Programming 2
Exercises 1

# Introduction

Nedim Šišić

2023

## General information

- Teaching assistant: Nedim Šišić
- Email: nedim.sisic@famnit.upr.si
    - Please direct all online questions to email!

## General information

- https://e.famnit.upr.si/mod/page/view.php?id=97648
- Written exam (50%) + Homework (35%) + Quizzes (15%)
- Homework:
  - Each day past deadline:  - 2%
  - **Must** be your own work
  - In the event of cheating, university rules apply.

## Running OCaml

- Windows:
  - https://fdopen.github.io/opam-repository-mingw/installation/, graphical installer
  - Run OCaml in OCaml64 shell
- Linux:
  - https://ocaml.org/docs/up-and-running
- Online:
  - https://try.ocamlpro.com/

## Running OCaml

Executing commands in shell:

- run by executing `ocaml`
    - `#` signifies OCaml is waiting for commands
- end input with double semicolon: `;;`
    - if not, compiler waits for further instruction
- exit by `exit 0;;` , or CTRL + C (watch out when copying!)

## Running OCaml

Compiling and running a program:

- navigate to toplevel directory (`Sys.getcwd()` in OCaml)
- create file `helloworld.ml` containing:

```
print_string "Hello World!\n";;
```

- compile file with:
  - `ocamlc -o helloworld helloworld.ml`
- run with:
  - `./helloworld`

## Expressions

- An expression is executable code which returns a value
  - a value is of some type
- OCaml data types:
  - unit
  - int
  - bool
  - char
  - list
  - custom types, …

## Expressions

- An expression is executable code which returns a value
    - a value is of some type
- OCaml data types:
    - unit
    - int
    - bool
    - char
    - list
    - custom types, …

## Basic types

### Integer

-
  ```
  # 4;;
  - : int = 4
  ```

-
  ```
  # 2 + 5;;
  - : int = 7
  ```
- Operators: +, -, *, /, mod
  - precedence: `2 * 5 mod 10;;`
- Limited range: max_int and min_int
  - `max_int + 1;;`

## Basic types

### Integer

- `# 4;;`
  `- : int = 4`
- `# 2 + 5;;`
  `- : int = 7`
- Operators: +, -, *, /, mod
  - precedence: `2 * 5 mod 10;;`
- Limited range: max_int and min_int
  - `max_int + 1;;`

### Float

- `# 4.0;;`
  `- : float = 4`
- `# 4.;;`
  `- : float = 4`
- `# 2. +. 5. ;;`
  `- : float = 7`

Cannot compare int and float directly; use type conversion!

## Basic types

### Boolean

- `# true;;`
  `- : bool = true`
- `# false;;`
  `- : bool = false`
- Operators: `not, &&, ||`
- precedence:
  - `not true || true;;`
  - `false &&  false || true;;`

## Basic types

Char

- `# ‘a’;;`
  `- : char = ‘4’`

String

- `# “Hello World!”;;`
  `- : string = "Hello World!“`

## Basic types

### Char

-
  ```
  # ‘a’;;
  - : char = ‘4’
  ```

### String

-
  ```
  # “Hello World!”;;
  - : string = "Hello World!“
  ```

### Cannot be compared:

-
  ```
  'a' = "a";;
  ```

  Error: This expression has type string but an expression was expected of type char

## Type conversion

-
  ```
  # float_of_int 1;;
   - : float = 1.
  ```

-
  ```
  # int_of_float 1.;;
   - : int = 1
  ```

-
  ```
  # string_of_int (1+1);;
   - : string = "2"
  ```

-
  ```
  # int_of_char '1';;
  ```

## Type conversion

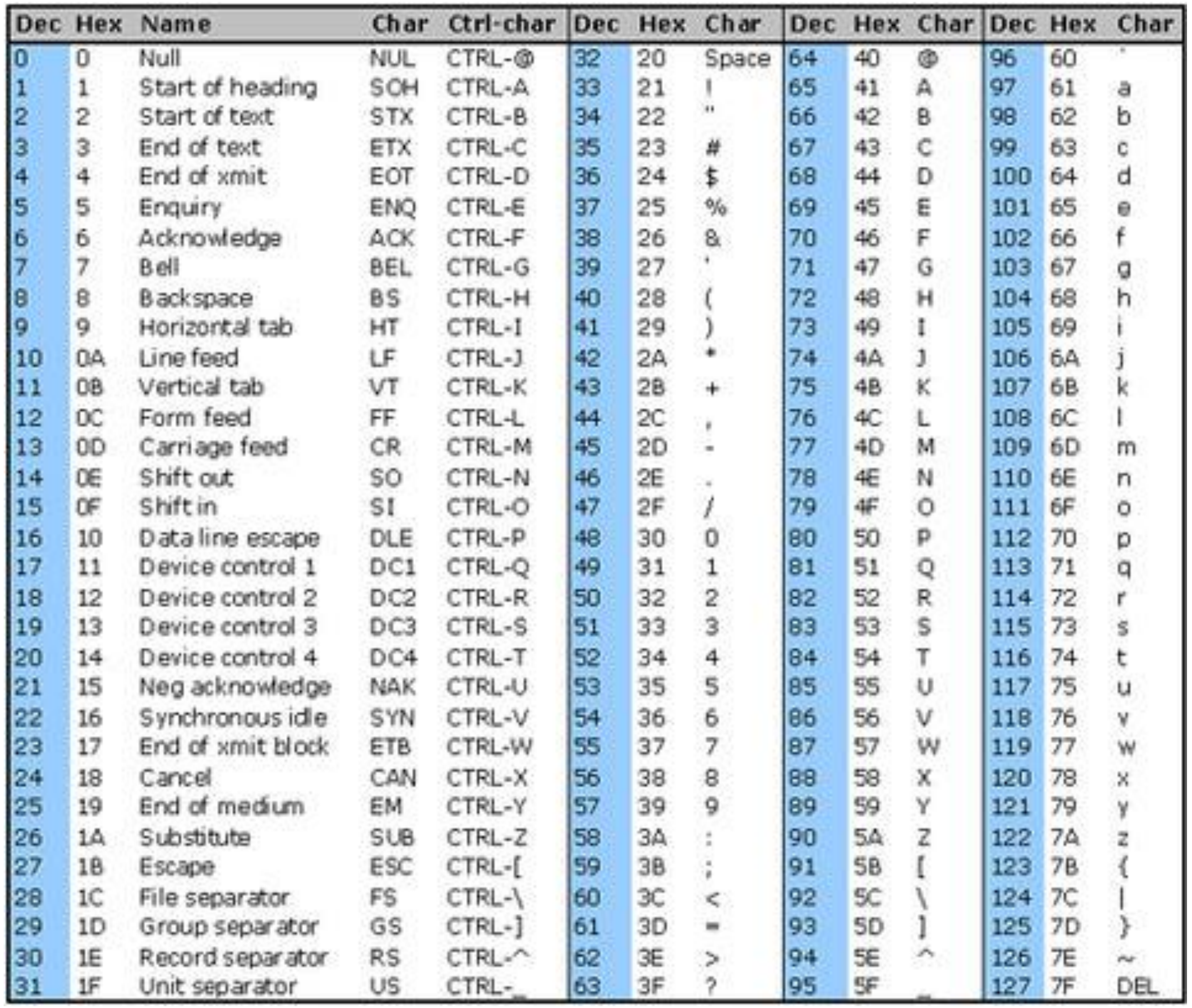
-
  ```
  # float_of_int 1;;
   - : float = 1.
  ```

-
  ```
  # int_of_float 1.;;
   - : int = 1
  ```

-
  ```
  # string_of_int (1+1);;
   - : string = "2"
  ```

-
  ```
  # int_of_char '1';;
  - : int = 49
  ```

-
  ```
  # char_of_int 98;;
  - : char = 'b'
  ```



| Dec | Hex | Name | Char | Ctrl-char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Null | NUL | CTRL-@ | 32 | 20 | Space | 64 | 40 | @ | 96 | 60 | ` |
| 1 | 1 | Start of heading | SOH | CTRL-A | 33 | 21 | ! | 65 | 41 | A | 97 | 61 | a |
| 2 | 2 | Start of text | STX | CTRL-B | 34 | 22 | " | 66 | 42 | B | 98 | 62 | b |
| 3 | 3 | End of text | ETX | CTRL-C | 35 | 23 | # | 67 | 43 | C | 99 | 63 | c |
| 4 | 4 | End of xmit | EOT | CTRL-D | 36 | 24 | $ | 68 | 44 | D | 100 | 64 | d |
| 5 | 5 | Enquiry | ENQ | CTRL-E | 37 | 25 | % | 69 | 45 | E | 101 | 65 | e |
| 6 | 6 | Acknowledge | ACK | CTRL-F | 38 | 26 | & | 70 | 46 | F | 102 | 66 | f |
| 7 | 7 | Bell | BEL | CTRL-G | 39 | 27 | ' | 71 | 47 | G | 103 | 67 | g |
| 8 | 8 | Backspace | BS | CTRL-H | 40 | 28 | ( | 72 | 48 | H | 104 | 68 | h |
| 9 | 9 | Horizontal tab | HT | CTRL-I | 41 | 29 | ) | 73 | 49 | I | 105 | 69 | i |
| 10 | 0A | Line feed | LF | CTRL-J | 42 | 2A | * | 74 | 4A | J | 106 | 6A | j |
| 11 | 0B | Vertical tab | VT | CTRL-K | 43 | 2B | + | 75 | 4B | K | 107 | 6B | k |
| 12 | 0C | Form feed | FF | CTRL-L | 44 | 2C | , | 76 | 4C | L | 108 | 6C | l |
| 13 | 0D | Carriage feed | CR | CTRL-M | 45 | 2D | - | 77 | 4D | M | 109 | 6D | m |
| 14 | 0E | Shift out | SO | CTRL-N | 46 | 2E | . | 78 | 4E | N | 110 | 6E | n |
| 15 | 0F | Shift in | SI | CTRL-O | 47 | 2F | / | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | Data line escape | DLE | CTRL-P | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | Device control 1 | DC1 | CTRL-Q | 49 | 31 | 1 | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | Device control 2 | DC2 | CTRL-R | 50 | 32 | 2 | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | Device control 3 | DC3 | CTRL-S | 51 | 33 | 3 | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | Device control 4 | DC4 | CTRL-T | 52 | 34 | 4 | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | Neg acknowledge | NAK | CTRL-U | 53 | 35 | 5 | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | Synchronous idle | SYN | CTRL-V | 54 | 36 | 6 | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | End of xmit block | ETB | CTRL-W | 55 | 37 | 7 | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | Cancel | CAN | CTRL-X | 56 | 38 | 8 | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | End of medium | EM | CTRL-Y | 57 | 39 | 9 | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | Substitute | SUB | CTRL-Z | 58 | 3A | : | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | Escape | ESC | CTRL-[ | 59 | 3B | ; | 91 | 5B | [ | 123 | 7B | { |
| 28 | 1C | File separator | FS | CTRL-\ | 60 | 3C | < | 92 | 5C | \ | 124 | 7C | \| |
| 29 | 1D | Group separator | GS | CTRL-] | 61 | 3D | = | 93 | 5D | ] | 125 | 7D | } |
| 30 | 1E | Record separator | RS | CTRL-^ | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | Unit separator | US | CTRL-_ | 63 | 3F | ? | 95 | 5F | _ | 127 | 7F | DEL |

## Global Variables

### Command `let`

-
  ```
  # let x = 10;;
  val x : int = 10
  ```

-
  ```
  # let x = x < 20;;
  val x : bool = true
  ```

### Command `and`

-
  ```
  let c = 18 and d = 2 and e = 7;;
  val c : int = 18
  val d : int = 2
  val e : int = 7
  ```

-
  ```
  let i = 1 and j = i + 2;;
  Unbound value i
  ```

## Exercises

- Determine if `not true || true;;` is equal to`:`
  - `(not true) || true;;`
  - `not (true || true);;`
- Determine if `false &&  false || true;;` is equal to`:`
  - `(false &&  false) || true;;`
  - `false &&  (false || true);;`
- See if an int value of 10 is lesser than a float value of 20 (use type conversion)

## Exercises

- Define a variable of the following type:
  - int
  - float
  - bool
  - string

- Declare a variable `lowercase : char` and assign any lower character to it. Then declare a variable `uppercase : char` and assign it the upper case character of the character in `lowerCase` using type conversion. Use the ASCII table for help.

## N-tuples

n-tuples are written inside (, ), elements are separated with commas. Elements of n-tuples can be of the same or different types.

-
  ```
  # (1,2);;
  - : int * int = (1, 2)
  ```

-
  ```
  # ("abc", 45, ’c’,true);;
  - : string * int * char * bool = ("abc", 45, ’c’, true)
  ```
- We can get the first and second elements of a 2-tuple with:

  -
    ```
    # fst(3,23);;
    - : int = 3 #
    ```

  -
    ```
    snd(3,23);;
    - : int = 23
    ```

For larger tuples the commands do not work

## N-tuples

n-tuples are written inside (, ), elements are separated with commas. Elements of n-tuples can be of the same or different types.

-
  ```
  # (1,2);;
  - : int * int = (1, 2)
  ```

-
  ```
  # ("abc", 45, ’c’,true);;
  - : string * int * char * bool = ("abc", 45, ’c’, true)
  ```
- We can get the first and second elements of a 2-tuple with:

  -
    ```
    # fst(3,23);;
    - : int = 3 #
    ```

  -
    ```
    snd(3,23);;
    - : int = 23
    ```

For larger tuples the commands do not work