



Die ASCII Tabelle

Hörsalübung

ASCII - American Standard Code for Information Interchange

000	NUL	033	!	066	B	099	c	132	ä	165	Ñ	198	ã	231	þ
001	Start Of Header	034	"	067	C	100	d	133	å	166	ª	199	Ä	232	Ë
002	Start Of Text	035	#	068	D	101	e	134	ä	167	º	200	Ľ	233	Ú
003	End Of Text	036	\$	069	E	102	f	135	ç	168	¿	201	Œ	234	Û
004	End Of Transmission	037	%	070	F	103	g	136	ê	169	®	202	⋈	235	Ü
005	Enquiry	038	&	071	G	104	h	137	ë	170	¬	203	⏐	236	Ý
006	Acknowledge	039		072	H	105	i	138	è	171	½	204	⏏	237	Ÿ
007	Bell	040	(073	I	106	j	139	ï	172	¼	205	=	238	ˉ
008	Backspace	041)	074	J	107	k	140	î	173	ı	206	≠	239	˘
009	Horizontal Tab	042	*	075	K	108	l	141	ì	174	«	207	×	240	-
010	Line Feed	043	+	076	L	109	m	142	Ā	175	»	208	δ	241	±
011	Vertical Tab	044	,	077	M	110	n	143	Ā	176	∴	209	∅	242	_
012	Form Feed	045	-	078	N	111	o	144	É	177	∵	210	È	243	¼
013	Carriage Return	046	.	079	O	112	p	145	æ	178	⏏	211	É	244	⏏
014	Shift Out	047	/	080	P	113	q	146	Æ	179		212	Ê	245	§
015	Shift In	048	0	081	Q	114	r	147	ø	180	†	213	ı	246	÷
016	Delete	049	1	082	R	115	s	148	ö	181	À	214	í	247	˙
017	-- frei --	050	2	083	S	116	t	149	ò	182	Á	215	î	248	˚
018	-- frei --	051	3	084	T	117	u	150	ú	183	Ā	216	ï	249	˛
019	-- frei --	052	4	085	U	118	v	151	û	184	⊙	217	ı	250	˜
020	-- frei --	053	5	086	V	119	w	152	ÿ	185	⏏	218	ı	251	˘
021	Negative Acknowledge	054	6	087	W	120	x	153	Ö	186		219	■	252	˚
022	Synchronous Idle	055	7	088	X	121	y	154	Ü	187	⏏	220	■	253	˚
023	End Of Transmission Block	056	8	089	Y	122	z	155	ø	188	⏏	221	ı	254	■
024	Cancel	057	9	090	Z	123	{	156	£	189	¢	222	ı	255	
025	End Of Medium	058	:	091	[124		157	∅	190	¥	223	■		
026	Substitute	059	;	092	\	125	}	158	×	191	₹	224	Ó		
027	Escape	060	<	093]	126	~	159	f	192	₹	225	Ö		
028	File Separator	061	=	094	^	127	▯	160	á	193	₹	226	Ó		
029	Group Separator	062	>	095	_	128	Ç	161	í	194	₹	227	Ó		
030	Record Separator	063	?	096	`	129	ü	162	ó	195	₹	228	ö		
031	Unit Separator	064	@	097	a	130	é	163	ú	196	₹	229	Ö		
032		065	A	098	b	131	â	164	ñ	197	₹	230	μ		

Was steckt dahinter

Als Zahl dargestellt

$$1 * 2^6 + 1 * 2^0 = 65$$

1 - "byte"



als char dargestellt

'A'

```
main () {  
    char Zeichen;  
    int Zahl;  
    Zeichen = 'A'; // Zuweisung entsprechend dem Datentyp  
    Zahl = Zeichen // Implizite Umwandlung = es wird das Bitmuster des Zeichens kopiert  
                  // und damit der Zahlenwert 65  
  
    Zeichen = 66; // Es wird das Bitmuster f+r 65 kopiert und dann als Zeichen interpretiert.  
                 // Die Übersetzung "welches Zeichen" erfolgt nach ASCII "Codepage"
```

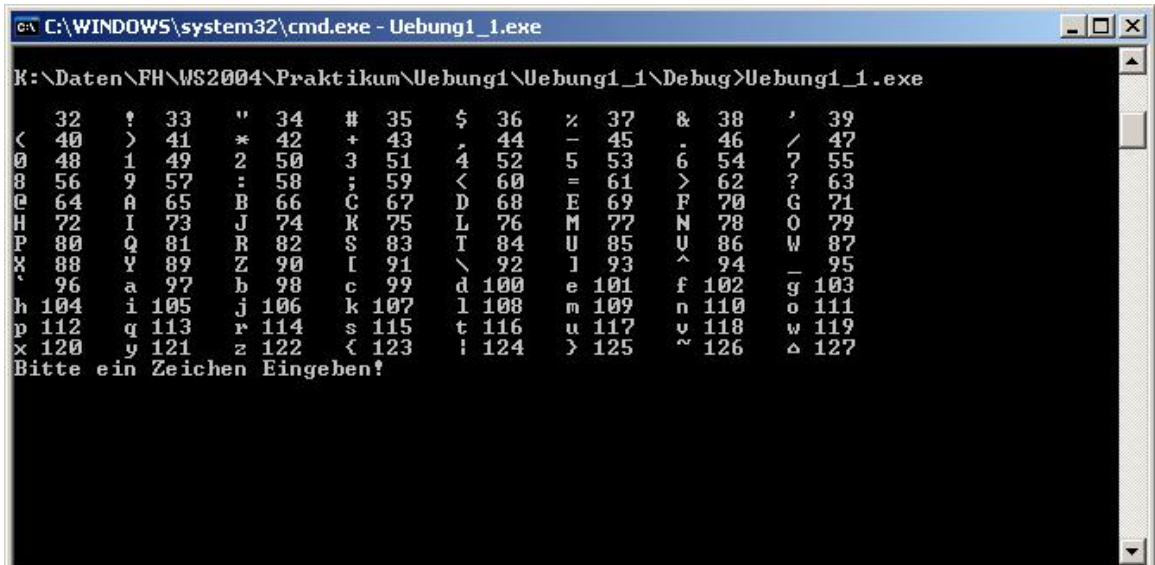
Unser Programm aus Übung 1

```
#include <iostream.h>
#include <iomanip.h>
void main ()
{
for (int i=32; i<=127; i++) {
int r = i%8;
if (r == 0)
cout << endl;
char c = i;
cout << c << " " << setw(3) << i << "  ";
}
cout << endl;
}
```

int i;
char c;
Schleife über i

c = i; // implizite Zuweisung

cout << c // als Zeichen
<< i // als Zahl



```
C:\WINDOWS\system32\cmd.exe - Uebung1_1.exe
K:\Daten\FH\WS2004\Praktikum\Uebung1\Uebung1_1\Debug>Uebung1_1.exe
 32  !  33  "  34  #  35  $  36  %  37  &  38  '  39
< 40  > 41  *  42  +  43  ,  44  -  45  .  46  /  47
@ 48  1  49  2  50  3  51  4  52  5  53  6  54  7  55
8 56  9  57  :  58  ;  59  < 60  =  61  > 62  ?  63
@ 64  A  65  B  66  C  67  D  68  E  69  F  70  G  71
H 72  I  73  J  74  K  75  L  76  M  77  N  78  O  79
P 80  Q  81  R  82  S  83  T  84  U  85  V  86  W  87
X 88  Y  89  Z  90  [  91  \  92  ]  93  ^  94  _  95
` 96  a  97  b  98  c  99  d 100  e 101  f 102  g 103
h 104  i 105  j 106  k 107  l 108  m 109  n 110  o 111
p 112  q 113  r 114  s 115  t 116  u 117  v 118  w 119
x 120  y 121  z 122  < 123  ! 124  > 125  ~ 126  Δ 127
Bitte ein Zeichen Eingeben!
```

Mit Zeichen kann man rechnen!

- Ob Zahl oder Zeichen hängt nur vom Kontext ab.
 - In der Regel macht der Compiler alles richtig.
- `char C = 'A' + 1;`
 - Der int-Wert von A (65) wird um
 - 1 erhöht (d.h. 66)
 - und dies in der Variablen C gespeichert
 - Die Interpretation als Zeichen erfolgt nach der ASCII Tabelle als:
66 == 'B'
 - Zum Glück sind die Buchstaben "alphabetisch" aufsteigend sortiert.
- Tipp: `unsigned char` erlaubt auch Werte > 128 (d.h. Sonderzeichen)

```
int main () {  
  
    char C = 'C';  
    int   I = C - 'A';  
  
    cout << C << " C - A = " << I;  
  
    C = C++ ;  
    cout << "C++ = " << C;  
  
    char Text[10] = "ABCDEFGHGI";  
    for (int j = 0; j < 10; j++) {  
        I = Text[j];  
        cout << " Text["<<j<<"] = " <<  
            Text[j] << " : " << I;  
    }
```

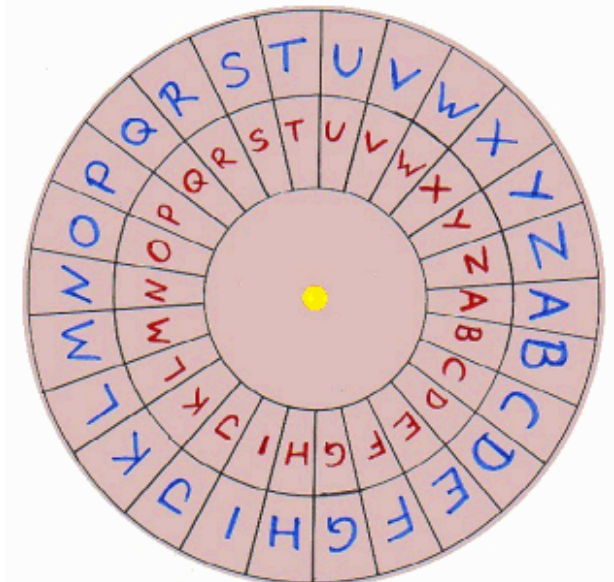
Die Caesar - Verschlüsselung

- Caesar verwendet eine Verschlüsselung, bei der jeder Buchstabe um 3 Stellen im Alphabet verschoben wird.

Klar	A	B	C	D	E	F	G	H	I	J	K	L	M
Geheim	D	E	F	G	H	I	J	K	L	M	N	O	P
Klar	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Geheim	Q	R	S	T	U	V	W	X	Y	Z	A	B	C



- D.h. Geheimbuchstabe = Klarbuchstabe + 3
- Man kann sich auch eine Verschlüsselungs-Maschine bauen bei der der Verschiebe-Wert (d.h. der Schlüssel/Key) beliebig gewählt werden kann.



Ihre Aufgabe

- Entwerfen Sie ein Programm zum Verschlüsseln eines Wortes nach dem Caesar Verfahren.
 - Der Key soll vom Programm erfragt werden.
 - Das Programm soll nach dem Klartext-Wort fragen und das verschlüsselte Wort ausgeben.

 - Einschränkungen für Ihr Design
 - Der Key solle zwischen 0 und 26 liegen (automatisch???)
 - Nur Großbuchstaben sollen konvertiert werden, alle anderen Zeichen bleiben unverändert
 - Tipp
 - Arbeiten Sie mit einem Zeichenketten Array `char Text[256];`
-