

9 Quest

9.1 Aufgabe

Compilieren mit: `gcc -lm -o myapp Blatt-09-A1.c`

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

char* find(char *sp, char c)
{
    while (*sp != c)
    {
        if (*sp == '\0')
            return NULL;
        sp++;
    }
    return sp;
}

int add(float a, float b, float c, float d, float* e, float* f)
{
    *e = a + c;
    *f = b + d;
}

int multiply(float a, float b, float c, float d, float* e, float* f)
{
    *e = a * c - b * d;
    *f = a * d + b * c;
}

int divide(float a, float b, float c, float d, float* e, float* f)
{
    *e = (a * c + b * d) / (c * c + d * d);
    *f = (b * c - a * d) / (c * c + d * d);
}

int main(int argc, char **argv[])
{
    float Rex, Imx, Rey, Imy, Reres, Imres, r, phi;
    char op, menu;

    while ( 1 )
    {
        printf("Menu\n1: Berechnung von x op y\n");
        printf("2: Umrechnung von x in Vektorform\n");
        printf("x: Programm beenden\n\n");
    }
}
```

```

printf(" Auswahl: \n");
scanf("\n%c", &menu);
if (menu == '1')
{
    printf(" Berechnung von x op y \n");
    printf(" — x, y komplex, Operationen: + - * /\n");

    printf(" Eingabe (a+bi) op (c+di): \n");
    scanf("\n(%f+ %f i) %c (%f+ %f i)", &Rex, &Imx, &op, &Rey, &Imy);

    switch ( op )
    {
        case '+': add(Rex, Imx, Rey, Imy, &Reres, &Imres); break;
        case '-': add(Rex, Imx, -Rey, -Imy, &Reres, &Imres); break;
        case '*': multiply(Rex, Imx, Rey, Imy, &Reres, &Imres); break;
        case '/': divide(Rex, Imx, Rey, Imy, &Reres, &Imres); break;
        default: printf(" invalid operator");
    }

    printf("( %f+ %f i) %c (%f+ %f i) = (%f+ %f i)\n",
           Rex, Imx, op, Rey, Imy, Reres, Imres);
}
else
    if (menu == '2')
    {
        printf(" Eingabe (a+bi): \n");
        scanf("\n(%f+ %f i)", &Rex, &Imx);
        r = (float) sqrt(Rex * Rex + Imx * Imx);
        if ( r > 0 )
            phi = (float) acos(Rex/r);
        else if ( r < 0 )
            phi = (float) -acos(Rex/r);
        else
            phi = 0;
        printf("( %f+ %f i) = %f * exp(%f i)\n\n", Rex, Imx, r, phi);
    }
else
    if (menu == 'x')
        exit(0);
    else
        printf(" Ungueltige Eingabe\n\n");
}
}

```

9.2 Aufgabe

9.2.1 FIFO

Die Zahl der Seitenfehler beträgt 12:

S	1	2	3	4	5	6	2	4	3	1	7	2	3	8	2	1	7	6	7	2
	1	2	3	4	5	6	6	6	6	1	7	2	3	8	8	8	8	6	6	6
		1	2	3	4	5	5	5	5	6	1	7	2	3	3	3	3	8	8	8
			1	2	3	4	4	4	4	5	6	1	7	2	2	2	2	3	3	3
				1	2	3	3	3	3	4	5	6	1	7	7	7	7	2	2	2
					1	2	2	2	2	3	4	5	6	1	1	1	1	7	7	7
F	P	P	P	P	P	P	-	-	-	P	P	P	P	P	-	-	-	P	-	-

9.2.2 LRU

Die Zahl der Seitenfehler beträgt 10:

S	1	2	3	4	5	6	2	4	3	1	7	2	3	8	2	1	7	6	7	2
	1	2	3	4	5	6	2	4	3	1	7	2	3	8	2	1	7	6	7	2
		1	2	3	4	5	6	2	4	3	1	7	2	3	8	2	1	7	6	7
			1	2	3	4	5	6	2	4	3	1	7	2	3	8	2	1	1	6
				1	2	3	4	5	6	2	4	3	1	7	7	3	8	2	2	1
					1	2	3	3	5	6	2	4	4	1	1	7	3	8	8	8
						1	1	1	1	5	6	6	6	4	4	4	4	3	3	3
											5	5	5	6	6	6	6	4	4	4
														5	5	5	5	5	5	5
F	P	P	P	P	P	P	-	-	-	P	P	-	-	P	-	-	-	P	-	-
D	∞	∞	∞	∞	∞	∞	5	4	5	6	∞	5	4	∞	3	5	5	7	2	4

9.3 Aufgabe

- Seite 2 bei NRU
- Seite 3 bei FIFO
- Seite 1 bei LRO
- Seite 2 bei Second-Chance

9.4 Aufgabe

0 - 300	1 - 8	1 - 12	2 - 300	3 - 450	4 - 100	4 - 96
225	904	908	Fehler	6F1	Fehler	Fehler