Things to take care of

- How to read binary files?
- How to handle size of input dynamically?
- Data correct?
  - Endianess? (endian.h)
  - all positive?
  - Doublicates (two methods)
- How to treat errors

Handle size of input dynamically

```
long get_file_size(FILE* file)
{
    fseek(file, 0L, SEEK_END);
    long size = ftell(file);
    rewind(file);
    return size;
}
```

- Alternatives
- Note race condition / security

Not so nice

```
fread((void*)buffer,sizeof(int),500000001,Quelldatei);


for(int i=0; i<500000001; i++){
numbersBig[intNumber-1073741824]=true;
while(fread(&a,1,sizeof(int),fp) == sizeof(int))
```

Design decisions

1. How to read the input

   direct read

   buffered fread

   memory mapped

Design decisions

2. Sort or Bitmap ?

Bitmap:

- only works for ints up to 31 bit
- is a little slower for small numbers of ints
- highly specialized, no choice regarding dublicates or negatives

+ is faster for larger numbers of ints

+ takes less space for larger numbers of ints

Design decisions

3. Store data in static buffer or allocate storage on the heap?

Honorable mention

```
#! /usr/bin/env python
import sys
import numpy as np

filename = sys.argv[1]
data = np.fromfile(filename, dtype='int32')
data = np.sort(data)
data.tofile(sys.stdout, sep='\n')
print('') # final newline
```

- Still negatives, doublicates

Experiment: Collecting Data at CO@Work-II

Combinatorial Optimization at Work II took place at ZIB from September 21 to October 9, 2009 with 105 participants from 23 countries.

We wanted to compute the seat allocation for the lecture hall.

To do this we required ever participant to state their preferences.

Everyone should send an email with a data file.

Lets see how long it took...

File Format

ASCII text with only a LF (ASCII 10) as line separator.

Fields are separated by a single space (ASCII 32)

Line 1: ParticipantNo HasLaptop EmailAddress

e.g. `67 1 koch@zib.de`

0 = has no Laptop, 1 = has a Laptop

Lines 2-???: SeatNumber PreferenceValue

- Seat numbers start down at the low entrance, left to right, row by row.
- The hightest numbered seat is at the window side at the top.
- Count only seats that are physically there.
- The seat numbers in the file should be monotonically increasing.
- The preference values should be between 0 and 100.

e.g.

```
12 55
13 40
14 35 …
```

Rules Regarding Preference Values

Allowed values are between 0 and 100

Only seats which are not available for the participants are allowed to get a value of 0

All numbers 1-100 have to be used at least once

The average has to be between 40-60

The difference to an adjacent seat has to be < 40

The difference to a neighboring seat has to be < 20

The data should not be randomly generated

Specifying Preference Offsets

Lines ???-???: ParticipantNo   PreferenceOffset
List indicating persons which you would like or not like to be your seat neighbor.
(You have to know the ParticipantNo of the person.)

- A ParticipantNo of 0 indicates an empty seat.

- The PreferenceOffset is between -20 and 20 and will be added to your PreferenceValue if the person with the given ParticipantNo is your neighbor.
  e.g.
  ```
  55 17
  27 -5
  72 8
  0 -10 …
  ```

- This list can have as many entries as you like, but there should be at least 2 entries, and the occurring participant numbers have to be unique and valid.

How To Submit

Submission of this file is required for the course

The name of the file has to be *ParticipantNo*`.txt`

It should be **attached** to an email

Send the email to koch@zib.de

The subject of the email should be
CO@Work: SeatData for *ParticipantNo*

Please, as soon as possible.

2 Days after the lecture

Mails received : 13

Different Subjects : 4 (10 1 1 1)

Wrong field spacing : 4

Seat counts : 2 (12 1)

Missing data : 1

Too much data : 1

Ok, from first view : 5 out of 13

3 Days after the lecture

Mails received : 23

Different Subjects : 6 (17 2 1 1 1 1)

Wrong field spacing : 4

Seat counts : 4 (19 1 1)

Missing data : 2

Too much data : 0

Ok, from first view : 10

Corrected : 1

Add to the specification:

A seat without a desk is not allowed for the participants
Seats with a 0 preference value are not relevant for the adjacency/neighboring difference rules.

4 Days after the lecture

Mails received	: 37
Wrong subject	: 11
Wrong field spacing	: 8
Strange seat counts	: 5
Missing data	: 2
Corrected	: 3

5 Days after the lecture

Mails received	: 47
Data sets	: 41 (6 corrections)
Wrong subject	: 12
Wrong attachment name	: 2
Wrong line separator	: 29
Wrong field separator	: 10
Pref. value not used	: 11
Other Errors	: 1
Number of seats	: 153 - 181
No complains so far	: 4

7 Days after the lecture

Mails received	: 79
Data sets	: 64
Wrong subject	: 16
Wrong attachment name	: 2
Wrong line separator	: 45
Wrong field separator	: 11
Pref.value not used	: 22
Other Errors	: 2
Number of seats	: 153 - 181
No complains so far	: 8

9 Days after the lecture

Mails received	: 104
Data sets	: 76
Wrong subject	: 18
Wrong attachment name	: 2
Pref. value not used	: 19
Neighbor difference	: 21
Wrong no/seq. seats:	: 10
Wrong 0 seats	: 20
No complains so far	: 10

Overview of Errors in Data

	E7	E10	E11	E12	E13	E14	E16
5							X
6							X
12					X		X
13							X
16							X
18					X		X
19						X	X
20						X	
23					X		
24						X	
26							X
27						X	
36						X	
42					X		
45			X	X	X	X	X
47					X		
53					X		
59						X	
63			X		X	X	
64			X		X	X	X
71					X	X	

E7 bad seatno

E10 bad offset

E11 wrong seatno

E12 bad average

E13 prefval missing

E14 neigbour diff

E16 seat not 0

	E7	E10	E11	E12	E13	E14	E16
77							X
78	X		X			X	X
81				X	X		X
98					X		
99	X		X			X	
103					X	X	
107			X			X	X
108			X			X	X
111							X
121							X
128			X			X	X
129		X					
134			X	X	X	X	
135					X		
137		X			X	X	X
139					X		X
145	X		X		X	X	
160						X	
166					X	X	

Please correct and resubmit

11 Days after the lecture

Mails received	: 144
Wrong subject	: ~23
Wrong attachment name	: 4

Data sets	: 92
To be corrected	: 28
Missing	: 6

Pref. value not used	: 14
Neighbor difference	: 18
Wrong no/seq. seats	: 2

Overview of Errors in Data

	E7	E10	E11	E12	E13	E14
12					X	
18					X	
23	X		X			X
24						X
27						X
45					X	X
47					X	
63			X		X	X
71					X	X
78	X		X			X
79		X	X	X	X	
103						X
107			X			X
108			X			X
110		X				
114						X
118					X	X
128			X			X
134			X	X	X	X
135					X	
136						X
137		X			X	X
138					X	
139					X	
160						X
166					X	X

E7 bad seatno

E10 bad offset

E11 wrong seatno

E12 bad average

E13 prefval missing

E14 neigbour diff

Please correct and resubmit

13 Days after the lecture

Mails received	: 159
Wrong subject	: ~26
Wrong attachment name	: 4

Data sets	: 94
To be corrected	: 18
Missing	: 4

Preference value not used	: 9
Neighbor difference	: 14
Wrong no/sequence seats	: 3

Overview of Errors in Data

	E7	E10	E11	E12	E13	E14
18					X	
24						X
27						X
45					X	X
63					X	
71					X	X
78	X		X			X
79		X	X	X	X	
103						X
107			X			X
108			X			X
114						X
118					X	X
128			X			X
134			X	X	X	X
136						X
137		X			X	X
138					X	

E7 bad seatno

E10 bad offset

E11 wrong seatno

E12 bad average

E13 prefval missing

E14 neigbour diff

Please correct and resubmit

14 Days after the lecture

Mails received	: 166
Wrong subject	: ~28
Wrong attachment name	: 4

Data sets	: 95
To be corrected	: 18
Missing	: 3

Preference value not used	: 7
Neighbor difference	: 14
Wrong no/sequence seats	: 3

Overview of Errors in Data

	E7	E10	E11	E12	E13	E14
24						X
27						X
45					X	X
71					X	X
78	X		X			X
79		X	X	X	X	
92					X	X
107			X			X
108			X			X
114						X
118					X	X
128			X			X
134			X	X	X	X
136						X
137		X			X	X

E7 bad seatno

E10 bad offset

E11 wrong seatno

E12 bad average

E13 prefval missing

E14 neigbour diff

Please correct and resubmit

15 Days after the lecture – the final day

Mails received	: 172
Wrong subject	: ~31
Wrong attachment name	: 4
Data sets	: 95
To be corrected	: 13
Preference value not used	: 5
Neighbor difference	: 13
Wrong no/sequence seats	: 2

Subject Variations

The subject of the email should be

CO@Work: SeatData for *ParticipantNo*

```
CO@Work: SeatData for 022
CO@Work:SeatData for 222
CO@Work:SeatDatafor222
CO@work: SeatData for 222
CO@Work: Seat Data for 222
Co@Work: SeatData for 222
CO@Work: SeatData for Participant222
CO@Work: SeatData for ParticipantNo
Co@Work: SeatData for Participan222
CO@WORK:  seatdata for 222
COatWork: SeatData for 222
COatWork for 222
SeatData for 222
SeatData for ParticipantNo 222
set data for participant number 222
data set participant number 222
Sitting assignment
Seats assignment
```

Overview of Errors in Data

	E7	E10	E11	E12	E13	E14
24						X
27						X
45					X	X
71					X	X
78	X		X			X
92					X	X
107			X			X
108			X			X
114						X
128			X			X
134			X	X	X	X
136						X
137		X			X	X

E7 bad seatno

E10 bad offset

E11 wrong seatno

E12 bad average

E13 prefval missing

E14 neigbour diff

Sorry,
too late to correct!

Wrong line 1: 81, 129

Data

You would think a …

- … cellular network operator knows where its base stations are located?
- … fixed network operator can tell where the parts of its network are connected?
- … chemical company knows how many plants they have?
- … 5 m long pipeline cannot have a height difference from end-to-end of 100 m?

- Many companies have their data in Excel.
  There is no formal validation or referential integrality check.
- If they did formal validation, usually they found there was information they needed which they could not input and they started to “reuse” some data fields.
- If there is not at least 1 error per 100 data sets you are not looking hard enough.
- Usually the data changes all the time.
- They might not want to give it to you.
- The data might just not exist.

The first result of an optimization project is usually to improve the quality of planning data available at the company.

Exercise 2: What to do

Write a program in C or your favorite compiled language, which takes no input and produces a copy of its own source code as its only output.

The program should be as short as possible (not important) and have at least one character (because there are languages where the empty program is a valid program).

The standard terms for these programs in the computability theory and computer science literature are "self-replicating programs", "self-reproducing programs", and "self-copying programs".

Otherwise it is called a \emph{Quine}.

{\bf Please}, given the info above it is easy enough to look this up in the Internet. The purpose of this excercise is that you try it youself.

We will discuss the most interesting ones in the lecture.

\section*{How to report}

Exercise 2: How to report

Send the source code to <thorsten.koch@tu-berlin.de> with a subject of APPFS ex2 vorname nachname

Combinatorial Optimization at Work

A course at TU Berlin in cooperation with MATHEON, the Berlin Mathematical School and COST TD1207

Everything you every wanted to know about LP/MIP and real-world industrial applications

(lectures and exercises)

Dates of the course:	September 28 – October 9, 2015
Language:	English
Location:	Zuse Institute Berlin
Application deadline:	August 1, 2015
Participation fee:	none
URL (info/application):	http://co-at-work.zib.de
Intended audience:	master/PhD students, Post-docs
Contact:	coaw@zib.de
Lectures by:	M. Grötschel, B. Bixby, A. Martin, R. Borndörfer, T. Koch, the SCIP Team, developers of Gurobi, XPress, CPlex, Mosek, Gams, ThinkCubic, and many more