Aufgabe 33)

1)

```
#include <stdio.h>

int main(){
   int w[3];
   int *r = w;
   w[0] = 1;
   w[1] = 2;
   w[2] = 3;

   r = r + 2;
   printf("Start:");
   printf("%i\n", *r);
   printf("%i\n", w[2]);
   return 0;
}
```

2)

```
#include <stdio.h>

int *p;
int main(){
   if(p == NULL)
      printf("Zeiger zeigt nirgendwohin\n");
   return 0;
}
```

3)

```
#include <stdio.h>

int main(){
   int n;
   int *p= &n;
   n = 5;
   printf("     %d\n---working---", *p);
   return 0;
}
```

4)

```
#include <stdio.h>

int main(){
   int v[] = {5, 7, 9};
   printf("%d", *(v + 2));
   return 0;
}
```

5)

```
#include <stdio.h>

int main(){
   int v[] = {5, 7, 9};
   int *p = v;
   printf("Erster Durchlauf:  %d\n", *(p++));
   printf("Zweiter Durchlauf: %d\n", *(p++));
   printf("Dritter Durchlauf: %d\n", *(p++));
   return 0;
}
```

6)

```
#include <stdio.h>

int main(){
   int v[] = {5, 7, 9};
   int *p = v;
   printf("Erster Durchlauf:  %d\n", ++(*p));
   printf("Zweiter Durchlauf: %d\n", ++(*p));
   printf("Dritter Durchlauf: %d\n", ++(*p));
   return 0;
}
```

7)

```
#include <stdio.h>

int main(){
   int v[] = {5, 7, 9};
   int *p = v;
   printf("Erster Durchlauf:  %d\n", *(++p));
   printf("Zweiter Durchlauf: %d\n", *(++p));
   printf("Dritter Durchlauf: %d\n", *(++p));
   return 0;
}
```

8)

```
#include <stdio.h>

int main(){
   char v[] = "Hallo";
   v[2] = '\0';
   printf("%s", v);
   return 0;
}
```

9)

```
#include <stdio.h>

int main(){
   char v[] = "Hallo";
   printf("%s", v + 2);
   return 0;
}
```

10)

```
#include <stdio.h>

int main(){
   char v[] = "Informatik";
   int p = *v;
   return 0;
}
```

11)

```
#include <stdio.h>

int main(){
   int v[20];
   int *p = v;
   ++(*p);
   return 0;
}
/*  kein Fehler? */
```

12)

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

int main(){
   char v[] = "Informatik";
   char w[sizeof v];
   strcpy(w,v);
   printf("%s", w);
   return 0;
}
```

13)

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

int main(){
   char *p = malloc (sizeof (char));
   *p = '5';
   return 0;
}
```

14)

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

int main(){
   char *p = malloc( sizeof ( char));        /* wert, auf den pointer ohne adresse zeigt, kann nicht
                                                verändert werden*/
   *p = '5';
   return 0;
}
```

15)

```
#include <stdio.h>

int main(){
   char v;
   char *p = &v;   /* kein Array sondern Variable, also '&' */
   *p = '5';       /* kein Array, also kein [] */
   return 0;
}
```