with r2, key, rghtClr and rghtPos
                case 2 call function round with r3, key, rghtClr and rghtPos
                case 3 call function round with r4, key, rghtClr and rghtPos
                case 4 call function round with r5, key, rghtClr and rghtPos
                case 5 call function round with r6, key, rghtClr and rghtPos
                case 6 call function round with r7, key, rghtClr and rghtPos
                case 7 call function round with r8, key, rghtClr and rghtPos
                default output that something has gone horribly wrong
}
```

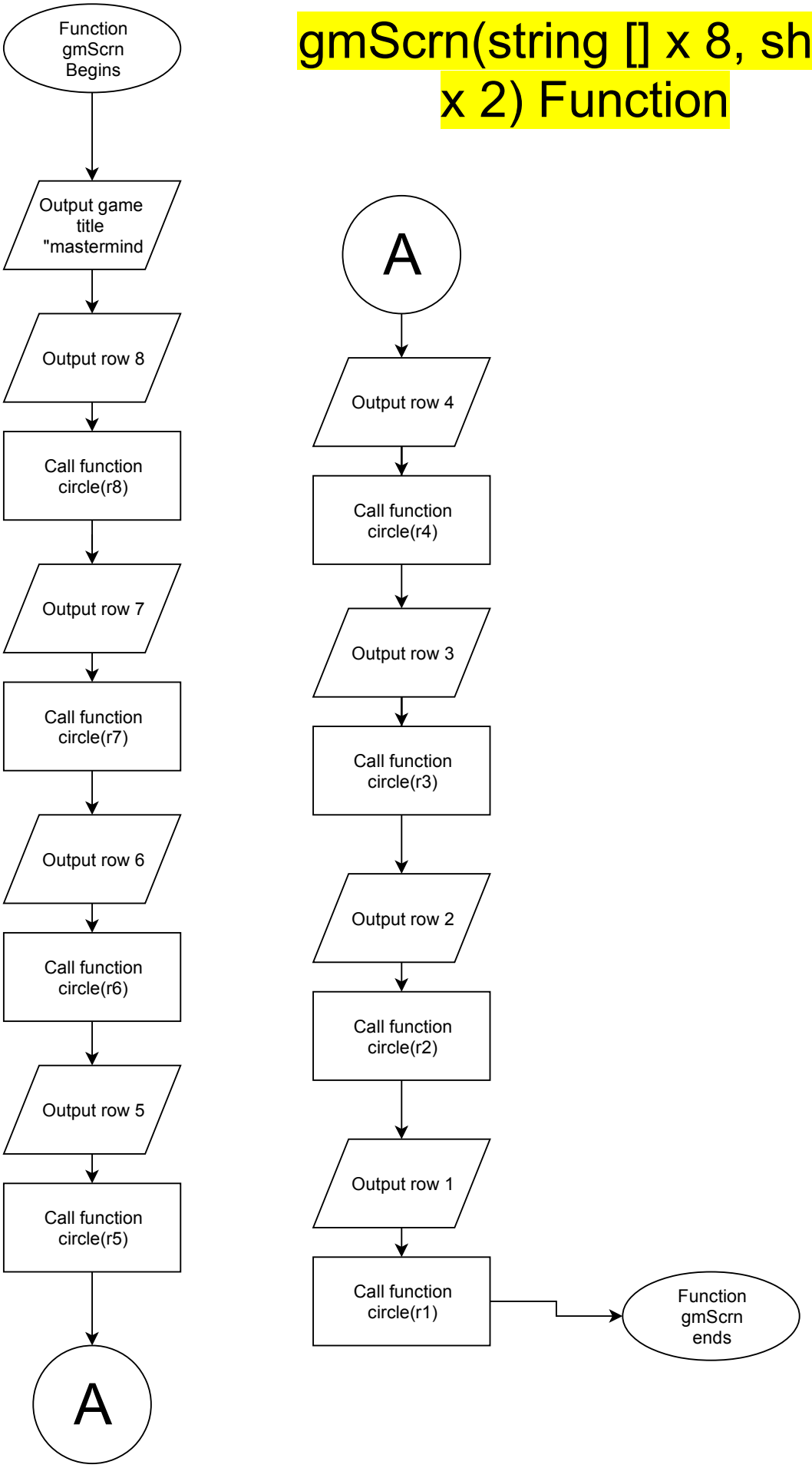
End Pseudocode

Program Photos

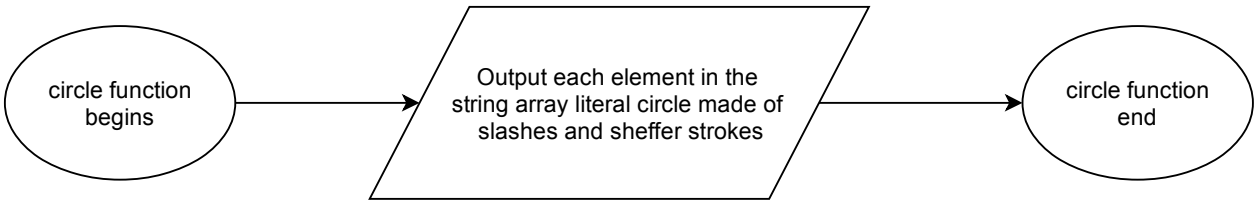
Mastermind
Main Function



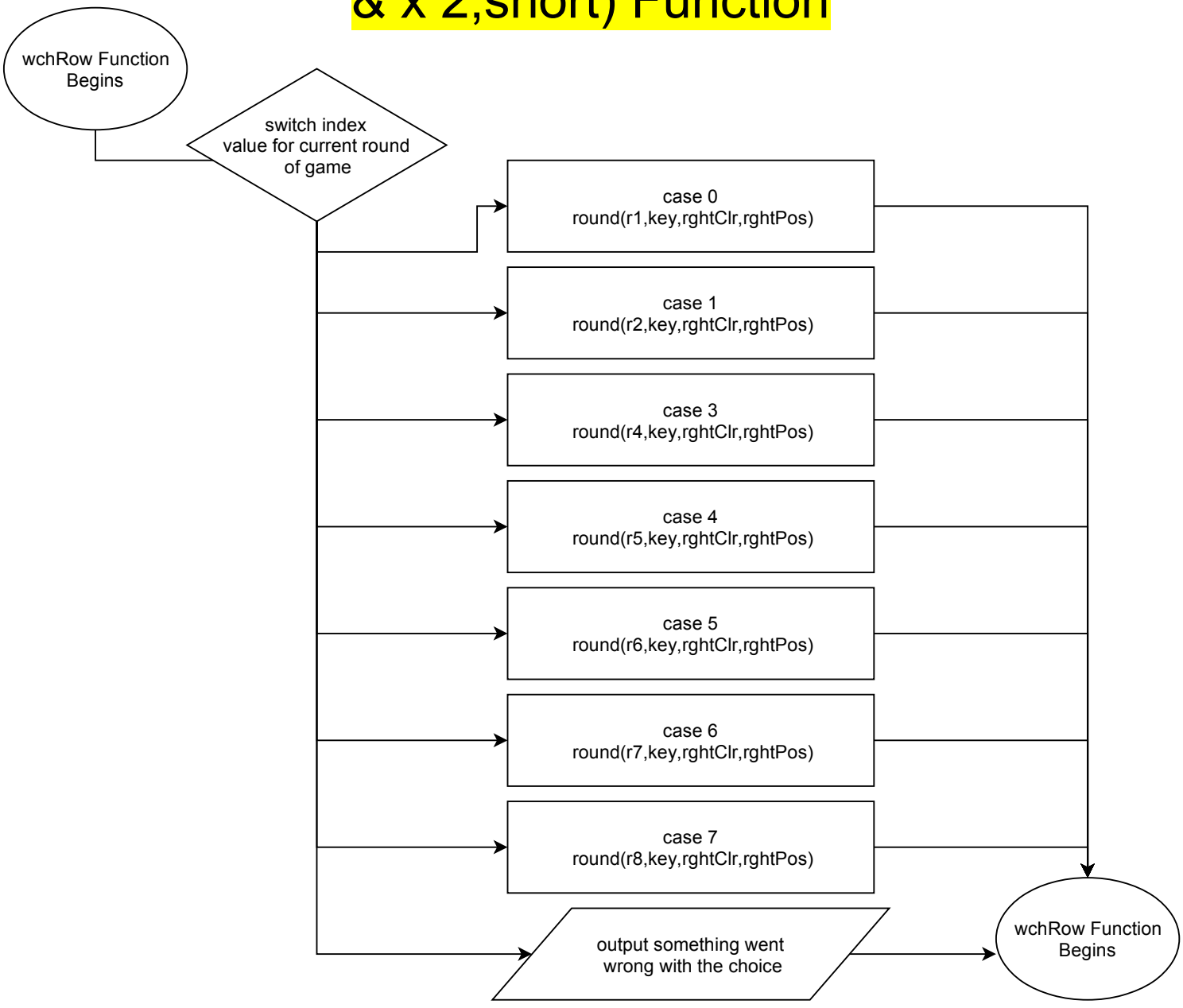
gmScrn(string [] x 8, short x 2) Function



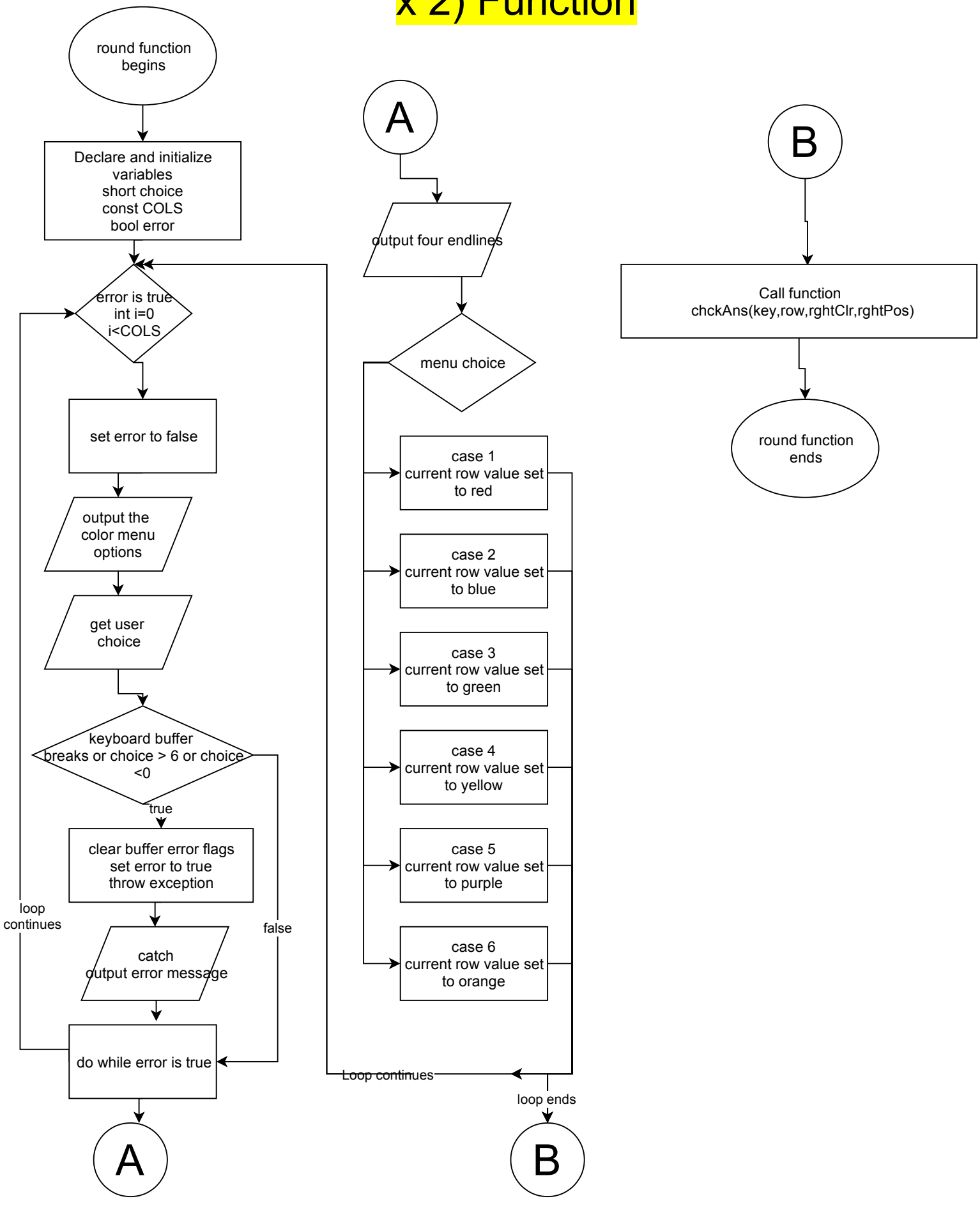
circle(string []) Function



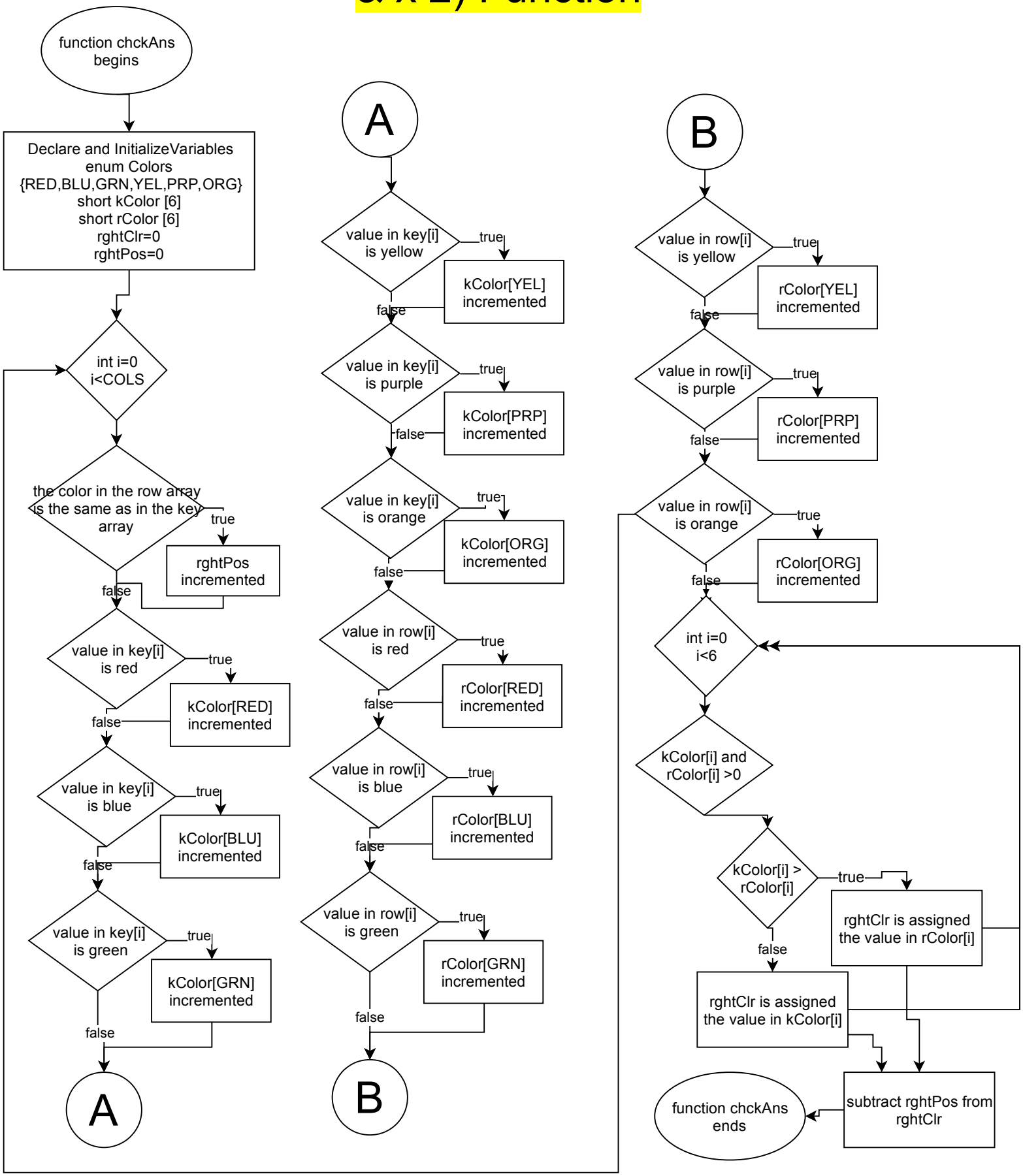
wchRow(string [] x 9, short & x 2,short) Function



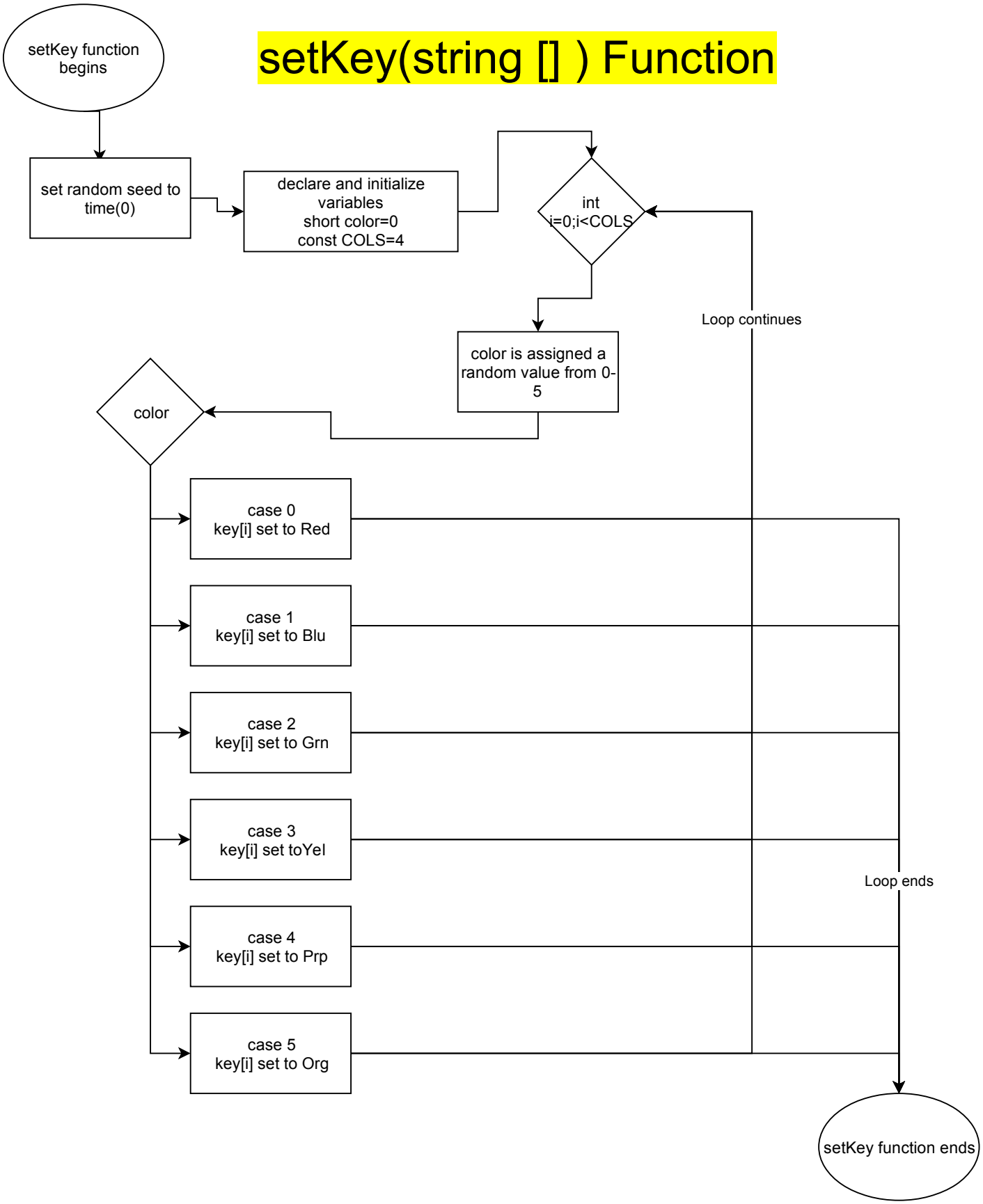
round(string [] x 2, short & x 2) Function

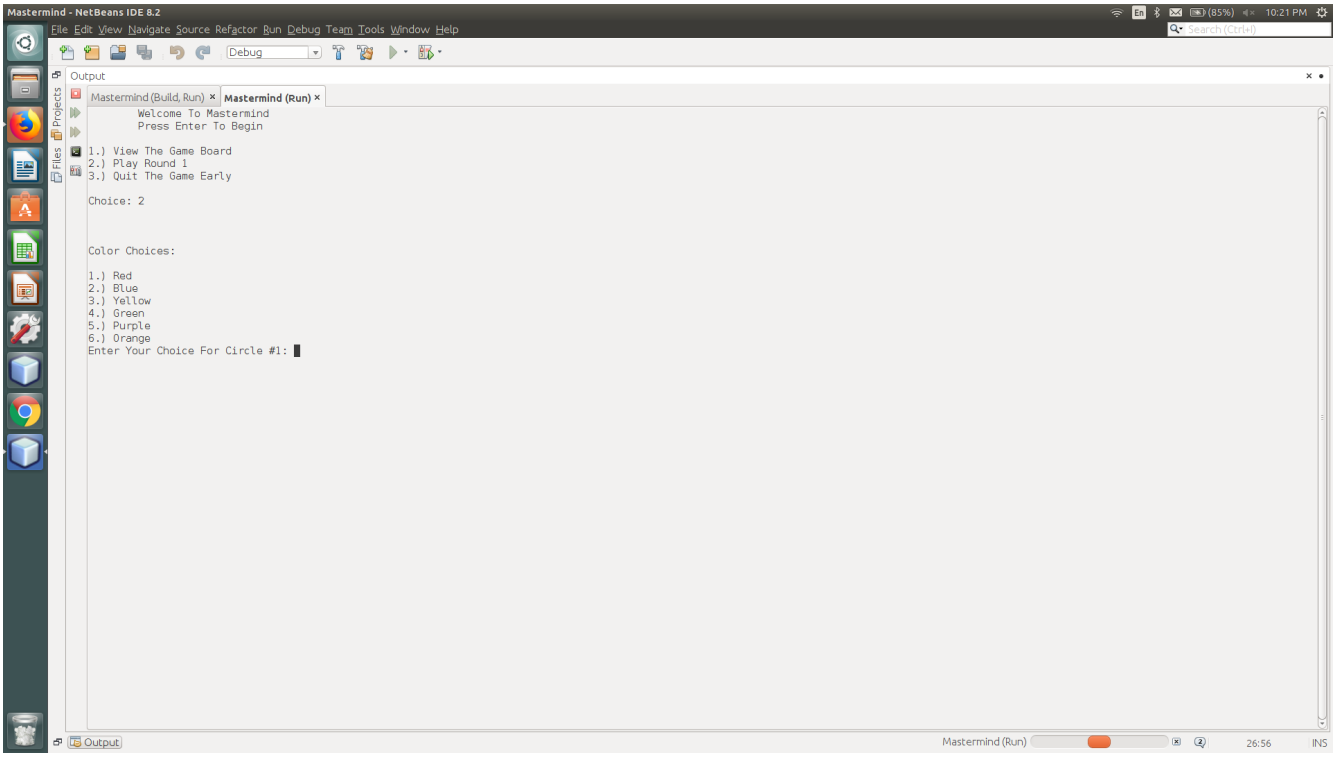


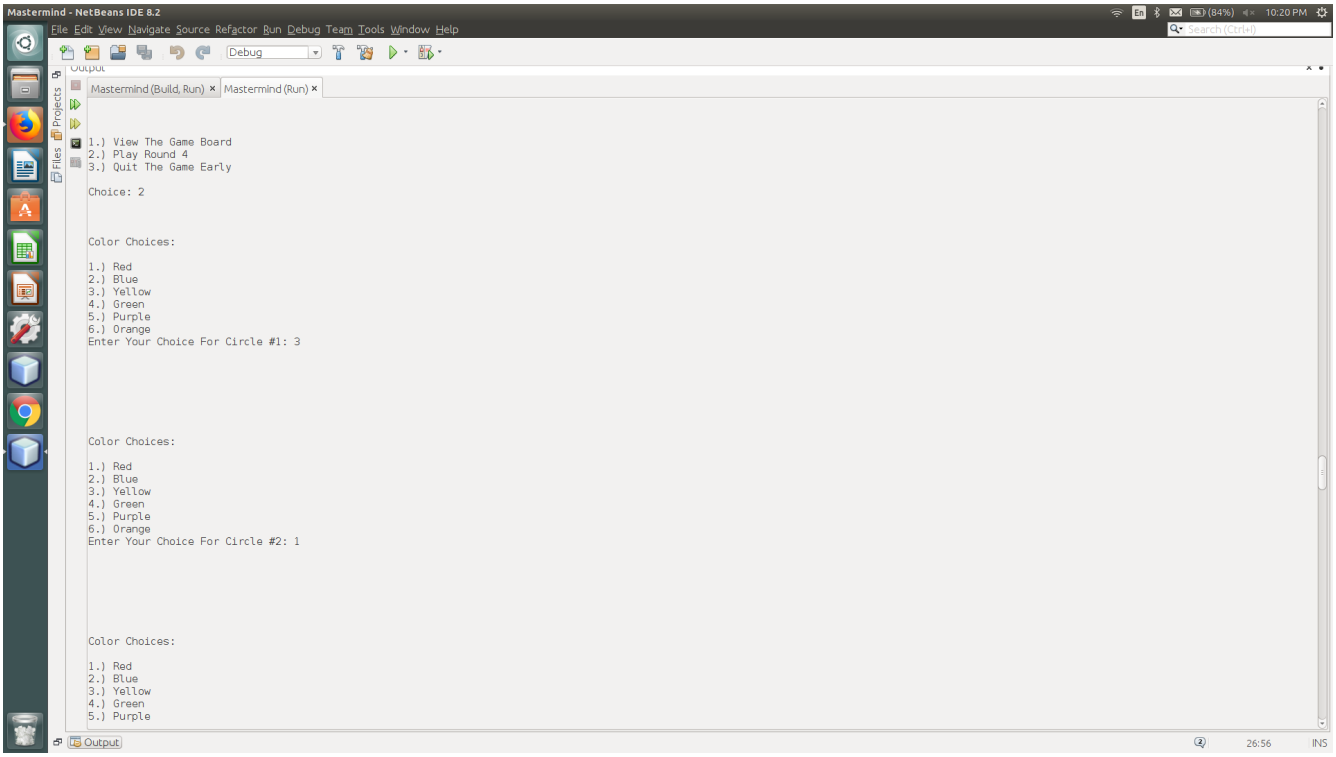
chckAns(string [] x 2, short & x 2) Function

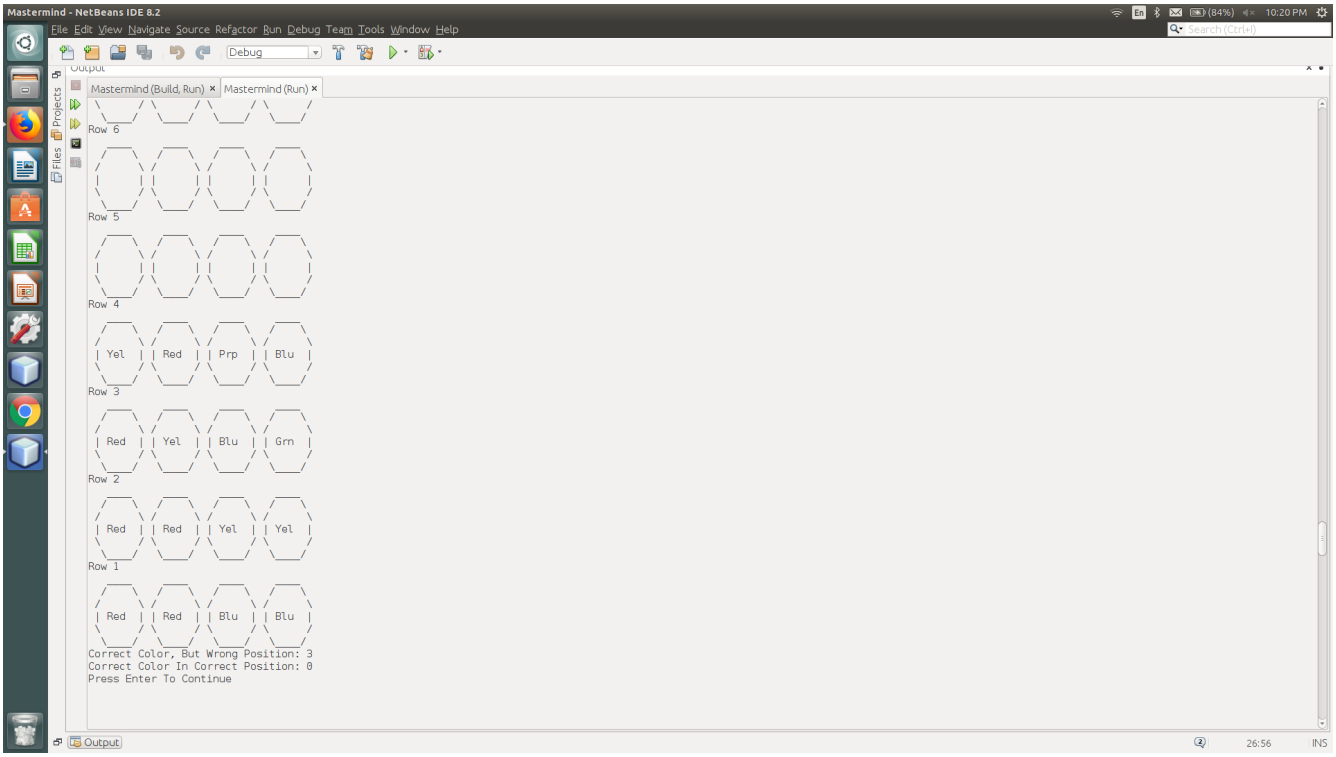


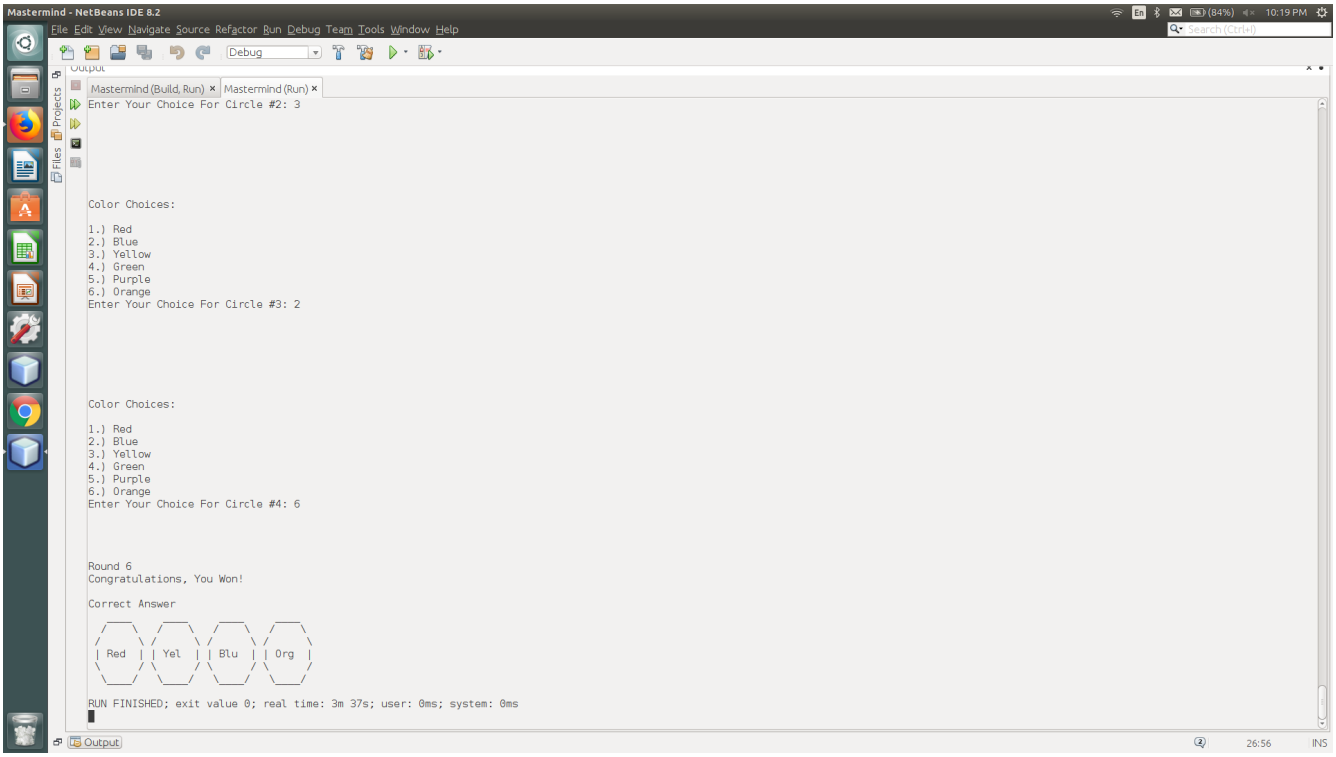
setKey(string []) Function




Mastermind - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
Debug
Output
Mastermind (Build, Run)
Mastermind (Run)
Welcome To Mastermind
Press Enter To Begin
1.) View The Game Board
2.) Play Round 1
3.) Quit The Game Early
Choice: 2
Color Choices:
1.) Red
2.) Blue
3.) Yellow
4.) Green
5.) Purple
6.) Orange
Enter Your Choice For Circle #1:
Output
Mastermind (Run)
26:56
INS


Mastermind - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
Debug
Mastermind (Build, Run)
Mastermind (Run)
1.) View The Game Board
2.) Play Round 4
3.) Quit The Game Early
Choice: 2
Color Choices:
1.) Red
2.) Blue
3.) Yellow
4.) Green
5.) Purple
6.) Orange
Enter Your Choice For Circle #1: 3
Color Choices:
1.) Red
2.) Blue
3.) Yellow
4.) Green
5.) Purple
6.) Orange
Enter Your Choice For Circle #2: 1
Color Choices:
1.) Red
2.) Blue
3.) Yellow
4.) Green
5.) Purple
Output
26:56
INS


Mastermind - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
Debug
Mastermind (Build, Run)
Mastermind (Run)
Row 6
Row 5
Row 4
Row 3
Row 2
Row 1
Correct Color, But Wrong Position: 3
Correct Color In Correct Position: 0
Press Enter To Continue
Output
26:56
INS


Mastermind - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
Debug
Mastermind (Build, Run)
Mastermind (Run)
Enter Your Choice For Circle #2: 3
Color Choices:
1.) Red
2.) Blue
3.) Yellow
4.) Green
5.) Purple
6.) Orange
Enter Your Choice For Circle #3: 2
Color Choices:
1.) Red
2.) Blue
3.) Yellow
4.) Green
5.) Purple
6.) Orange
Enter Your Choice For Circle #4: 6
Round 6
Congratulations, You Won!
Correct Answer
RUN FINISHED; exit value 0; real time: 3m 37s; user: 0ms; system: 0ms
Output
26:56
INS