

1. Zeigermanipulationen (1 + 2 + 2 Punkte)

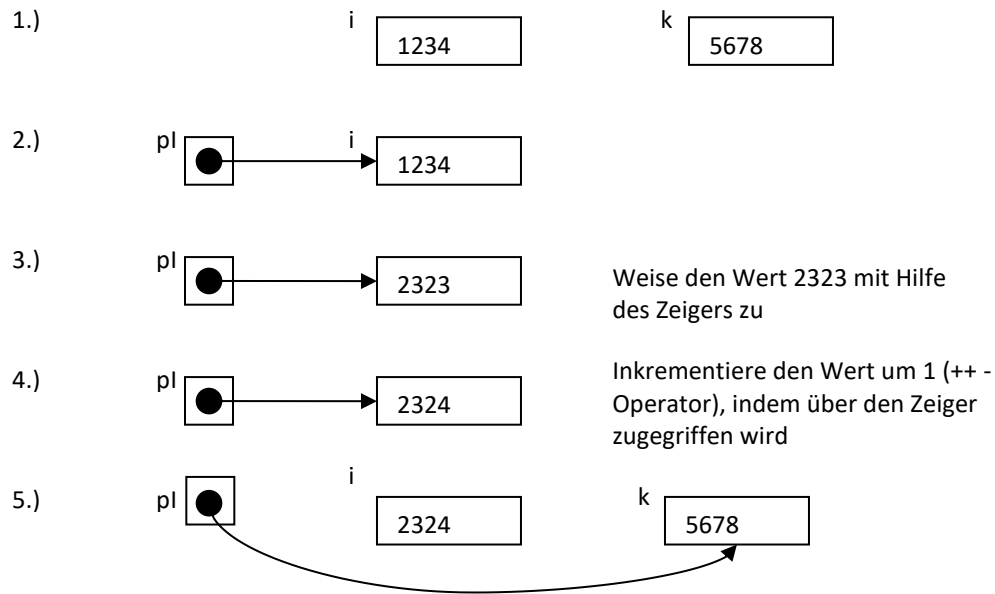
a) Was ist ein Zeiger? Wie unterscheidet sich eine Zeigervariable von einer „herkömmlichen“ Variable?

b) Beschreiben Sie für den folgenden Code Zeile für Zeile, was gemacht wird,

z.B.: `printf("%d", *pA)` „Der Pointer pA wird mittels *-Operator dereferenziert und ausgegeben, erwartetes Ergebnis ist: ...“

```
/* 01 */      int i = 5;
/* 02 */      int j = 6;
/* 03 */      int t = 0;
/* 04 */      int* pI = NULL;
/* 05 */      int* pJ = &j;
/* 06 */      float* pF = (float*) &i;
/* 07 */      pI = &i;
/* 08 */      printf("%i, %i", *pI, *pJ);
/* 09 */      t = *pI;
/* 10 */      *pI = *pJ;
/* 11 */      *pJ = t;
/* 12 */      printf("%i, %i", *pI, *pJ);
/* 13 */      pJ = pI;
/* 14 */      printf("%i, %i", *pI, *pJ);
/* 15 */      *pJ = 10;
/* 16 */      printf("%i, %i", *pI, *pJ);
```

c) Schreiben Sie ein Programm, das die folgende Grafik Schritt für Schritt implementiert mit einer Ausgabe nach jedem Schritt:



2. Tokenizer (5 + 5 Punkte)

Entwickeln Sie eine Funktion `tokenize`, die innerhalb eines Satzes die einzelnen Worte identifiziert. Ein Satz ist eine Zeichenkette, die aus Groß- und Kleinbuchstaben und Leerzeichen zur Worttrennung besteht.

Entwickeln Sie zwei Versionen der Funktion. Die erste Version liefert die Indices (Feld von Integerwerten), die zweite die Adressen (Feld von `char`-Zeigern), an denen die vorkommenden Worte anfangen, zurück.

Beispiel:

Text	D	i	e	s		i	s	t		e	i	n		P	r	o	b	e	t	e	x	t	\0
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
Adr	0x12			0x17																	

Resultat der 1. Version: 0, 5,...

Resultat der 2. Version: 0x12, 0x17, ...

Geben Sie die Resultate im Testtreiber für verschiedene Sätze aus!

3. Zeichenkette kopieren mit dynamischem Speicher (5 Punkte)

Implementieren Sie eine C-Funktion, die als Parameter eine Zeichenkette erhält, diese in einen neuen, dynamisch angelegten Speicherbereich kopiert und die duplizierte Zeichenkette als Rückgabewert zurückliefert.

```
char* strDuplicate(const char * pSrc);
```

Achten Sie auf ordnungsgemäße Allokation und Deallokation des dynamisch angelegten Speichers!