

Programmieren

Sammlung gegliedert nach Modul

Fabian Suter, 19. Dezember 2023

<https://github.com/FabianSuter/Programmieren.git>

1 ProgC

1.1 Wichtige Kurzbefehle

cd "Path"	Pfad anwählen
cd ..	um eine Ebene nach oben (zurück)
mkdir "Ordnername"	Ordner erstellen
rmkdir "Ordnername"	Ordner löschen
rm -rf *	Alles innerhalb vom aktuellen Ordner löschen
rm "Datei"	Datei löschen
mv "Name alt" "Name neu"	Datei umbenennen
cp "Datei alt" "Datei neu"	Datei kopieren und benennen
clang -Wall -o "Outputname" "Inputdatei"	clang-Compiler mit Warnungen
clang -Wall -o "Outputname" "Inputdatei" -lm	-lm für Mathebibliothek
ls	Listet alle Files im akt. Verzeichnis auf
ls -l	Inkl. Informationen wie Grösse u.a.
ls -a	Inkl. versteckten Dateien
ls -al	Beide Varianten

1.2 Zahlensysteme

2 ⁰ = 1	2 ¹ = 2	2 ² = 4	2 ³ = 8	2 ⁴ = 16	2 ⁵ = 32	2 ⁶ = 64	2 ⁷ = 128
Grösse	Abk.	Genauer Wert				Näherung	
Kilobyte	kB	2 ¹⁰ = 1024 Bytes				10 ³ Bytes	
Megabyte	MB	2 ²⁰ = 1 048 576 Bytes				10 ⁶ Bytes	
Gigabyte	GB	2 ³⁰ = 1 073 741 824 Bytes				10 ⁹ Bytes	
Terabyte	TB	2 ⁴⁰ = 1 099 511 627 776 Bytes				10 ¹² Bytes	
Oktal	3 Bits	X ₈	X _O	X _q	X _{oct}	0X	
Hex	4 Bits	X ₁₆	X _h	X _H	X _{hex}	0xX	

Hexadezimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

ASCII (7-Bit) Ordnet gängigen Schriftzeichen einen Zahlenwert zu, um diese in einem Digitalrechner präsentieren zu können. Die Tabelle ist wichtig, um für geg. Schriftzeichen den repräsentierten Zahlenwert zu ermitteln (und umgekehrt).

Nachfolger: Unicode (8-, 16-, 32-Bit)

1.3 Datentypen und Variablen

1.3.1 Datentypen

Name	Gebrauch	Grösse
Ganzzahltypen		
char	Buchstaben, Zahlen	8 Bit (1 Byte)
short	kleine, ganzz. Werte	abh. (min 16bit)
int	eff. Grösse des Prozessors	abh. (min 16bit)
long	grosse ganzz. Werte	abh. (min 32bit)
long long	sehr grosse ganzz. Werte	abh. (min 64bit)
Gleitpunkttypen		
float	Gleitpunkt, single precision	abh.
double	Gleitpunkt, high precision, Standard	abh.
long double	Gleitpunkt, higher precision	abh.
Platzhalter		
char	%c	l, ll > Typ länger als ein int
Zeichenkette	%s	h, hh > Typ kürzer als ein int
Signed Ganzzahl dez.	%d	
Unsigned Ganzzahl dez.	%u	
Unsigned Ganzzahl okt.	%o	
Unsigned Ganzzahl hex.	%x	
Ganzzahlen können überlaufen!		
Gleitpunktzahlen haben meist Rundungsfehler. Nie auf Gleichheit prüfen!		

1.3.2 Namen

1.3.3 Wertebereich

unsigned	0...($2^n - 1$)	n=8 : 0...255
signed	$-2^{n-1} \dots + (2^{n-1} - 1)$	n=8 : -128...+127

1.4 Schleifen

1.4.1 For-Schleife

for (Ausdruck_init; solange Ausdruck; Ausdruck_update) Anweisung								
<table border="1"><tr><td>Ausdruck_init</td></tr><tr><td>solange Ausdruck</td></tr><tr><td><table border="1"><tr><td>Anweisung</td></tr><tr><td>Ausdruck_update</td></tr></table></td></tr></table>	Ausdruck_init	solange Ausdruck	<table border="1"><tr><td>Anweisung</td></tr><tr><td>Ausdruck_update</td></tr></table>	Anweisung	Ausdruck_update	entspricht	<pre>Ausdruck_init; while (solange Ausdruck) { Anweisung Ausdruck_update }</pre>	For-Schleife
Ausdruck_init								
solange Ausdruck								
<table border="1"><tr><td>Anweisung</td></tr><tr><td>Ausdruck_update</td></tr></table>	Anweisung	Ausdruck_update						
Anweisung								
Ausdruck_update								

1.4.2 Switch-Schleife

1.4.3 Do-While-Schleife

1.4.4 While-Schleife

1.4.5 Sprunganweisungen

1.5 Code-Snippets

1.5.1 Array und Pointer

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

```
int main(){
    enum{array_size = 6};
```

```

    int test[array_size] = {1,2,3,4,5,6};
    for(int i =0; i<array_size; ++i)
        printf("Element %u: %i\n", i, test[i]);

    printf("Groesster: %d", *findAbsMax(test, array_size));
    return 0;
}

```

Main-Funktion zum Finden eines **betragsmässig** grössten Wertes innerhalb eines Arrays.

```

int* findAbsMax(int* arr, size_t size){
    int* max_ptr = &arr[0];
    for(size_t i = 0; i < size; ++i){
        if((arr[i] >=0 && *max_ptr >=0 && arr[i] > *max_ptr)
            || (arr[i] <=0 && *max_ptr <=0 && arr[i] < *max_ptr)
            || (arr[i] >=0 && *max_ptr <=0 && arr[i] > *max_ptr * -1)
            || (arr[i] <=0 && *max_ptr >=0 && arr[i] * -1 > *max_ptr))
            max_ptr = &arr[i];
    }
    return max_ptr;
}

```