Die Transpositionschiffre

Inf-Einf-B · Vorlesung 3, zweiter Termin, erster Block

Prof. Dr. Dominik Herrmann

Einführung in die Informatik (Inf-Einf-B)

Otto-Friedrich-Universität Bamberg

WiSe 2026/27

Heute

14:20	Montagsfragen auf Papier
danach	Rätsel aus Vorlesung 2
nach der Pause	Sortieren
Frist	Übung 2 heute fällig



live.inf.zone

Montagsfragen

So kreuzen Sie an

Pseudonym	Je Wort den Anfangsbuchstaben ankreuzen. Kein Name.
Fragen	3 Fragen, je 2 Minuten, jede Person für sich. Dann lösen wir auf.
Ankreuzen	Je Frage: gelöst oder nicht gelöst.
Vier Fragen dazu	Unten auf dem Bogen, zu Übung 2, Tutorium, Hilfen und Übung 3.
Keine Punkte	Sie sehen, ob Sie auf Kurs sind.

Frage 1

2 Minuten, jede Person für sich, ohne Laptop

Was gibt das Programm aus? Und was gibt es aus, wenn in der Bedingung statt 100 die 7 steht?

```
int numbers[] = {20, 500, 10, 5, 100, 1, 50};
int i;
for (i = 0; i < 7; i++)
{
    if (numbers[i] == 100)
    {
        break;
    }
}
printf("%i\n", i);
```

Frage 2

2 Minuten

Was gibt das Programm aus? Was gibt es bei n = 200 aus?

```c
int n = 100;
int schritte = 0;
while (n > 1)
{
    n = n / 2;
    schritte++;
}
printf("%i\n", schritte);
```

Frage 3

2 Minuten

Was gibt das Programm aus? Wie muss die Zeile mit schnitt lauten, damit 7.5 herauskommt?

```
typedef struct
{
    string name;
    int punkte;
}
student;

// in main:
student s[2];
s[0].name = "Ada";
s[0].punkte = 7;
s[1].name = "Alan";
s[1].punkte = 8;
float schnitt = (s[0].punkte + s[1].punkte) / 2;
printf("%s: %.1f\n", s[1].name, schnitt);
```

Frage 1

```c
int numbers[] = {20, 500, 10, 5, 100, 1, 50};
int i;
for (i = 0; i < 7; i++)
{
    if (numbers[i] == 100)
    {
        break;
    }
}
printf("%i\n", i);
```

Was gibt das Programm aus? Und mit der 7?

4, und mit der 7 dann 7.

Die 100 steht an Index 4: `break` verlässt die Schleife sofort, das `i++` läuft nicht mehr.

Die 7 kommt nicht vor. Die Schleife läuft ganz durch, bis `i` durch `i++` zu 7 wird und `i < 7` falsch ist.

Einen Index 7 gibt es in diesem Array nicht.

Frage 2

```c
int n = 100;
int schritte = 0;
while (n > 1)
{
    n = n / 2;
    schritte++;
}
printf("%i\n", schritte);
```

Was gibt das Programm aus? Und bei n = 200?

6, bei n = 200 dann 7.

n durchläuft 100, 50, 25, 12, 6, 3, 1: sechs Halbierungen. 25 / 2 ist als Ganzzahldivision 12.

Doppelt so großes n, genau ein Schritt mehr. Das ist O(log n), wie bei der binären Suche.

Frage 3

```
float schnitt = (s[0].punkte + s[1].punkte) / 2;
printf("%s: %.1f\n", s[1].name, schnitt);
```

Was gibt das Programm aus? Wie muss die Zeile mit `schnitt` lauten?

`Alan: 7.0`

`s[1].name` ist das Feld `name` des zweiten Elements,
also Alan. 7 + 8 = 15 und 2 sind beide `int`: 15 / 2 ist als
Ganzzahldivision 7, erst danach wird daraus der `float` 7.0.

Korrektur zum Beispiel `(s[0].punkte + s[1].punkte) / 2.0`,
oder ein Cast wie `(float)(s[0].punkte + s[1].punkte) /`
2. Dann rechnet C mit Kommazahl, und es kommt 7.5 heraus.

Kreuzen Sie an

Ergebnis	Je Frage: gelöst oder nicht gelöst.
Vier Fragen dazu	Unten auf dem Bogen. Auch ankreuzen.
Abgabe	Zum Gang durchgeben. Wer am Gang sitzt, bringt den Stapel in der Pause nach vorne.

Das Rätsel aus Vorlesung 2

Heute

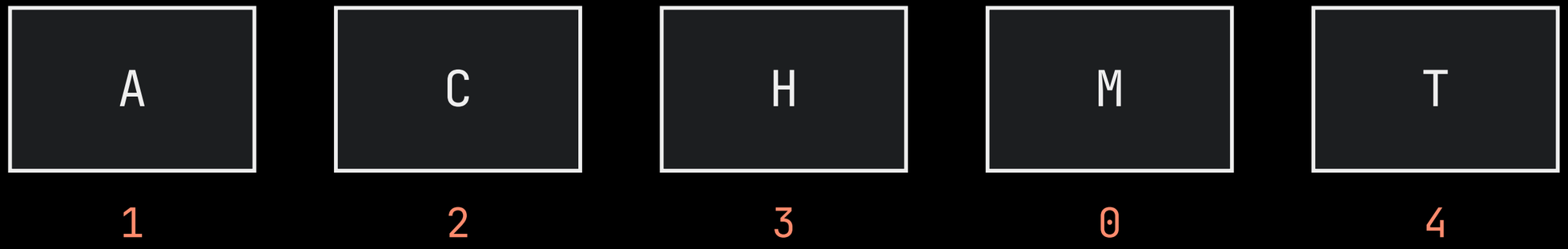
Rätsel	Der Geheimtext aus Vorlesung 2, von Hand gelöst.
Programm	Fertigen Code lesen und vorhersagen, was er tut.
Nach der Pause	Sortieren.

Handy bereit: live.inf.zone

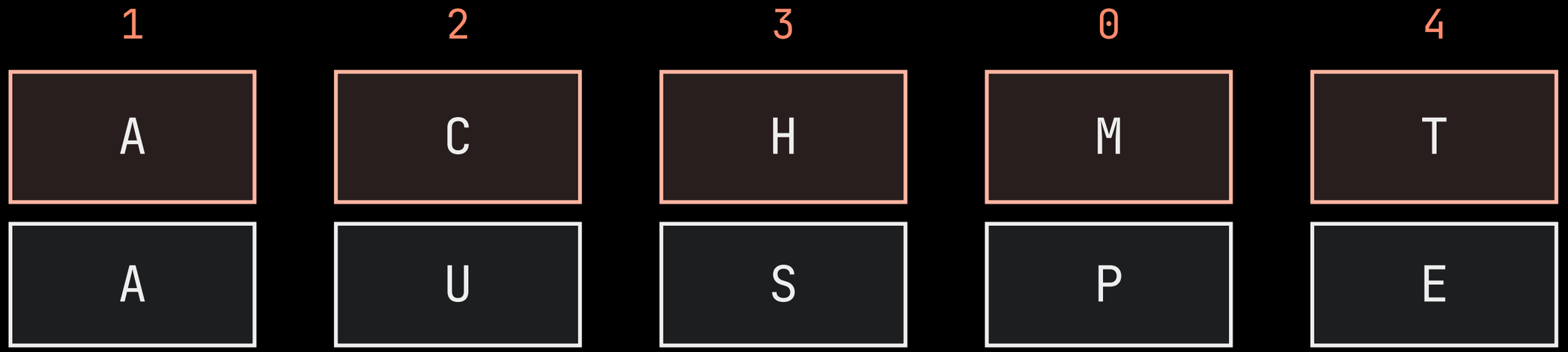
UCHBS ABETN ERDWE
NURNU SORMT ERTIX

Verschoben wie bei Cäsar?

Blöcke zu fünf. Was fehlt zum Zurücksortieren?



Schlüssel: 1 2 3 0 4



AUSPE

An Stelle i steht der Buchstabe von Stelle key[i].

Jetzt Sie: das Rätsel entschlüsseln

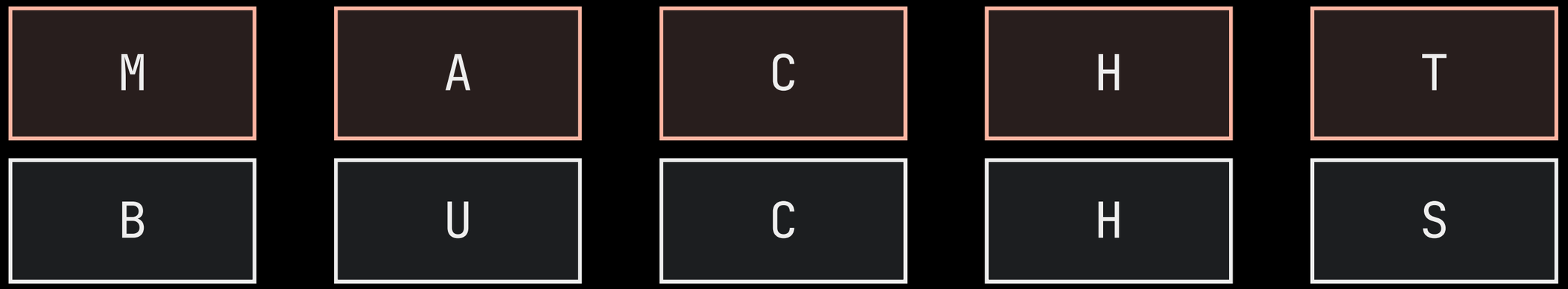
5 Minuten, auf dem Rätselzettel, Laptop zu. Wählen Sie die Stufe, die Sie schaffen.

Stufe 1 Entschlüsseln Sie die ersten beiden Blöcke: `UCHBS ABETN`.

Stufe 2 Entschlüsseln Sie den ganzen Geheimtext.

Stufe 3 Neues Schlüsselwort `BRAVO`, Geheimtext `TGEUG CMTAH`.
Bestimmen Sie den Schlüssel selbst und entschlüsseln Sie.

Hilfe: `A C H M T` über den Block, `M A C H T` über das leere Raster, dann die Spalten in diese Reihenfolge bringen.



BUCHS

Der ganze Satz

```
BUCHS TABEN WERDE NNURU MSORT IERTX
```

BUCHSTABEN WERDEN NUR UMSORTIERT, das X füllt den letzten Block auf.

Stufe 3: `BRAVO` sortiert ergibt `A B O R V`, also den Schlüssel `2 0 4 1 3`, und aus `TGEUG CMTAH` wird GUT GEMACHT.

Vom Papier zum Programm

Was muss das Programm können?

Lesen	Schlüssel und Text von der Kommandozeile lesen
Umwandeln	Schlüssel: Zeichen → Zahlen
Zerlegen	Text in Blöcke zerlegen
Umsortieren	einen Block umsortieren und ausgeben
Prüfen	Eingaben prüfen
Aufräumen	Funktionen

Drei Funktionen

```
bool parse_key(string s, int key[]);
bool is_permutation(int key[], int n);
void print_block(string text, int start, int key[], int n);
```

parse_key	Schlüssel: Zeichen → Zahlen
is_permutation	Schlüssel prüfen: jede Stelle genau einmal
print_block	einen Block umsortieren und ausgeben

`main` als Gerüst


```
int main(int argc, string argv[])
{
    // [ausgeblendet: Aufruf prüfen]              -> Station 1
    int n = strlen(argv[1]);
    int key[MAX_KEY];
    // [ausgeblendet: Schlüssel lesen und prüfen]
    string text = argv[2];
    int len = strlen(text);
    // [ausgeblendet: Länge prüfen]               -> Station 2
    for (int start = 0; start < len; start += n) // -> Station 3
    {
        print_block(text, start, key, n);
    }
    printf("\n");
}
```

`parse_key` füllt `key` mit `key[i] = s[i] - '0';`. Was stünde bei `./permute 201 VSCODE` ohne `- '0'` in `key[0]`?

50, der ASCII-Code von `'2'`: Erst `s[i] - '0'` in `parse_key` macht aus dem Zeichen die Ziffer.

Das Programm, das das Rätsel gemacht hat

```
$ ./permute 12304 BUCHSTABENWERDENNURUMSORTIERTX
UCHBSABETNERDWENURNUSORMTERTIX
```

Genau der Geheimtext aus Vorlesung 2.

Drei Stationen

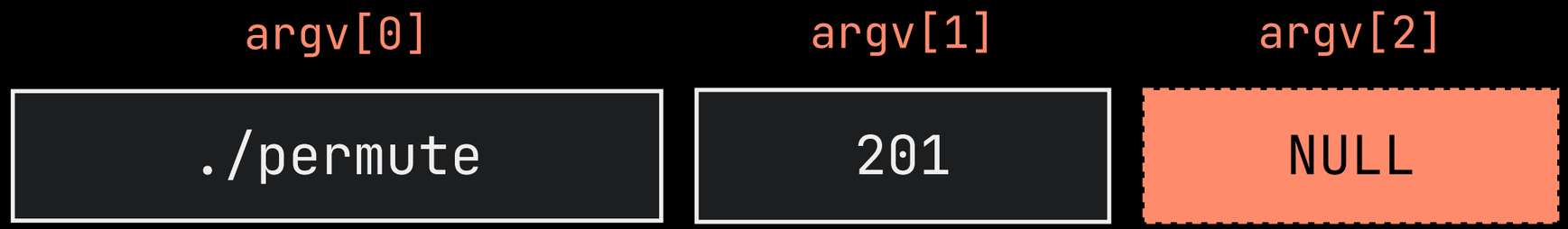
Station 1: Ohne Prüfung von `argc`

```
int main(int argc, string argv[])
{
    int n = strlen(argv[1]);
    …
    string text = argv[2];
    int len = strlen(text);
```



Das Programm wird mit `./permute 201` aufgerufen. Welchen Wert hat `argc`?

Auflösung: `argc` ist 2




```
$ ./permute-argc 201
Segmentation fault
$ echo $?
139
```

Die Prüfung des Aufrufs


```c
if (argc != 3)
{
    printf("Usage: ./permute key text\n");
    return 1;
}
```

```
$ ./permute 201; echo $?
Usage: ./permute key text
1
```

Station 2: Ohne Prüfung der Länge

```
    string text = argv[2];
    int len = strlen(text);
    // Wie viele Blöcke? 7 Zeichen: 7 / 3 = 2, der Rest fehlt -> einer mehr
    int blocks = len / n + 1;
    for (int b = 0; b < blocks; b++)
    {
        print_block(text, b * n, key, n);
    }
    printf("\n");
    return 0;
}
```

`./permute-bloecke 201 VSCODE`: Was steht in `text[6]`?

A

`'E'`

B

`'\0'`

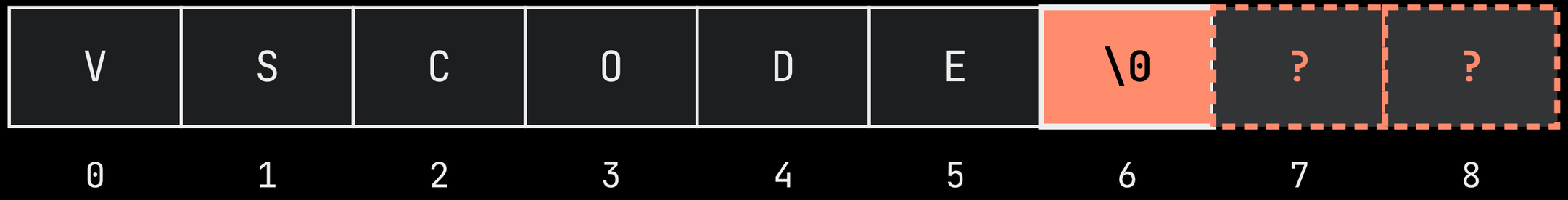
C

nichts, der Compiler meldet einen Fehler

D

ein zufälliges Zeichen

len = 6, also 6 / 3 + 1 = 3 Blöcke



dritter Block: Stellen 8, 6, 7

```
$ ./permute-bloecke 201 VSCODE | cat -v
CVSEODI^@H
```

Die Prüfung der Länge

```
if (len % n != 0)
{
    printf("Text length (%i) must be a multiple of key length (%i).\n",
           len, n);
    return 1;
}
```

```
$ ./permute 201 VSCODEX
Text length (7) must be a multiple of key length (3).
```

Auch der Schlüssel 205 liest hinter das Ende.

Station 3: Weiterrücken in einer Funktion

```c
int start = 0;
while (start < len)
{
    print_block(text, start, key, n);
    next_block(start, n);
}

// zum nächsten Block weiterrücken
void next_block(int start, int n)
{
    start = start + n;
}
```

Was gibt `./permute-weiter 201 VSCODE` aus?

A

`CVSEOD`

B

`CVS`, immer wieder, bis Strg+C

C

`CVS`, dann `Segmentation fault`

D

nichts, `make` meldet: `start` gibt es schon

Huch, warum hört das nicht auf?

```
$ ./permute-weiter 201 VSCODE
CVSCVSCVSCVSCVSCVSCVSCVSCVSCVSCVSCVSCVSC…
```

Warum `start` in `main` 0 bleibt: nach der Pause.

Holen oder bringen

Welche Fassung entschlüsselt?

```
// A: holen (permute.c)
        printf("%c", text[start + key[i]]);

// B: bringen (permute-bringen.c)
        out[key[i]] = text[start + i];
    …
        printf("%c", out[i]);
```

Beide Fassungen bekommen denselben Schlüssel `12304`. Welche macht aus `UCHBS` wieder `BUCHS`?

A

Fassung A (holen)

B

Fassung B (bringen)

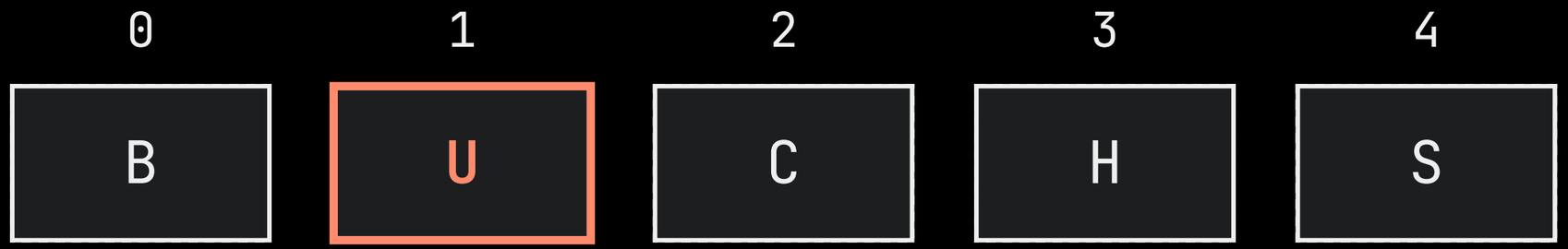
C

beide

D

keine, mit `12304` kommt man nicht zurück


A
C
H
M
T
1
2
3
0
4


0
1
2
3
4
B
U
C
H
S

Der ganze Geheimtext zurück

```
$ ./permute-bringen 12304 UCHBSABETNERDWENURNUSORMTERTIX
BUCHSTABENWERDENNURUMSORTIERTX
```

Es gibt einen zweiten Weg zurück. Der steht in Video 1.

Ausblick

Wie man so ein Programm schreibt

Video 1	Vor dem leeren Editor anfangen, Schritt für Schritt bauen und testen. Und der zweite Weg zurück.
Video 2	Falsche Eingaben abfangen: die Prüfungen von heute.
Video 3	Wie aus MACHT der Schlüssel 12304 wird.
Übung 3	Cäsar, Abgabe bis Montag, 16. November.

Wie sortiert der Rechner MACHT?

Pause · 20 Minuten

Aus dem letzten Jahrgang

2 von 10 wünschten sich eine Lerngruppe und hatten keine. Ins Tutorium ging etwa 1 von 10.

Probieren Sie erst selbst. Mit dem, was nicht läuft, ins Tutorium.