Übung zur Vorlesung Informatik I WS 2024/25
Fakultät für Angewandte Informatik
Institut für Informatik
Prof. Dr. J. Hähner, J. Linne, H. Cui, V. Gerling, N. Kemper

Übungsgruppe 69

Abgabe des 4. Übungsblatts

Erik Wiedmann, Marwin Merkl, Manuel Henker

Aufgabe 29

Datei: temperatur.h

```c
#ifndef TEMPERATURE_H
#define TEMPERATURE_H

#include <stdio.h>
#include <stdlib.h>

#define MAX_CELSIUS -273.15
#define MIN_CELSIUS 100
#define SUCCESS 0
#define ERROR -1

#define CELSIUS_TO_FAHRENHEIT(celsius) ((celsius) *1.8 + 32)

void print_temp_in_fahrenheit(double *temperatures, int size);
int input_temperature(double *temperatures, int *size, int
    max_size);
int connect_temperature_arrays(double *temps1, int *temps1_size,
    int max_size, double *temps2, int temps2_size);

#endif
```

Datei: temperatur.c

```c
#include "temperature.h"

void print_temp_in_fahrenheit(double *temperatures, int  size) {
    int i;
    for (i = 0; i < size; i++) {
            printf("Temperatur %f C zu: %.2f F\n", temperatures[
                i], CELSIUS_TO_FAHRENHEIT(temperatures[i]));
    }
}

int input_temperature(double *temperatures, int *size, int
    max_size) {
    double input;

    if (*size >= max_size) {
            printf("Fehler: Maximale Kapazitaet des Arrays
                erreicht.\n");
```

```
            return ERROR;
        }

        printf("Geben Sie eine Temperatur in Grad Celsius ein (%d bis %.2f): ", MIN_CELSIUS, MAX_CELSIUS);

        if (scanf("%lf", &input) != 1 || input < MIN_CELSIUS || input > MAX_CELSIUS) {
            printf("Ungueltige Eingabe! Bitte eine gueltige Temperatur eingeben.\n");
            return ERROR;
        }

        temperatures[*size] = input;
        (*size)++;
        return SUCCESS;
}
int connect_temperature_arrays(double *temps1, int *temps1_size, int max_size, double *temps2, int temps2_size) {
    int i;

    if (*temps1_size + temps2_size > max_size) {
            printf("Fehler: Ziel-Array hat nicht genuegend Kapazitaet.\n");
            return ERROR;
        }

        for (i = 0; i < temps2_size; i++) {
            temps1[*temps1_size + i] = temps2[i];
        }

        *temps1_size += temps2_size;
        return SUCCESS;
}
```

Datei: 29main.c

```
#include "temperature.h"

int main(void) {
    int max_size = 10;
        double temperatures[10];
    int *size = 0;

    double second_temperatures[] = {25.5, -100.0};
    int second_size = 2;

        if (input_temperature(temperatures, size, max_size) == SUCCESS) {
            printf("Temperatur erfolgreich hinzugefuegt.\n");
        }

        printf("Alle Temperaturen in Fahrenheit:\n");
        print_temp_in_fahrenheit(temperatures, *size);

        if (connect_temperature_arrays(temperatures, size, max_size, second_temperatures, second_size) == SUCCESS)
```

```
            {
                printf("Arrays erfolgreich zusammengeführt.\n");
        } else {
        printf("Das 2. Array ist zu lang um es an das 1. Array
            anzufuegen!");
    }

        printf("Aktualisierte Temperaturen in Fahrenheit:\n");
        print_temp_in_fahrenheit(temperatures, *size);

    return 0;
}
```

Aufgabe 30

Datei: 30.h

```
#ifndef DREIZIG_H
#define DREIZIG_H

#include<stdio.h>

#define ISPRIME(x) printf("\nDIGAAAAA %i ist DIE Primzahl", x)
#define ABSTANT(a,b) (a > b) ? a - b : b - a

int prim(int a);
int quad(int q);

#endif
```

Datei: 30.c

```
#include "30.h"
#include<stdio.h>


int prim(int a){
    int i;
    for(i = 2; i * i <= a; i++){
        if(a % i  == 0){
            return 0;             /*flase*/
        }
    }
    return 1;                     /*true*/
}

int quad(int a){
    int i;
    for(i = 0; i < a; i++){
        if((i * i) == a){
            return 1;              /*true*/
        }
    }
    return 0;                      /*false*/
}

int main(void){
```

```
    int a, b, n;
    n = 0;

    for(a = 1000; a <= 1100; a++){
        if(prim(a)){
             ISPRIME(a);

            if(n > 0){
            printf("\nabstand zwischen %i und %i ist: %i", n, a,
                ABSTANT(a, n));
            }
        n = a;
        }

    }

    for(b = 1000; b <= 2000; b++){
        if(quad(b)){
            printf("\nBoa %i ist übelst die Quadrahtzahl", b);
        }

    }
    return 0;
}
```

Aufgabe 31

Datei: 31.c

```
#include <stdio.h>

unsigned long int fibonacci();
unsigned long int next(void);
void start(unsigned long int seed);

static unsigned long int current = 0;

int fibstop = 0;

int main () {
    static unsigned long int loops = 0;

    unsigned long int seed;
    int loopthreenplus1 = 0;

    static unsigned long int v1;
    static unsigned long int v2;
    static unsigned long int v3;

    fibstop = 50;

    fibonacci();

    printf("ab 48 unsigned long int limit reached. fib(48), fib
        (49) und fib(50) stimmen also nicht \n\n");

```

```
27      for (seed = 1; seed <= 10; seed++) {
28          printf("3n+1-Folge fuer: %i \n", seed);
29          start(seed);
30
31          v1 = 0;
32          v2 = 0;
33          v3 = 0;
34
35          while (1) {
36              if (loops == 0) {
37                  printf("%lu \n", current);
38              } else {
39                  printf("%lu \n", next());
40
41                  if (loopthreenplus1 % 3 == 0) {
42                      v1 = current;
43                      loopthreenplus1++;
44                  } else if (loopthreenplus1  % 3 == 1) {
45                      v2 = current;
46                      loopthreenplus1++;
47                  } else {
48                      v3 = current;
49                      loopthreenplus1 = 0;
50                  }
51              }
52
53              if ((v1 == 4 && v2 == 2 && v3 == 1) || (v1 == 1 && v2
                    == 4 && v3 == 2) || (v1 == 2 && v2 == 1 && v3 == 4)
                    ) {
54                  printf("4,2,1 loop reached \n\n");
55                  break;
56              }
57
58              loops++;
59          }
60          loops = 0;
61      }
62      return 0;
63  }
64
65  unsigned long int fibonacci(void) {
66      static unsigned long int last = 0, current = 0;
67      int n = 0, temp = 0;
68      while(n >= 0) {
69          if (n == fibstop + 1) {
70              return current;
71          } else if (n == 0) {
72
73          } else if (n == 1) {
74              current++;
75          } else {
76              temp = current;
77              current = last + current;
78              last = temp;
79          }
80          printf(" %i = %lu \n", n, current);
```

```
        n++;
    }
    return current;
}

void start(unsigned long int seed) {
    current = seed;
}

unsigned long int next(void) {
    if (current % 2 == 0) {
            current = current / 2;
    } else {
        current = 3 * current + 1;
    }
    return current;
}
```

Aufgabe 32

a

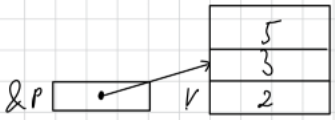
1. Datei: 32a1.c
```
int main(void){
    char w[3];
    int *p = &w[2];
}
```
2. Datei: 32a2.c
```
int main(void){
    int **p;
}
```
3. Datei: 32a3.c
```
int ** p;

int main(void){}
```
4. *v + 2 = 3
5. +(v + 2) = -1
6. Datei: 32a6.c
```
int main(void){
    int n;
    int *p = &n;
}
```
7. *(p++) = 6
8. ++(*p) = 7
9. *(++p) = 6
10. Mar
11. lade
12. ++(*p); -> (*p)++;
13. char w[] = v -> char w[strlen(v)]; strcpy(w, v);
14. *p = '5' -> p = '5'

15. *p = v -> *p = &v

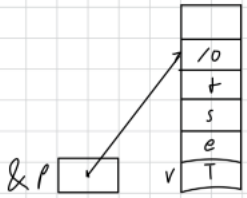
b)

32

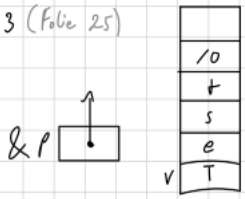
b) 1.



2.



3 (Folie 25)



c)

Anweisung	Typ
p = *a	char *

c)

Anweisung	Typ
p = *a	char *
p = a+1	char
p = &a	char *
p = **a	char **
p = a[0]	char
p = *a[0]	char *
p = &a[1]	char *