Lösungshinweise zur
Teilprüfung
Software- und Internettechnologie
Programmierkurs 2
Herbstsemester 2007/2008

Aufgabe 1:

a) Ausgabe der Funktion:

```
0
1
2
3
```

b) `a=7` und `b=1`

c)
```
void min_max(int*a, int n, int* min, int* max){
  int i;
  if (n<=0 || a==NULL || min==NULL || max==NULL) return;
  *min = *max = a[0];
  for (i=1 ; i < n ; i++){
    if (a[i] < *min) *min = a[i];
    if (a[i] > *max) *max = a[i];
  }
}
```

d)
```
  int a[] = {5,4,2,3,1};
  int min, max;
  min_max(a,5,&min,&max);
  printf("min=%i max=%i\n",min,max);
```

e)
```
int strlen(char* s){
  int i=0;
  if (s==NULL) return 0;
  while (*s!='\0'){
    i++;
    s++;
  }
  return i;
}
```

Aufgabe 2:

a)
```
void draw(int** a){
  int i,j;
  for (i=0 ; i < 9 ; i++){
    for (j=0 ; j < 9 ; j++){
      if (a[i][j]!=0) printf("%d ",a[i][j]);
      else printf("  ");
    }
    printf("\n");
  }
}
```

b)

```
int error(int** a){
  int i,j;
  int tmp[9];
  for (i=0 ; i < 9 ; i++){
    for (j=0 ; j < 9 ; j++) tmp[j]=0;
    for (j=0 ; j < 9 ; j++){
      if (a[i][j]!=0){
        if (tmp[a[i][j]-1]!=0) return i;
        tmp[a[i][j]-1]=1;
      }
    }
  }
  return -1;
}
```

Aufgabe 3:

a)

```
list_t* insert(list_t* list, int wert){
  list_t* head = list;
  list_t* new;
  if (list==NULL){
    list = (list_t*) malloc(sizeof(list_t));
    if (list==NULL) return NULL;
    head=list;
  }
  while (list->next!=NULL) list = list->next;
  new = (list_t*) malloc(sizeof(list_t));
  if (new==NULL) return NULL;
  new->data = wert;
  new->next = NULL;
  list->next = new;
  return head;
}
```

b)

```
list_t* delete_even(list_t* list){
  list_t* head = list;
  list_t* tmp;
  if (list==NULL) return NULL;
  while(list->next != NULL){
    tmp = list->next;
    if (tmp->data %2 == 0){
      list->next = tmp->next;
      free(tmp);
    }
    else list = tmp;
  }
  return head;
}
```

c)

```
list_t* reverse(list_t* list){
  list_t* head = list;
  list_t *tmp, *p=NULL;
  if (list==NULL) return NULL;
  list = list->next;
  while (list !=NULL){
    tmp = list->next;
    list->next = p;
    p=list;
    list=tmp;
  }
  head->next = p;
  return head;
}
```

Aufgabe 4:

a) `mov r11,r10`

b) `R10`=7, `R11`=17

c) `R11`=0x0101=257

d) Der Program Counter enthält die Speicheradresse, an der der nächste auszuführende Befehl steht.

Aufgabe 5:

```
        mov     #1,r11          ; z=1
loop:   tst     r9
        jge     end             ; if (0>=b) goto end
        mov     r9,r6
        and     #1,r6           ; b%2
        jz      cont            ; if (b%2==0) goto cont
        mov     r11,r12         ; berechne z*a in r14
        mov     r8,r13
        call    #mul
        mov     r14,r12         ; kopiere Ergebnis nach r12
        mov     r10,r13
        call    #mod
        mov     r14,r11         ; kopiere z*a mod n nach r11
cont:   mov     r8,r12          ; berechne a*a in r14
        mov     r8,r13
        call    #mul
        mov     r14,r12         ; kopiere Ergebnis nach r12
        mov     r10,r13
        call    #mod
        mov     r14,r8          ; kopiere a * a mod n nach r8
        rra     r9              ; b = b / 2
        jmp     loop
end:    ret
```