Verkefni 1.1

1.

# búa til public static int með (int n)

# gera integer fyrir tölu a og b

# for loopa sem fer endalaust með n

# láta int temp = a

# gera a = b

# láta b = temp + b

# loka svo for loopu

# returna a

# loka public static int

# Búa til static void

# Búa til for loopu sem mun skrifa niður töluna sem kemur frá public static int og mun keyra sérstakt oft sinnum

# loka svo for loopu

# loka static void

2.

#forritið mun lesta listan og sjá hvaða hlutir það eru sem það þarf

#fara að hverjum gang en mun ekki fara inn í þan gáng nema það passar við eitthvern hlut á listanum

Verkefni 1.2

Verkefni 1.3

a.Því það er 2 bara við hliðiná livandi kössunum breytast þeir ekki



b. mynstur sem skilar sér í upphaflegu ástandi, í sömu stefnu og stöðu, eftir endanlegt fjölda kynslóða.



c. það kemur aftur eftir ákveðinn fjölda kynslóða í sömu stefnu en í mismunandi stöðu.



d. Það byrjar í einu minstry og svo endar í endalausu loopi eftir sérstakan tíma

 

Verkefni 1.4

* Log2 er hægt að nota til að sýna alla möguleikana sem geta komið , til dæmis ef það eru 16 möguleikar þá er 2\*2\*2\*2 = 16 þannig ef 4 sinnum er skift í tvennt þá færðu það sem leitað er af.
* Það er hægt að nota Decision tree til að nota morse code, stutt til að fara til vinstri og löng til að fara til hægri



Verkefni 1.5

Fann það að FFF0F0F0 náði að minka bytes um meira en helming en enþá lítur eins út, fór alveg niður í 6000 bytes

Verkefni 1.6

#skrifað er niður hversu margar hliðar og fyrir hverja hlið mun stytta hversu mikið mun snúa með því að deila það með 360, og svo líka hversu oft á að snúa

kóði:

pen purple

sides = 5

for [1..sides]

rt 360/sides

fd 100

Verkefni 1.7

A)Alan Mathison Turing var enskur tölvunarfræðingur , stærðfræðingur, logician , cryptanalyst , heimspekingur, og fræðileg líffræðingur . Turing var mjög áhrifamikill í þróun fræðilegrar tölvunarfræði og veitti formlega hugmyndina um reiknirit og útreikning með Turing vélinni , sem er talin líkan af almennum tölvu . Turing er víða talinn vera faðir fræðilegrar tölvunarfræði og gervigreind . Turing vél er stærðfræðileg líkan af útreikningi sem skilgreinir abstrakt vél , sem vinnur með táknum á ræma borði samkvæmt reglustafli. Þrátt fyrir einfaldleika líkansins, gefið hvaða tölvu reiknirit , Turing vél sem er fær um að líkja eftir því að rökfræði er hægt að smíða.

B) vélræn nám er svið í tölvunarfræði sem gefur tölvum kleift að læra án þess að vera sérstaklega forritaðar.

C)Í tölvunarfræði hefur hermun nokkur sérhæfð merkingu: Alan Turing notaði hugtakið "hermun" til að vísa til hvað gerist þegar alhliða vél framkvæmir ástandsskiptatöflu (í nútíma hugtökum, tölva keyrir forrit) sem lýsir ástandsskiptum, inntakum og framleiðsla af efni sem er stakur-ástand vél. Tölvan hermir við efnið. Samkvæmt því, í fræðilegri tölvunarfræði er hugtakið eftirlíking tengsl milli umskiptakerfa ríkisins , gagnleg í rannsókn á merkingarfræði .

Verkefni 1.8

square(a)→a2:

# tala a og b verða það sama því tala í öðru veldi er bara hún margfölduð með sér sjálfri

a = b

def multiply(a, b):

if b == 1:

return a

if b > 1:

return a + multiply(a, b-1)

log(a)→log10a:

Verkefni 1.9

2n−1 fyrir að sjá hversu mörg skref það tekur til að klára

* Move a disk to or from towers A and C
* Move a disk to or from towers A and B
* Move a disk to or from towers B and C
* Repeat this process until the puzzle is solved

public class TowersOfHanoi

{

public static void Main(String[] args)

{

char startPeg = 'A'; // start tower in output

char endPeg = 'C'; // end tower in output

char tempPeg = 'B'; // temporary tower in output

int totalDisks = 3; // number of disks

solveTowers(totalDisks, startPeg, endPeg, tempPeg);

}

private static void solveTowers(int n, char startPeg, char endPeg, char tempPeg)

{

if (n > 0)

{

solveTowers(n - 1, startPeg, tempPeg, endPeg);

Console.WriteLine("Move disk from " + startPeg + ' ' + endPeg);

solveTowers(n - 1, tempPeg, endPeg, startPeg);

}

}

}

Verkefni 1.10

14) 837799 gaf leingsta listan með 525 tölur

const int number = 1000000;

long sequenceLength = 0;

long startingNumber = 0;

long sequence;

for (int i = 2; i <= number; i++) {

int length = 1;

sequence = i;

while (sequence != 1) {

if ((sequence % 2) == 0) {

sequence = sequence / 2;

} else {

sequence = sequence \* 3 + 1;

}

length++;

}

if (length > sequenceLength) {

sequenceLength = length;

startingNumber = i;

}

}

15) það eru 137.846.528.820 mögulegar lausnir

const int gridSize = 20;

long paths = 1;

for (int i = 0; i < gridSize; i++) {

paths \*= (2 \* gridSize) - i;

paths /= i + 1;

}

31) það eru 73682 möguleikar

int target = 200;

int ways = 0;

for (int a = target; a >= 0; a -= 200) {

for (int b = a; b >= 0; b -= 100) {

for (int c = b; c >= 0; c -= 50) {

for (int d = c; d >= 0; d -= 20) {

for (int e = d; e >= 0; e -= 10) {

for (int f = e; f >= 0; f -= 5) {

for (int g = f; g >= 0; g -= 2) {

ways++;

}

}

}

}

}

}

}