

Matrikelnummer: 1750251



Fakultät: Technik
Studiengang: Informatik
Studienrichtung: IT-Automotive
Jahrgang/ Kurs: 2024 / 24ITA
Semester: 1. Semester

KLAUSURDECKBLATT

Datum:	26.02.2025	Bearbeitungszeit:	200 Min.
Modul:	T4INF1004	DozentIn:	Bohl
Unit:	Programmieren		
Hilfsmittel:	alles, außer elektronische Medien		
Punkte:	_____	Note:	_____

Bitte dokumentenechte Stifte verwenden!
Bitte die Blätter nicht aus der Heftung nehmen!

1. Sind Sie gesund und prüfungsfähig?
2. Sind Ihre Taschen und sämtliche Unterlagen, insbesondere alle nicht erlaubten Hilfsmittel, seitlich an der Wand zum Gang hin abgestellt und nicht in Reichweite des Arbeitsplatzes?
3. Haben Sie auch außerhalb des Klausorraumes im Gebäude keine unerlaubten Hilfsmittel oder ähnliche Unterlagen liegen lassen?
4. Haben Sie Ihr/e Smartphone /-watch ausgeschaltet und abgegeben?
5. Alle Klausurblätter sind abzugeben, keine Klausuraufgaben werden abgeschrieben!

(Falls Ziff. 2 oder 3 nicht erfüllt sind, liegt ein Täuschungsversuch vor, der die Note „nicht ausreichend“ zur Folge hat.)

Im Rahmen des Projektes soll ein Programm erstellt werden, dass eine Bitmap Datei lädt und das geladene Bitmap mit verschiedenen Filtern verändert. Zusätzlich soll das Programm die Möglichkeit bieten, eine Übersicht der Bitmap Informationen auszugeben.

Die Filter sollen jeweils einzeln, gesteuert durch die Kommandozeile, ausgeführt werden.

z.B.:

```
c:/> convert bild.bmp neu.bmp invert [blau|rot|gruen]
```

```
c:/> convert bild.bmp neu.bmp rotate
```

```
c:/> convert bild.bmp info
```

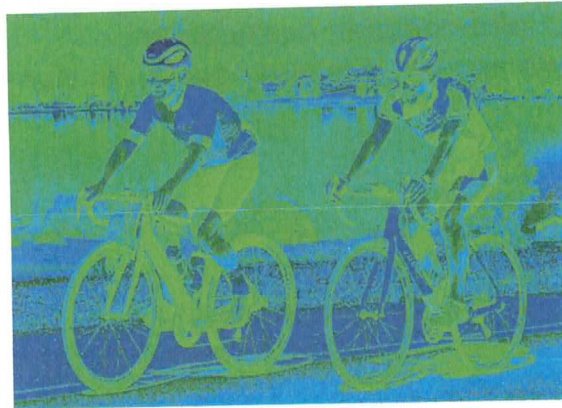
Bitmap: bild.bmp



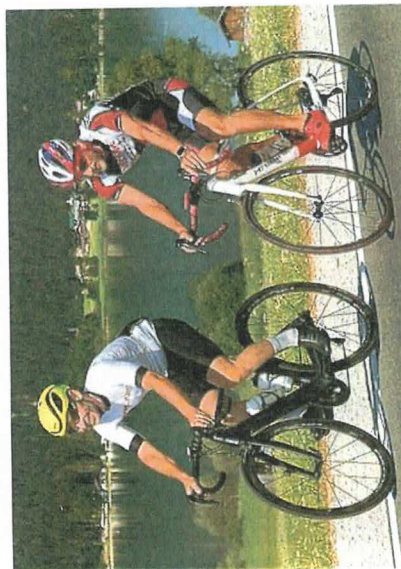
Ein Filter ist in dem Zusammenhang z.B. eine Rotation des Bilds um 90° oder verwischen der Schärfe, das Spiegeln des Bildes an der X- oder Y-Achse oder das Invertieren bzw. Abschwächen der Farbe.

Um es einfacher zu gestalten, sollen Sie zwei dieser Filter und die Ausgabe der Bild Info in C umsetzen.

Zum einen wäre das ein Filter zum Invertieren der Farben, wobei man die Farbe, welche invertiert werden soll, auswählen können soll:



Und zum anderen ein Filter um das Bild um 90° gegen den Uhrzeigersinn zu drehen:



Die Bild Info muss folgende tabellarische Ausgaben haben, wobei die Werte selbstverständlich aus der Bitmap Datei kommen müssen:

```
Bild Info:
biCompression:  BI_RGB
biBitCount:     16 Bit
biHeight:       768 px
biWidth:        1024 px
biSizeImage:    1653296 byte
```


BMP-Dateien bestehen aus drei Teilen:
Dem Dateikopf, dem Informationsblock und den Bilddaten.



BITMAPFILEHEADER (Größe: 14 Byte)						
Offset (Byte)		Datentyp		Größe	Name	Inhalt
Dez	Hex	Windows-Style	C-Style			
0	0	WORD	uint16_t	2 Byte	bfType	ASCII-Zeichenkette "BM" (Hex: 0x42 0x4D, Dezimal: 66 77).
2	2	DWORD	uint32_t	4 Byte	bfSize	Größe der BMP-Datei in Byte. (unzuverlässig)
6	6	DWORD	uint32_t	4 Byte	bfReserved	Reserviert, von der Software abhängig, standardmäßig 0
						Offset der Bilddaten in Byte vom Beginn der Datei an.
10	A	DWORD	uint32_t	4 Byte	bfOffBits	Dieser hat nicht immer den Wert 54 (er ergibt sich aus 14 Byte Header + 40 Byte Infoblock) und muss daher dynamisch ausgelesen werden, weil es sonst ggf. zu Fehldarstellungen im Bild kommt.

BITMAPINFOHEADER (Größe: 40 Byte)						
Offset (Byte)		Datentyp		Größe	Name	Inhalt
Dez	Hex	Windows-Style	C-Style			
14	0E	DWORD	uint32_t	4 Byte	biSize	Größe der BITMAPINFOHEADER-Struktur in Byte
18	12	LONG	int32_t	4 Byte	biWidth	Breite der Bitmap in Pixel. Dabei ist das erste Byte niederwertig und das letzte Byte höchstwertig. Der Betrag gibt die Höhe der Bitmap in Pixel an. Dabei ist das erste Byte niederwertig und das letzte Byte höchstwertig.
22	16	LONG	int32_t	4 Byte	biHeight	Ist der Wert positiv, so ist die Bitmap eine sogenannte "bottom-up"-Bitmap (die Bilddaten beginnen mit der untersten und enden mit der obersten Bildzeile). Dies ist die gebräuchlichste Variante. Ist der Wert negativ, so ist die Bitmap eine "top-down"-Bitmap (die Bilddaten beginnen mit der obersten und enden mit der untersten Bildzeile).
26	1A	WORD	uint16_t	2 Byte	biPlanes	1 (Stand in einigen älteren Formaten wie PCX für die Anzahl der Farbebenen, wird aber für BMP nicht verwendet)
28	1C	WORD	uint16_t	2 Byte	biBitCount	Gibt die Farbtiefe der Bitmap in bpp an; muss einer der folgenden Werte sein: 1, 4, 8, 16, 24 oder 32. Bei 1, 4 und 8 bpp sind die Farben indiziert.
						Einer der folgenden Werte:
30	1E	DWORD	uint32_t	4 Byte	biCompression	<ul style="list-style-type: none"> 0 (BI_RGB): Bilddaten sind unkomprimiert. 1 (BI_RLE8): Bilddaten sind lauffängenkodiert für 8 bpp. Nur erlaubt wenn biBitCount=8 und biHeight positiv. 2 (BI_RLE4): Bilddaten sind lauffängenkodiert für 4 bpp. Nur erlaubt wenn biBitCount=4 und biHeight positiv. 3 (BI_BITFIELDS): Bilddaten sind unkomprimiert und benutzerdefiniert (mittels Farbmasken) kodiert. Nur erlaubt wenn biBitCount=16 oder 32.
34	22	DWORD	uint32_t	4 Byte	biSizeImage	<ul style="list-style-type: none"> Wenn biCompression=BI_RGB: Entweder 0 oder die Größe der Bilddaten in Byte. Ansonsten: Größe der Bilddaten in Byte.
38	26	LONG	int32_t	4 Byte	biXPelsPerMeter	Horizontale Auflösung des Zielausgabegerätes in Pixel pro Meter; wird aber für BMP-Dateien meistens auf 0 gesetzt.
42	2A	LONG	int32_t	4 Byte	biYPelsPerMeter	Vertikale Auflösung des Zielausgabegerätes in Pixel pro Meter; wird aber für BMP-Dateien meistens auf 0 gesetzt.
46	2E	DWORD	uint32_t	4 Byte	biClrUsed	<ul style="list-style-type: none"> Wenn biBitCount=1: 0. Wenn biBitCount=4 oder 8: die Anzahl der Einträge der Farbtabelle; 0 bedeutet die maximale Anzahl (2, 16 oder 256). Ansonsten: Die Anzahl der Einträge der Farbtabelle (0=keine Farbtabelle). Auch wenn sie in diesem Fall nicht notwendig ist, kann dennoch eine für die Farbquantisierung empfohlene Farbtabelle angegeben werden.
50	32	DWORD	uint32_t	4 Byte	biClrImportant	<ul style="list-style-type: none"> Wenn biBitCount=1, 4 oder 8: Die Anzahl sämtlicher im Bild verwendeten Farben; 0 bedeutet alle Farben der Farbtabelle. Ansonsten: Wenn eine Farbtabelle vorhanden ist und diese sämtliche im Bild verwendeten Farben enthält: deren Anzahl. Ansonsten: 0.

Die geforderten Filter sollen nur auf Bitmap Dateien mit einer 16 Bit Farbkodierung angewendet werden können. Diese Info steht im Feld **biBitCount** des BITMAPINFOHEADER:

In allen anderen Fällen soll das Programm mit einer aussagekräftigen Info beendet werden.

In einem 16 Bit Bitmap ist ein Pixel mit den Farben rot, grün blau in einem 16 Bit Short gespeichert. Diese 16 Bit sind folgendermaßen strukturiert:

5 Bits für Rot (R)
6 Bits für Grün (G)
5 Bits für Blau (B)

Beispiel :

Pixelwert : 16 Bit dezimal Wert : **16063**
 16 Bit hex Wert : 0x3EBF
 16 Bit binär Wert : 0011111010111111

0 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1
 R G B

Eine einfache Bitmap mit 12 Pixel mit einer Höhe von 4 px und einer Breite von 3 px würde dann z.B. folgende Pixel Info beinhalten:

16063 23368 31979 42933 23397 27628 17571 675 1716 23669 61604 35083

Die lineare Abfolge der Short Werte wird in diesem Beispiel dann als eine Matrix mit 3 Zeilen und 4 Spalten interpretiert.

Filter Rotation :

Die Rotation um 90° bei einer 4 x 3 Bitmap hätte dann folgende Werte:

					9	5	1
1	2	3	4		10	6	2
5	6	7	8		11	7	3
9	10	11	12	wird zu	12	8	4

Filter Invert:

Der Filter wird auf jedes Pixel einzeln angewendet. Beispielhaft für ein Pixel würde ein invertieren des grün Anteil folgendermaßen aussehen:

Pixelwert: 16 Bit dezimal Wert: 16063
 16 Bit hex Wert: 0x3EBF
 16 Bit binär Wert: 0011111010111111

00111 110101 11111
 R. G B

Inverses Grün:

00111 001010 11111
 R G B

16 Bit dezimal Wert: 14687
 16 Bit hex Wert: 0x395F

Aufgabe:

1. Einlesen der gegebenen Bitmap Datei. Der Name der Bitmap Datei soll der erste Kommandozeilenparameter sein.
2. Auswerten der Kommandozeilen Information
3. In Abhängigkeit der Kommandozeilen Informationen den Filter anwenden oder die Datei Info ausgeben.
4. Wenn ein Filter gewählt wurde, das gefiltertes Bild als neuen Datei speichern.
Der Name der neuen Datei soll der zweite Parameter in der Kommandozeilen Info sein.

Hinweise:

- Versuchen Sie so weit als möglich alle Themengebiete die in der Vorlesung behandelt wurden in dem Projekt abzudecken.
Ausgenommen sind folgende Einzeltechniken:
 - Zeiger auf Funktionen
 - union
 - Fließkomma Zahlen
 - Rekursionen
- Unter Umständen ist es Hilfreich die Speicherausrichtung von Strukturen auf 1 Byte festzulegen