

```
int * ip;
int i;
```

Uttryck	Typ	Värde
ip	int *	Pekare till ett heltal
*ip	int	Det heltal pekaren pekar på
&ip	int **	Pekare till pekare till heltal
&i	int *	Pekare till heltal
i	int	Heltal
*&i	int	Heltal (avrefererar adressen till i = kommer tillbaka till i)

```
ip = &i                /* ip pekar på i */

i = *ip;              /* läs av värdet på det ip pekar på */
*ip = i;              /* sätt värdet på det ip pekar på */

ip = NULL;           /* ip pekar inte på något */
```

```
char * cp;           /* Pekare till teckensträng, inget minne allokerat */
int n;
```

Uttryck	Typ	Värde
cp	char *	Pekare till sträng (eller bara ett tecken, analogt med int * ovan)
*cp	char	Första tecknet i strängen.
&cp	char **	Pekare till pekare till sträng.
cp+n	char *	Pekare till n:te tecknet i strängen
&cp[n]	char *	Pekare till n:te tecknet i strängen
cp[n]	char	n:te tecket i strängen
*(cp+n)	char	n:te tecknet i strängen
cp++	char *	stega fram pekaren till nästa tecken

```
cp = malloc( storlek * sizeof(char)); /* allokerar storlek tecken */

while( *cp != '\0' ) /* hitta slutet på strängen */
    cp++;

for(n=0; n<storlek-1; n++) /* stoppa in 'a' i hela strängen */
    cp[n] = 'a';
cp[n] = '\0';           /* glöm inte avslutningsmarkering */

scanf("%s",cp);        /* läs in till sträng från tangentbord */
/* OBS! Inget "&" innan cp: cp är en pekare */
printf("%s",cp);      /* skriv ut sträng */
```

```
char cv[20];          /* Vektor av tecken: sträng av fix längd. */
int n;               /* Strängen allokerad, ingen malloc krävs */
```

Uttryck	Typ	Värde
cv	char *	Pekare till strängen (adress till första tecknet)
*cv	char	Första tecknet i strängen.
&cv	char *	Pekare till sträng: special fall!
cv+n	char *	Pekare till n:te tecknet i strängen
&cv[n]	char *	Pekare till n:te tecknet i strängen
cv[n]	char	n:te tecket i strängen
*(cv+n)	char	n:te tecknet i strängen
cv++		Får man inte göra!

```
char * cp;

cp = cv;          /* OK */
cp = &cv;        /* samma sak */

*cp = cv[0];     /* kopiera från vektor till utpekad sträng */

scanf("%s",cv);  /* läs in från tangentbord till sträng          */
                  /* cv är inte pekare, men konverteras automatiskt */
printf("%s",cv); /* skriv ut sträng */
```

```
void foo( char s[] ); /* teckenvektor som parameter till funktion */
```

Uttryck	Typ	Värde
s	char *	Pekare till strängen (adress till första tecknet)
*s	char	Första tecknet i strängen.
&s	char **	Pekare till pekare till sträng. Inte samma som för char s[20]!!!
s+n	char *	Pekare till n:te tecknet i strängen
&s[n]	char *	Pekare till n:te tecknet i strängen
s[n]	char	n:te tecket i strängen
*(s+n)	char	n:te tecknet i strängen
s++		Får man inte göra!

```
void print_string(char s[])
{
    int i = 0;
    while( s[i] != '\0' )
    {
        putchar(s[i]);    /* skriv ut n:te tecket i strängen */
        i++;
    }
}

return s;    /* skickar tillbaka pekare till strängen */
return &s;   /* skickar tillbaka adressen till lokala variabeln s */
```

```

/* länkad lista av structar */

struct book {
    char        title[40];
    long int    isbn;
    struct book * next; /* pekare till nästa bok i listan */
};

struct book    b; /* bokvariabel */
struct book    *bp; /* pekare till bok */

```

Uttryck	Typ	Värde
b	struct book	En bok
&b	struct book *	Pekare till boken b
bp	struct book *	Pekare till en bok
*bp	struct book	En bok
b.isbn	long int	Ett fält i boken b
bp->isbn	long int	Ett fält i den bok som bp pekar på
(*bp).isbn	long int	Ett fält i den bok som bp pekar på. Samma som bp->isbn
bp->next	struct book *	Pekare till nästa bok i listan
&(bp->isbn)	long int *	Pekare till fältet i isbn i den bok bp pekar på
bp->next->next	struct book *	Nästa boks pekare till boken efter den

```

struct book b2;
struct book *bp2; /* pekare till bok */

bp = NULL; /* bp pekar inte på något */

bp = malloc(sizeof(struct book)); /* skapa ny bok, bp pekar */

b2 = b; /* kopiera hela innehållet i b till b1 */
*bp = b; /* kopiera hela innehållet i b dit bp pekar */

b = b->next; /* gå till nästa bok */

b->isbn = 15; /* sätt värde i utpekad bok */

printf("%s",bp->title); /* skriv ut titeln */
scanf("%d",&(bp->isbn)); /* läs in till isbn i den bok bp pekar på */
scanf("%s",bp->title); /* jämför fallet för cp och cv ovan */

bp2 = bp; /* bp och bp2 pekar på samma bok */
bp2->isbn = 20; /* ändrar i boken: även bp->isbn ändras */
bp->next = NULL; /* ändrar också bp2->next */

```