SCC.111: GAMES TIME!

THE GOAL

This week's lab aims to:

- Get you really familiar with adjusting program control flow using loops and conditions
- Thinking about 'state', the model of your game and how you model this as variables (you should only need ' `int` ' for this task)
- It also exercises your problem decomposition and solving abilities. *We would definitely recommend designing the steps that make up your program on paper first*!

"ROCK, PAPER, SCISSORS"

This week we are building a simple, classic text based game where you play against the computer (ideally in a series of rounds). The output from a dialogue with the game looks like this:

```
Welcome to rock, paper, scissors!
Please enter 1 for Rock, 2 for Paper or 3 for Scissors, and 0 to quit
1
Player chose: Rock
Machine chose: Paper
Sorry, you lost this round!
Please enter 1 for Rock, 2 for Paper or 3 for Scissors, and 0 to quit
2
Player chose: Paper
Machine chose: Scissors
Sorry, you lost this round!
Please enter 1 for Rock, 2 for Paper or 3 for Scissors, and 0 to quit
0
Games played 2, games won 0
```

The user enters a number, in this case 1 for Rock, 2 for Paper or 3 for Scissors.

e.g. using:

```
printf("Please enter 1 for Rock, 2 for Paper or 3 for Scissors, and 0 to quit\n");


scanf("%d", &playerChoice);
```

the computer then *thinks* of their number. Actually just chooses a random number:

```
machineChoice = rand() % 3 + 1;
```

then the logic of the program calculates who wins!

The rules are straightforward:

- Paper wraps rock
- Rock blunts scissors
- Scissors cut paper

otherwise, you lose. In this version, the machine would win if you make the same choice.

GETTING STARTED

We *definitely* advise you to work up in stages to the final/complete solution!

1. Start with a new text (C sourcecode) file (e.g. ‘ `rock.c` ’).
2. A basic framework will be the ‘ `main` ’ function.
3. Compile this in the terminal using ‘ `gcc` ’, i.e. depending on what you named your source file:

```
$ gcc -o rock rock.c
```

Then your executable in your current directory ‘ `.` ’ is run by:

```
$ ./rock
```

Note if your program gets 'stuck in a loop' or you just want to terminate it's execution, then press ' `Ctrl-C` ' to kill it.

GENERAL APPROACH

1. Start with the minimal program, e.g. step one might be getting the user to input their choice; or the machine. Get this compiled and working before extending further.
2. Add functionality, e.g. as follows:
   - Get the choice of the machine (or user, depending on where you started)
   - Print out the choices of each player
   - Calculate who won and output the result
3. That's one round, if you want to improve your game, make it so they play a number of rounds, or even, choose when they want to quit
4. Ideally, keep a running total of games and wins, so you can display this, e.g. at the end of the game

Think carefully about how to structure your code using loops and selection statements. *Refer back to the lecture or the book if that would help!*

You will need to be able to choose a random number. This is done using a function called ' `rand()` '. See below for an example.

The following lines show you how to generate a random number in C (note you will need the `#include` lines to tell the C compiler how to find the functions you want to use):

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main ()
{
   int card;

   /* Do this only once at the start of your program,
      otherwise your sequence of numbers (game) will
      always be the same, which is boring! */

   srand(time(NULL));

   /* Rest of code goes here :) */

   /* Choose a number between 1 and 10 - we've called this 'card'
      because the number represents a card in this example from ace to a ten
      (rand() really produces a large random int, we find the
      remainder from diving by 10 using '%' mod operator, then add 1
      so the variable card can't be 0) */

   card = rand() % 10 + 1;
   printf ("It's a %d.\n", card);
}
```

SCC121: RELATIONS

This week's lab activities will cover the topic of *relations*. Relations play a fundamental role in exploring and understanding the connections between various elements or objects. At its core, a relation is a mathematical concept that defines how two or more elements, often represented as sets, are related or connected to each other. These relationships can be based on a wide range of criteria, and they provide a powerful framework for analyzing and modeling real-world systems.

THE GOAL

Using the knowledge of the lecture material from Week 2, you are expected to work through the following questions. These problems will equip you with the essential knowledge and skills to analyze relationships, make connections, and solve complex problems. You should work out the answers *with a pen and paper*!

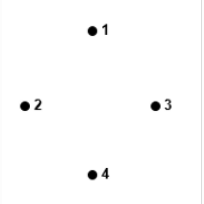
By the end of this task you should be able:

- ☐ To understand what is a relation.
- ☐ To define relations.
- ☐ To apply relations between sets, including performing unions, intersections and set differences.
- ☐ To reason about the nature of a relation.

1. For the exercises in this part, we have the following relations on A = {1, 2, 3, 4}:

   - R1 = {< 1, 1 >, < 1, 2 >, < 2, 1 >, < 2, 2 >, < 3, 4 >, < 4, 1 >, < 4, 4 >}
   - R2 = {< 2, 2 >, < 2, 3 >, < 2, 4 >, < 3, 2 >, < 3, 3 >, < 3, 4 >}
   - R3 = {< 1, 1 >, < 1, 2 >, < 1, 4 >, < 2, 1 >, < 2, 2 >, < 3, 3 >, < 4,1 >, < 4, 4 >}
   - R4 = {< 2, 1 >, < 3, 1 >, < 3, 2 >, < 4, 1 >, < 4, 2 >, < 4, 3 >}

a. Draw diagraphs for the 4 relations, using this as your starting point:



b. Which of these relations are reflexive? The ones that are not reflexive – why are they not reflexive?

c. Which of these relations are symmetric? The ones that are not symmetric – why are they not symmetric?

d. Which of these relations are transitive? The ones that are not transitive – why are they not transitive?

2. Identify the relation depicted by the following diagraphs:

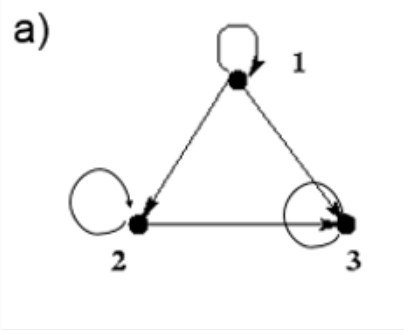


Figure 1:

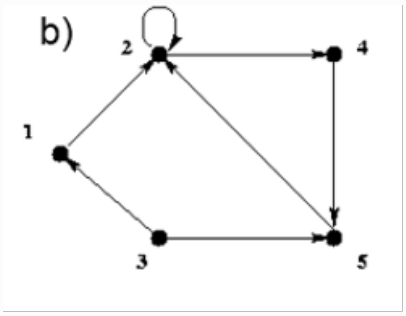


Figure 2:

3. Let R1 and R2 be two binary relations

- R1 = {<Ann, 22>, <Mary, 23>, <Laura, 20>}
- R2 = {<John, 19>, <Mike, 24>, <James, 21>, <Mary, 23>, <Laura, 20>}

What is the a) union, b) intersection and c) difference of the two relations?

4. Let A = {eggs, milk, corn} and B = {cows, goats, hens}. Define a relation R from A to B by <a, b> ∈ R if a is produced by b.

5. For each of the following relations defined on the positive integers $>, <, =, \geq, \leq$, justify whether the relation is:

   - reflexive
   - symmetric
   - transitive

   $E = \{ <1,1>, <2,2>, <3,3>, <4,4>, <5,5> \}$
   $L = \{ <1,2>, <1,3>, <1,4>, <1,5>, <2,3>,$
   $<2,4>, <2,5>, <3,4>, <3,5>, <4,5> \}$
   $G = \{ <2,1>, <3,1>, <3,2>, <4,1>, <4,2>, <4,3>, <5,1>, <5,2>, <5,3>, <5,4> \}$
   $LE = \{ <1,1>, <1,2>, <1,3>, <1,4>, <1,5>, <2,2>, <2,3>, <2,4>, <2,5>, <3,3>,$
   $<3,4>, <3,5>, <4,4,>, <4,5>, <5,5> \}$
   $GE = \{ <1,1>, <2,1>, <2,2>, <3,1>, <3,2>,$
   $<3,3>, <4,1>, <4,2>, <4,3>, <4,4>, <5,1>, <5,2>, <5,3>, <5,4>, <5,5>\}$

6. Consider the following relations on {1, 2, 3, 4}:

   - R1 = $\{<2,2>, <2,3>, <2,4>, <3,2>, <3,3>, <3,4>\}$
   - R2 = $\{<1,1>, <1,2>, <2,1>, <2,2>, <3,3>, <4,4>\}$
   - R3 = $\{<2,4>, <4,2>\}$

Which of these relations are

- reflexive
- symmetric
- transitive, and if not why not?

7. Let A= $\{a, b, c, d\}$ and R is the relation $\{<a, a>\}$. Explain if this relation is:

   - reflexive
   - symmetric

- irreflexive
- transitive

8. Specify the sets A and R, where A is the set on which the relation R is represented in the diagraph below:

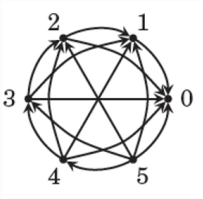


Figure 3:

9. Let A = $\{a, b, c, d\}$ and R = $\{< a, a >, < b, b >, < c, c >, < d, d >\}$. Is relation R reflexive, symmetric or transitive? If not, specify why.

SCC.131: INFORMATION CODING

In this week's lab activities, we will continue practicing number systems. In particular, we will practice negative number representation and operations such as sign and magnitude, excess n and 2's complement, and conversion between IEEE 754 floating point and decimal numbers.

THE GOAL

By further applying the knowledge of the lecture material from Week 2, you are expected to work through the following questions. You should work out the answers *with a pen and paper* - **no calculators nor computers**!

By the end of this task you should be able to:

- ☐ Convert numbers between decimal and hexadecimal systems.
- ☐ Represent negative numbers with sign and magnitude.
- ☐ Represent negative numbers with excess n.
- ☐ Perform arithmetic operations with 2's complement.
- ☐ Convert numbers between IEEE 754 floating point number and decimal.

TODAY'S TASKS

This is a worksheet for the SCC.131 week 3 lab practical. Work through the following questions and ask us if you need help. You should work out the answers *with a pen and paper* - no calculators nor computers!

1. Convert the following decimal numbers to hexadecimal:
   - 16
   - 132
   - 256
   - 5 11/16
   - 10 3/4
2. Convert the following decimal numbers to 8 bit sign and magnitude binary:

- -1
- -10
- 30
- -127
- -128

3. Convert the decimal number 3 to 8-bit binary using Excess 127.

4. Convert the following decimal numbers to 8-bit 2's complement binary and then add them (giving the answer also in 8-bit 2's complement binary). Then convert the answer back to decimal to check your answer:

   - 25 + 20
   - 30 + -1
   - -23 + 22
   - -18 + 7
   - -27 + -27

5. Convert the decimal number 9/64 into IEEE 754 floating point. Express your answer in hexadecimal.

6. Convert the hexadecimal 3F880000, representing IEEE 754 floating point number, into decimal.

HACKER EDITION

SCC.111: GAMES TIME!

If you completed the game and want to push further, try this:

1. First, extend your program with 'ties' if both you and the machine pick the same object. Your program should count the number of ties, wins and losses. A nice touch is to add the countdown to making your choice, to make gameplay more realistic. The ' `time` ' or ' `sleep` ' functions will be useful for this (see the manual ' `man` ').

2. Second, how about a nice game of "Rock Paper Scissors Lizard Spock" as popularised by TV sitcom "The Big Bang Theory"? You'll notice the rules are somewhat more complex! Think carefully about how to calculate the winner as efficiently (with as few statements) as possible.

3. A further nice touch is to print out the rule by which the player won or lost.

SCC.121: RELATIONS

1. Let R be the relation on the set of real numbers, described by x = 2y + 1. Is it reflexive? Is it symmetric? Is it transitive?

2. Which properties the relation in the following digraph has?

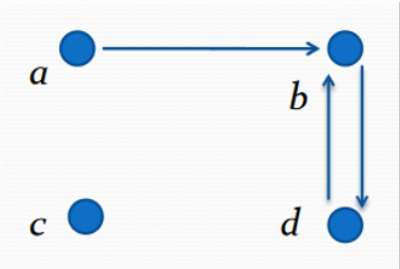


Figure 4:

3. Let's us have S, the set of finite strings of 0s and 1s, and L, the binary relation defined on S as the set of ordered pairs of strings <s, t> for which the length of s is less than or equal to the length of t. Is the binary relation L reflexive, symmetric, transitive, or equivalence relation?

Note: the length of a string is the number of characters in that string, for example the length of string: 00101 is

SCC.131: INFORMATION CODING

1. The conversion of the decimal number 0.1 to a 32-bit IEEE 754 floating-point representation generates the hex value of 0x3dcccccd. Convert the hexadecimal value into its decimal equivalent. Analyze the loss of precision during the conversion process.

2. The IEEE 754 floating-point representation is widely used and highly standardized for representing real numbers in computing systems. However, it does have several limitations. Describe a couple of limitation.