Aufgabenblatt 4: Schleifen

Florian Ludewig (Übungsgruppe 2)

November 19, 2019

Aufgabe 1 – Fröhliche Zahlen

```c
#include <stdio.h>
#include <math.h>

int is_unhappy_sum(int number)
{
  int unhappy_sums[8] = {20, 4, 16, 37, 58, 89, 145, 42};
  for (int i = 0; i < 8; i++)
  {
    if (unhappy_sums[i] == number)
      return 1;
  }
  return 0;
}

int calculate_sum(int number)
{
  int sum = 0;
  while (number > 0)
  {
    int digit = number % 10;
    number = number / 10;
    sum += digit * digit;
  }
  return sum;
}

int is_happy_number(int number)
{
  while (1)
  {
    int sum = calculate_sum(number);
    if (is_unhappy_sum(sum))
      return 0;
    if (sum == 1)
      return 1;
    number = sum;
  }
}

int main(void)
{
  for (int i = 1; i <= 500; i++)
  {
    if (is_happy_number(i))
      printf("%d\n", i);
  }
  return 0;
}
```

Aufgabe 2 – Berechnung von Pi

```c
#include <stdio.h>
#include <math.h>

int main()
{
  double accuracy;
  printf("Genauigkeit eingeben: ");
  scanf("%lf", &accuracy);

  double pi = 0;
  int n = 0;
  double sum = 0;

  while (1)
  {
    double numerator = n % 2 == 0 ? 1 : -1;
    double denominator = 2 * n + 1;
    double summand = numerator / denominator;

    double new_sum = sum + summand;

    if (fabs((sum * 4) - (new_sum * 4)) < accuracy)
      break;

    sum = new_sum;
    n++;
  }
  pi = sum * 4;

  printf("Pi nach %d Iterationen: %f\n", n, pi);
  printf("Abweichung: %f\n", fabs(pi - M_PI));

  return 0;
}
```

Aufgabe 3 – Primfaktorzerlegung

```c
#include <stdio.h>
#include <math.h>

int is_prime_number(int number)
{
  if (number == 2)
    return 1;
  if (number % 2 == 0 || number % 5 == 0)
    return 0;
  int biggest_possible_divider = floor(sqrt(number));
  for (int i = 2; i <= biggest_possible_divider; i++)
  {
    if (number % i == 0)
      return 0;
  }
  return 1;
}

int smallest_prime_divider(int number)
{
  int i = 2;
  while (i < number)
  {
    if (is_prime_number(i) && number % i == 0)
      return i;
    i++;
  }
  return number;
}

int main(void)
{
  int input;
  printf("Bitte gib eine Zahl ein: ");
  scanf("%d", &input);

  int factors[100];

  int iteration = 0;
  while (input > 1)
  {
    int divider = smallest_prime_divider(input);
    factors[iteration] = divider;
    input /= divider;
    iteration++;
  }

  for (int i = 0; i < iteration; i++)
  {
    if (i == iteration - 1)
      printf("%d\n", factors[i]);
    else
      printf("%d * ", factors[i]);
  }
  return 0;
}
```