# Programming Practice

Final Project

# Final Project

- Put together all you have learned in this class
- This is long-term project.　( about 4 weeks in total )
- We're going to make a simple game
  with external library "**NCRUSES**"

# Final Project

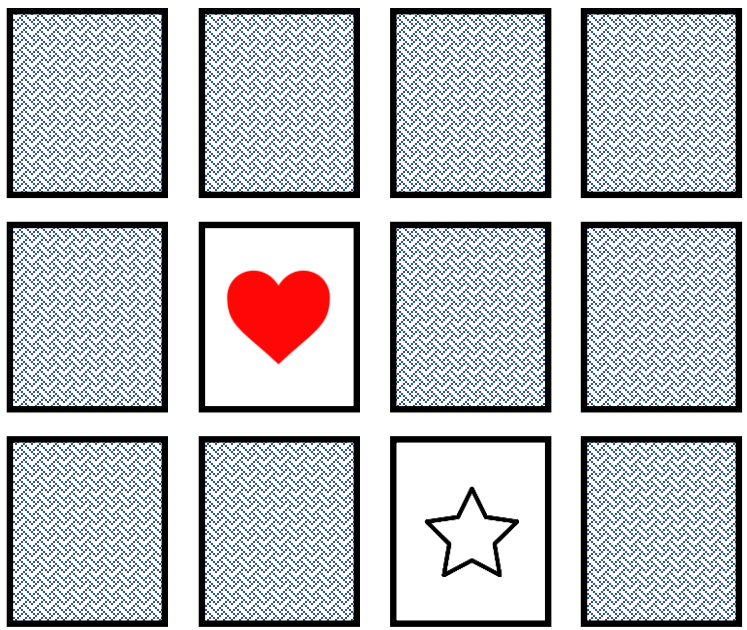
- You need to understand following features

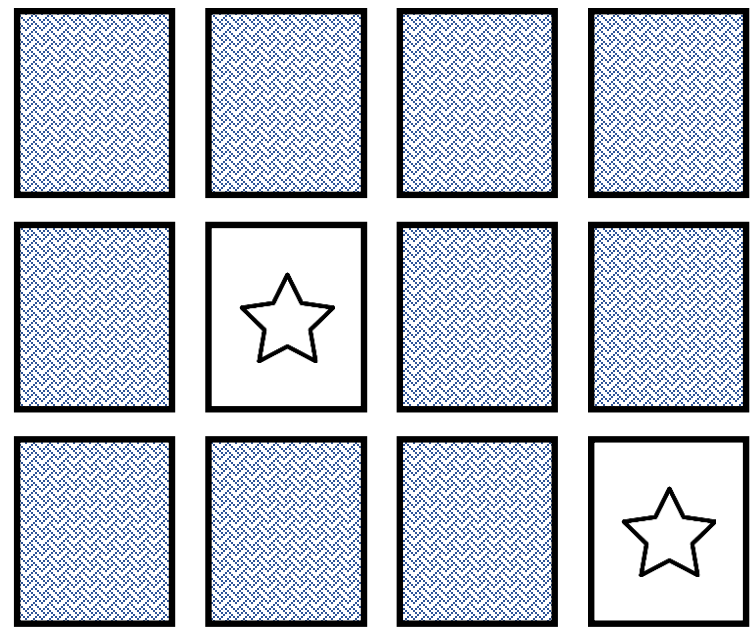
  - Basic Input / Output
  - Flow Control (if, else, switch, while, for …)
  - Functions
  - Arrays / Strings
  - Operators ( including bitwise operators )

  - … (and other stuff for extra implementation)

# Card Matching Game

- For final project, we'll make a simple "card matching game"





# Game Rules

- At the beginning, all cards are laid face down
- In turn, player chooses two cards and turns them face up
  - If matched, remove chosen cards.
  - If not, turns them face down again.
- Game finishes when all cards are removed.

# DEMO

# Tasks for 1st week

1. Get keyboard input ‘q’ or ‘Q’ to terminate your program.

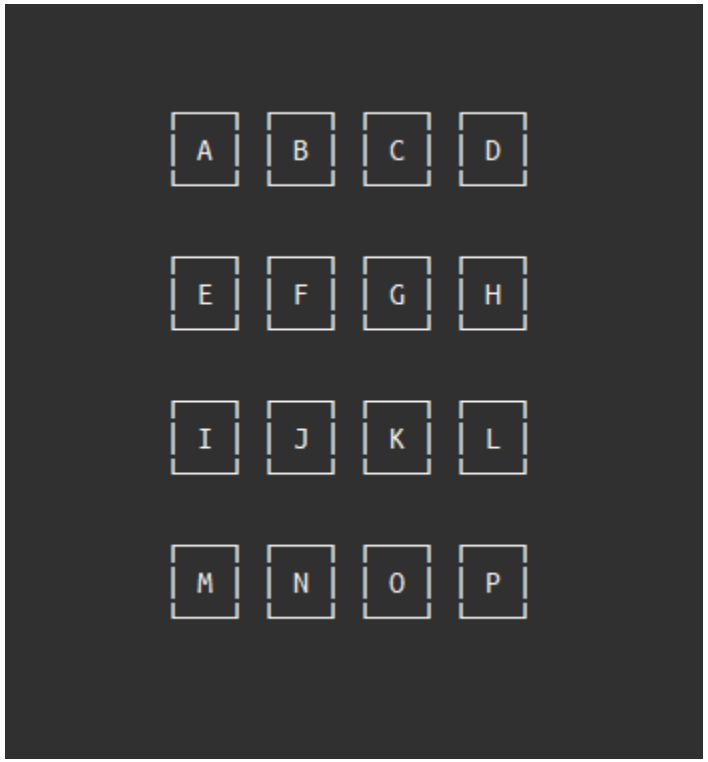
2. Display game board grid ( 4 x 4 )

3. Put alphabet from ‘A’ to ‘P’ in each box. (Picture in next slide)

**<u>Designs are all up to you.</u>**

**Use any color, any box size you want.**
**If game rules are not clear enough, define your own rules.**
**(In this case, please note new rules briefly in manual)**

# Tasks for 1st week



# Grading

- Program terminates when ‘q’ or ‘Q’ is pressed. (10 pt)
- Display 4 x 4 Grid (20 pt)
- Put alphabet in each box (20 pt)

| | |
|---|---|
| Grading Environment | Lab Computer & Martini Server<br>(Your code should be compiled & run correctly at least one of those environments) |
| Compile Command | **> gcc <yourfile.c> -o game -O2 -lm -lncurses** |

# Submission

- Submit by E-mail pp20182ta@gmail.com

- Email title format: [project] week# <student id> <name>

  example1) [project] week1 2018-12345 Chris Davis

  example2) [project] week3 2017-12321 홍길동

- 2 Files are needed.
  - `201x-xxxxx.c` ( Use your student ID as filename )
  - `manual.pdf` ( Brief explanation: How to Play, Newly defined rules )

# NCURSES Library

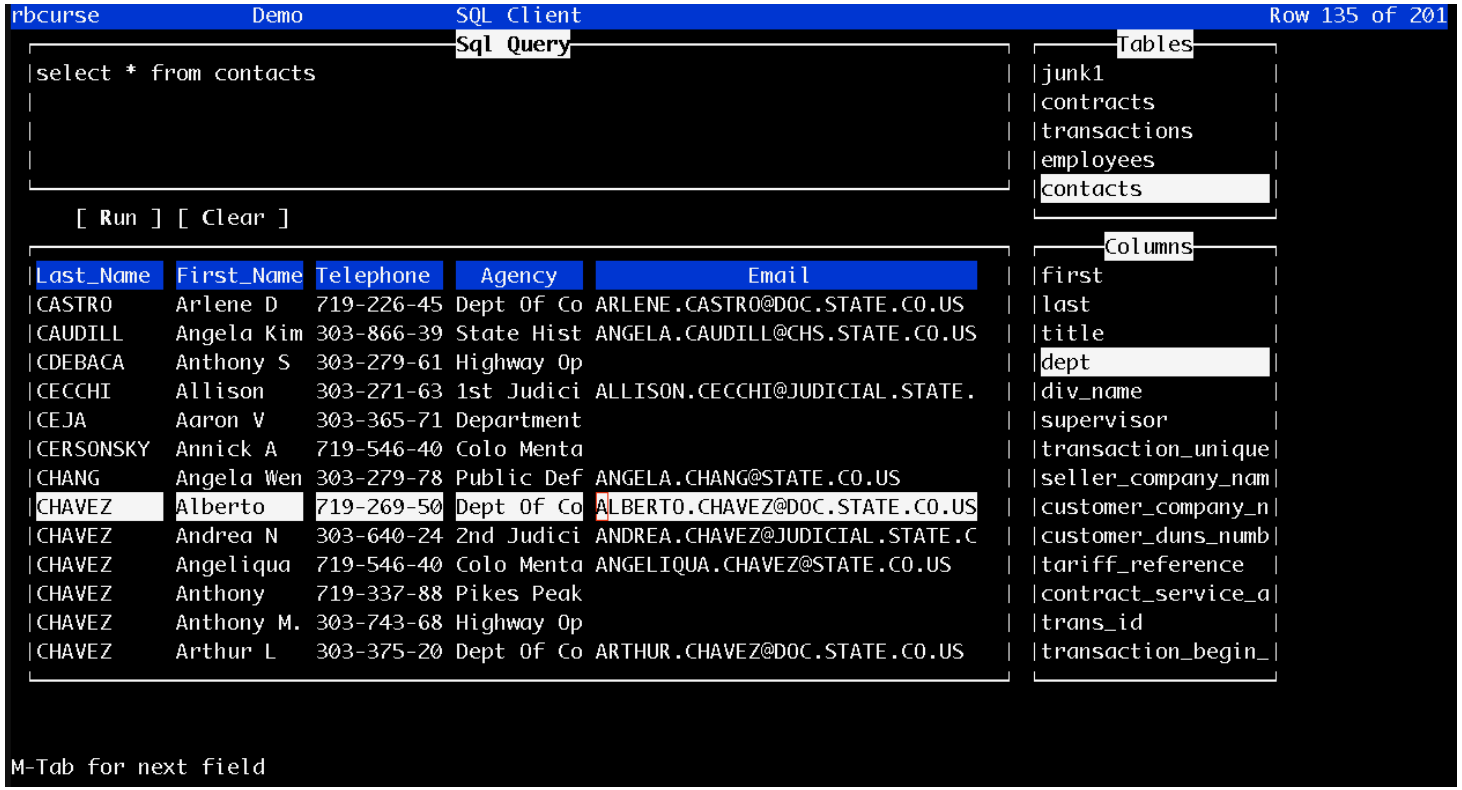
# NCURSES

- Programming library
- Provide Application Programming Interface (API)
  that allows to write 'Text-based User Interface'
- It helps us to develop 'GUI-like' application software

**Library** : Group of non-volatile resources that program uses.
You can simply understand it as a group of pre-defined functions & global variables

**GUI** : Graphical User Interface

# NCURSES Library




Text-based GUI Example

# NCURSES Library - How to Install

**Strongly recommend you to use “Linux” Environment.**
Other OS might uses slightly different version.

NCURSES library is already installed on Lab Computers and Martini Server.
Those who uses these environment, don’t have to care about installation.

If you want to install NCURSES library on your own Linux(Ubuntu) machine, type following command on terminal.
You should have administrative access control.

```
> sudo apt-get install libncurses5-dev libncursesw5-dev
```

# NCURSES Library - Mode Setting Functions

| | |
|---|---|
| `initscr()` | `Initialize Output Screen.`<br>`Called only once at the beginning of the program.` |
| `raw()` | `Raw keyboard input setting.`<br>`Keyboard input will be directly sent to program.` |
| `keypad(stdscr, TRUE)` | `Use extended keyset. It enables arrow keys.` |
| `curs_set(0)` | `Do not show cursor.` |
| `noecho()` | `Do not show keyboard input on screen.` |
| `start_color()` | `Start using color attributes.` |
| `endwin()` | `Restore all saved shell terminal mode.`<br>`Called only once at the end of the program.` |

These functions are used for modifying basic settings.
Usually called only once.

# NCURSES Library - Input Functions & Constants

| | |
|---|---|
| `int getch(void)` | Get keyboard input.<br>It Returns “Integer Type” keyboard input value.<br>All ascii characters are mapped as same value. (‘A’ = 65) |
| `KEY_ENTER` | ‘Enter’ Key Input. Please use both KEY_ENTER and ‘\n’. |
| `KEY_UP, KEY_DOWN,`<br>`KEY_LEFT, KEY_RIGHT` | Arrow Keys |
| `KEY_BACKSPACE` | ‘BACKSPACE’ Key Input. |

Usage Example

```
int input = getch();
if (input == ‘\n’ || input == KEY_ENTER) {...}
else if (input == KEY_UP) {...}
else if (‘A’ <= input && input <= ‘Z’) {...}
else {...}
```

Other KEYs are supported. You can google it.

Do NOT use ESC Key due to stability issue.
Use ‘Q’ and ‘q’ instead of ESC.

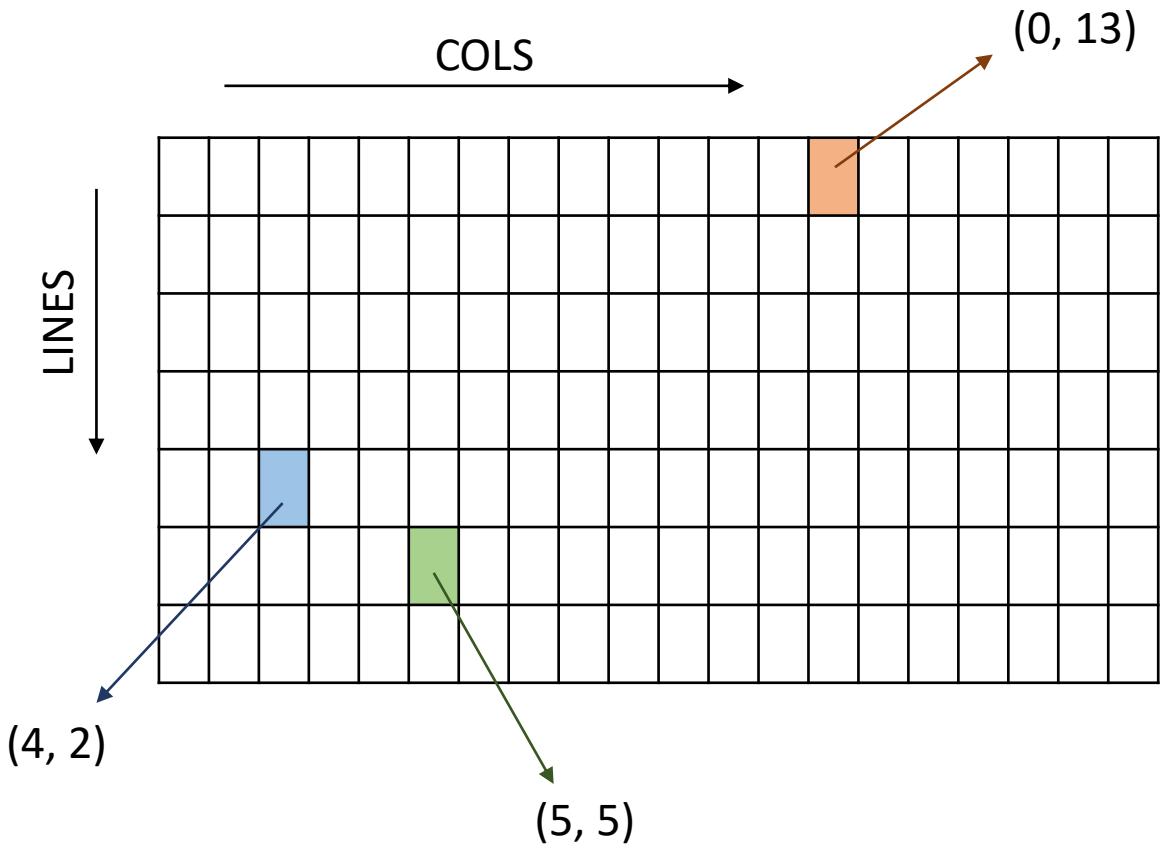
https://www.gnu.org/software/guile-ncurses/manual/html_node/Getting-characters-from-the-keyboard.html

# NCURSES Library - Display Functions

| | |
|---|---|
| clear() | Clear screen. If you don’t use this function before print, new items will be over-writtin on previous screen. |
| move(Line, Col) | Move cursor to given position. |
| addch(KEY) | Print 1 character on screen at current cursor position. |
| mvaddch(Line, Col, KEY) | Print 1 character at given position. |
| printw(format, args, …) | NCURSES version of printf function. Same usage with printf. String will be printed at current cursor position. |
| mvprintw(Line, Col, format, args, …) | NCURSES version of printf function with cursor position. String will be printed at given position. |
| refresh() | Display printed output to terminal. All printed items are saved internally[invisible] before this function called. |

You can implement with mv* functions only. Then you don’t have to care about cursor position.

# Line & Col




NCURSES library provides following 2 constants.

**LINES**
**COLS**

You can just divide them by 2
to get an index of middle point of the screen.

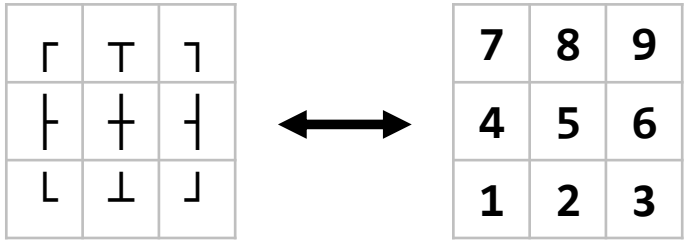
# NCURSES Library - Alternative Character Set (box)

| | | |
|---|---|---|
| ACS_LLCORNER | Lower Left Corner | └ |
| ACS_LRCORNER | Lower Right Corner | ┘ |
| ACS_ULCORNER | Upper Left Corner | ┌ |
| ACS_URCORNER | Upper Right Corner | ┐ |
| ACS_BTEE | Bottom T | ┴ |
| ACS_LTEE | Left T | ├ |
| ACS_RTEE | Right T | ┤ |
| ACS_TTEE | Top T | ┬ |
| ACS_PLUS | Plus | ┼ |
| ACS_HLINE | Horizontal Line | ─ |
| ACS_VLINE | Vertical Line | │ |

To make easier to use, translation function will be given in `skeleton.c` file.

**`int box_char(int x)`**

Characters are mapped like NUM-Keypad




```
ACS_HLINE : 10
ACS_VLINE : 11
```

# NCURSES Library - Character / Keyboard Input

- NCURSES library uses **`int`** type for both character / keyboard input
- All ASCII characters from 0 to 127 are mapped as same integer value.
- Extended characters / Keyboard input are mapped from 128.

Usage Example

```
int keyboard_input = getch();
if (keyboard_input == KEY_UP)
        addch(ACS_LLCORNER);
mvprintw(3, 4, “%d”, keyboard_input);
```

# NCURSES Library - Colors

| | |
|---|---|
| `init_color(COLOR, short r, short g, short b)` | Set color with RGB value.<br>RGB values are mapped in range (0 ~ 999) |
| `init_pair(PAIR, COLOR1, COLOR2)` | Set color pair. COLOR1 for text, COLOR2 for background. |

Pre-Defined Colors

| | |
|---|---|
| COLOR_BLACK | 0 |
| COLOR_RED | 1 |
| COLOR_GREEN | 2 |
| COLOR_YELLOW | 3 |
| COLOR_BLUE | 4 |
| COLOR_MAGENTA | 5 |
| COLOR_CYAN | 6 |
| COLOR_WHITE | 7 |

Red: 600
Green: 749
Blue: 400

Usage Example

```
init_color(8, 600, 749, 400);
init_pair(1, COLOR_BLACK, COLOR_CYAN);
init_pair(2, COLOR_RED, 8);
```

# NCURSES Library - Display Attributes

Use bitwise ‘OR’ to combine multiple attributes.

| | |
|---|---|
| attron(ATTRIBUTES) | Enable given attributes. |
| attroff(ATTRIBUTES) | Disable given attributes. |

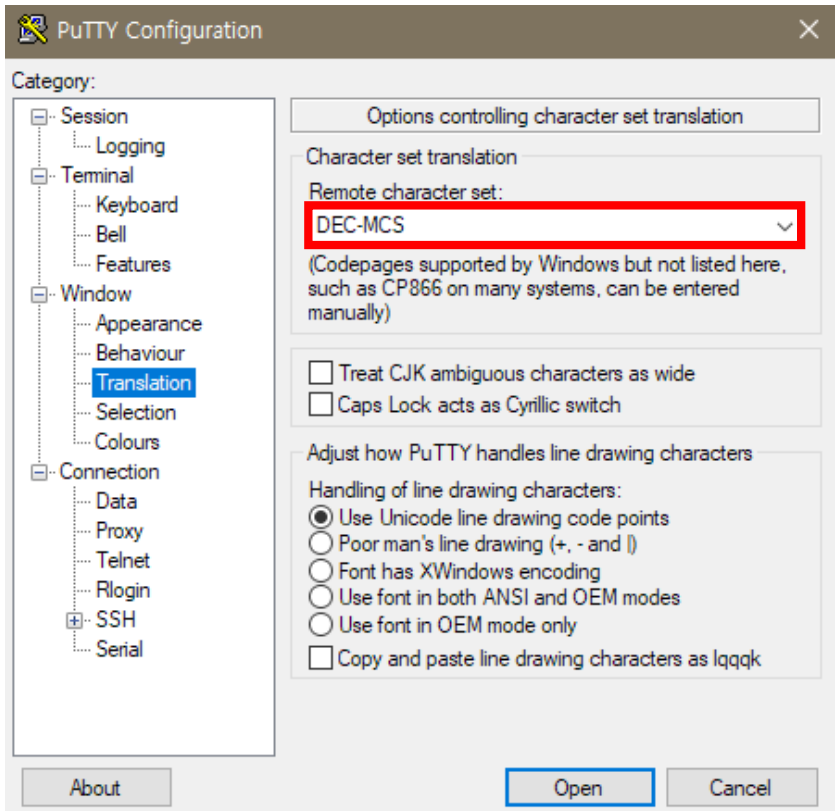
| | |
|---|---|
| A_BOLD | Extra bright or bold |
| A_UNDERLINE | Underlining |
| COLOR_PAIR(n) | Set text and background color as input. (Your color pair) |

Usage Example

```
attron(A_BOLD | A_UNDERLINE | COLOR_PAIR(1));
addch(‘A’);
attroff(A_UNDERLINE);
mvaddch(5, COLS/2, ACS_URCORNER);
attroff(A_BOLD);
attron(COLOR_PAIR(2));
mvprintw(3, 4, “%d”, 56);
```

You can search
more attribute options on google.

# Putty Settings




Window -> Translation

Set Remote character set as ‘DEC-MCS’

# Practice

- Given .zip file contains 2 files.

| | |
|---|---|
| **`libtest.c`** | Only for test. You don't need to submit this file.<br>Be familiar with NCURSES library. |
| **`skeleton.c`** | Start your task with this file.<br>Please change the filename before submit. Do NOT submit as "**`skeleton.c`**" |

HOW TO COMPILE?

```
> gcc <filename> -lncurses

> gcc libtest.c -lncurses
> gcc skeleton.c -lncurses
```

( If you used **`math.h`** library, put **`-lm`** option also )