„Du hattest das Lightsout Programm gewünscht. Ich habe sogar meine Dokumentation gefunden, die den Lösungsansatz nochmal detailliert erklärt. Das ist eine "leichte" Junioraufgabe von vor ein paar Jahren, ich habe auch bei Bedarf kompliziertere Aufgaben. Hier vielleicht ein paar Infos: Der Wettbewerb heißt BWINF, bei dem es eine Reihe an solcher Aufgaben gibt, die man dokumentiert lösen muss. Gerade für Schüler, die noch nicht in der Oberstufe sind, gibt es die zwei leichteren  Junioraufgaben. Man kann als "Frühstarter" aber nur einen Preis bekommen und nicht weiterkommen. Es gibt neben dem Bundeswettbewerb auch einen Jugendwettbewerb Informatik sowie den Informatik Biber.

Der Quelltext ist in der Dokumentation enthalten. Ich habe leider die Finale Dokumentation nicht gefunden, aber die hier sollte abgesehen von Rechtschreibfehlern, inhaltlich korrekt sein. Vielleicht schaust Du nochmal darüber. „ Tara

Inhalt

[Aufgabe 2](#_Toc468122289)

[Lösungsidee 2](#_Toc468122290)

[Variation 2: 4](#_Toc468122291)

[Variation 3: 4](#_Toc468122292)

[Umsetzung 4](#_Toc468122293)

[Variante 2: 5](#_Toc468122294)

[Variant 3: 5](#_Toc468122295)

[Beispiele 5](#_Toc468122296)

[Variante 1: 5](#_Toc468122297)

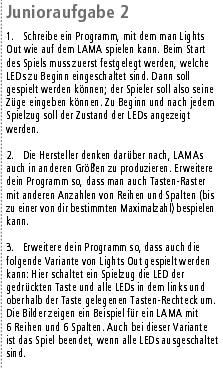
[Variante 2: 6](#_Toc468122298)

[Quellcode 6](#_Toc468122299)

[Variante 1: 6](#_Toc468122300)

[Variante 2: 8](#_Toc468122301)

# Aufgabe



# Lösungsidee

Es gibt ein 5 mal 5 Feld mit leutchtenden Knöpfen. Entweder sie sind an oder aus. Die Aufgabe besteht darin dieses Spiel „lightsout“ zu simulieren.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |

Ich möchte für den Anfang ein Array erstellen, was 2d ist und es soll intitialisiert werden, d.h. alle Felder müssen entweder mit 0 oder 1 gefüllt werden. Dies erfolgt per Zufall und ändert sich bei jedem Neustart (siehe oben).

Nun brauche ich die Eingabe des Nutzers/Spielers. Er gibt mir die Koordinaten (Reihe und Spalte). Nachdem ich diese habe, müssen alle Nachbarlichter gewechselt werden. Z.B. Koordinate: 2,2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |

Um das zu erreichen, muss man mithilfe der Koordinaten die Lichter wie folgt wechseln:

Eingabe: Reihe, Spalte z. B. 4,5

Dann werden diese Lichter gewechselt:

(Reihe, Spalte)

(Reihe-1, Spalte)

(Reihe+1, Spalte)

(Reihe, Spalte-1)

(Reihe, Spalte+1)

Schließlich müssen wir nur noch überprüfen, ob der Nutzer/Spieler Koordinaten angibt die überhaupt existieren, also nicht z.B. (a,3) oder (8,239) oder (Hallo, Tschüss)

## Variation 2:

Um das Spielfeld erweitern zu können, muss man die Angaben mit „5“, die wir zu Begin festgelegt haben mit der Eingabe des Nutzers/Spielers substituieren.

## Variation 3:

Für diese Variante muss nur das Wechseln der Lichter geändert werden. Alle Koordinaten der Lichter die kleiner oder gleich der eingegebenen Koordinaten sind, werden demnach an oder aus geschaltet.

# Umsetzung

Wie in der Lösunngsidee beschreiben brauchen wir ein 2d-Array. Dafür muss man ein Array in ein Array einbauen.

Array2d = [ [0]\*5 for i in range(0, 5)]

Nun müssen die Felder initialisiert werden mit 0 und 1. Das erfolgt per Zufall.

Jetzt werden die Koordinaten vom Spieler gefragt. Nachdem er sie eingibt muss überprüft werden ob es überhaupt eine Zahl ist die in diesem Feld existiert oder ein Buchstabe, etc. Mit ‘if not variable.isdigit‘ kann man überprüfen ob es irgendetwas ist, außer eine Zahl.

Anschließend führen wir den Spielzug aus:

Wir haben Spalte= sp

Und Zeile= zl

All diese Koordinaten der Lichter müssen gewechselt werden.

(sp, zl)

(sp-1, zl)

(sp+1, zl)

(sp, zl-1)

(sp, zl+1)

Wenn der Spieler es geschafft hat, alle Lichter auszuschalten, endet das Programm.

## Variante 2:

Alles was die „5“ beinhaltet muss mit der Eingabe des Spielers ersetzt werden, wie z. B:

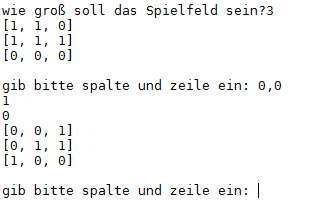
Array2d = [ [0]\*eingabe for i in range(0, eingabe)]

## Variante 3:

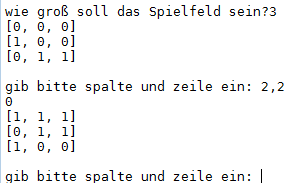
Anstatt die Nebenliegenden Lichter umzuschalten werden hierbei alle Lichter mit den Koordinaten die gleich oder kleiner sind umgeschaltet.

Beispiele

# Variante 1:



# Variante 2:



# Quellcode

## Variante 1:

import random

def matrixprint (lightout):

for i in range (0,len(lightout)):

print (lightout[i][:])

def eingabe (size):

sp, zl= input("gib bitte spalte und zeile ein: ").split(','or ' ' or '.' )

if not zl.isdigit() or not sp.isdigit() or (int(zl)>size-1 or int(zl)<0 or int(sp)<0 or int(sp)>size-1):

return eingabe(size)

else:

return int(sp), int(zl)

def spielzug (sp, zl, size):

if zl>size-1 or sp>size-1:

return None

if lightout[sp][zl]==1:

lightout[sp][zl]= lightout[sp][zl]=0

elif lightout[sp][zl]==0:

lightout[sp][zl]= lightout[sp][zl]=1

return None

###########################################################################

############################# start vom Programm ##########################

###########################################################################

size= int(input("wie groß soll das Spielfeld sein?"))

lightout = [ [0]\*size for i in range(0,size)]

for i in range (0,size):

for j in range(0,size):

lightout[i][j] =random.randint(0,1)

matrixprint(lightout)

while sum(sum(lightout,[]))>0:

zl, sp= eingabe(size)

#print ("deine koordinaten", zl, sp)

print(lightout[sp][zl] )

spielzug(sp, zl, size)

spielzug(sp-1,zl, size)

spielzug(sp+1,zl, size)

spielzug(sp,zl-1, size)

spielzug(sp,zl+1, size)

print(lightout[sp][zl] )

matrixprint(lightout)

print("toll, du hast gewonnen!")

## Variante 2:

import random

def matrixprint (lightout):

for i in range (0,len(lightout)):

print (lightout[i][:])

def eingabe (size):

sp, zl= input("gib bitte spalte und zeile ein: ").split(','or ' ' or '.' )

if not zl.isdigit() or not sp.isdigit() or (int(zl)>size-1 or int(zl)<0 or int(sp)<0 or int(sp)>size-1):

return eingabe(size)

else:

return int(sp), int(zl)

def spielzug (sp, zl, size):

if zl>size-1 or sp>size-1:

return None

if lightout[sp][zl]==1:

lightout[sp][zl]= lightout[sp][zl]=0

elif lightout[sp][zl]==0:

lightout[sp][zl]= lightout[sp][zl]=1

return None

###########################################################################

############################# start vom Programm ##########################

###########################################################################

size= int(input("wie groß soll das Spielfeld sein?"))

lightout = [ [0]\*size for i in range(0,size)]

for i in range (0,size):

for j in range(0,size):

lightout[i][j] =random.randint(0,1)

matrixprint(lightout)

while sum(sum(lightout,[]))>0:

zl, sp= eingabe(size)

#print ("deine koordinaten", zl, sp)

for i in range (0, sp+1):

for j in range (0, zl+1):

spielzug(i, j, size)

print(lightout[sp][zl] )

matrixprint(lightout)

print("toll, du hast gewonnen!")