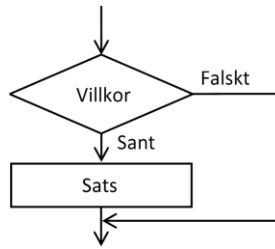


Styrstrukturer i C

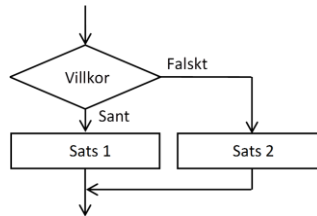
**Globala och lokala variabler,
inparametrar vid funktionsanrop
och
returvärden från funktioner i XCC.**

Styrstrukturer i C

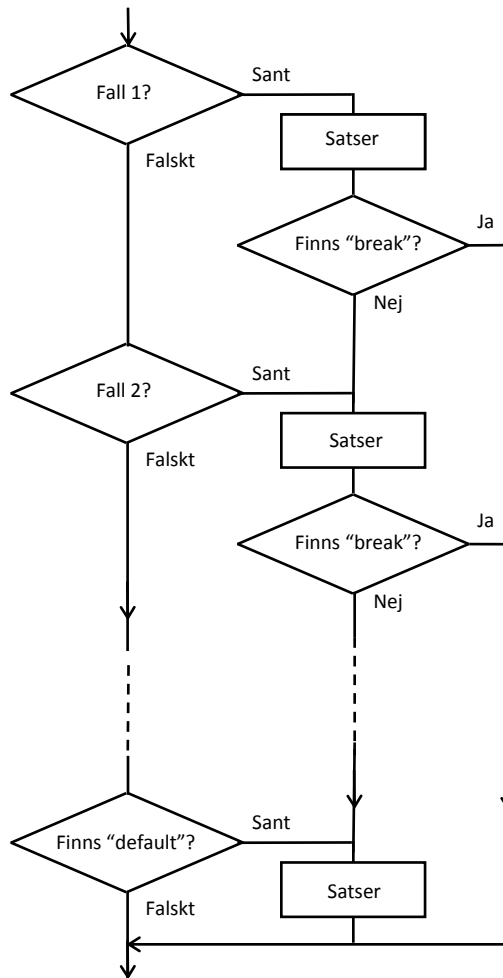
Ex.
if (i > 0)
 sats



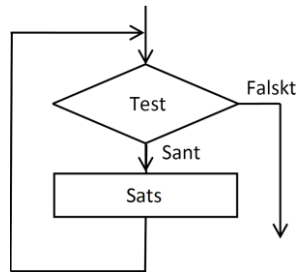
Ex.
if (i > 0)
 sats1
else
 sats2



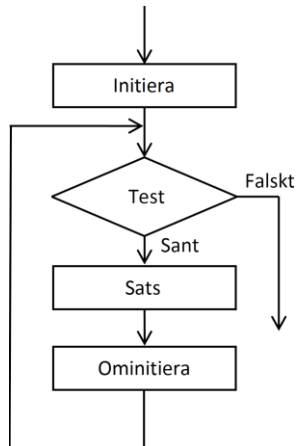
Ex.
switch (i) {
 case Fall1:
 satser
 break;
 case Fall2:
 satser
 break;
 .
 .
 .
 default:
 satser
 break;
}



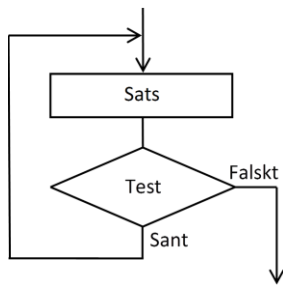
Ex.
 while ($i > 0$)
 sats



Ex.
 for ($i = 0; i < n; i++$)
 sats



Ex.
 do
 sats
 while ($i > 0$);



Globala och lokala variabler i XCC.

// Deklaration av globala variabler

short	shortint;	_shortint:	RMB	\$2
long	longint;	_longint:	RMB	\$4
int	justint;	_justint:	RMB	\$2
int	intvec[3];	_intvec:	RMB	\$6

struct {				
int	s1;			
char	s2;			
char*	s3;			
} komplex;		_komplex:	RMB	\$5

// Deklaration och initiering av globala variabler

short	shortint = 0x1122;	_shortint:	FDB	\$1122
long	longint = 0x33445566;	_longint:	FQB	\$33445566
int	justint = 0x7788;	_justint:	FDB	\$7788
int	intvec[3] = {0x1234,0x5678,0x9ABC};	_intvec:	FDB	\$1234
			FDB	\$5678
			FDB	\$9ABC

struct {				
int	s1;			
char	s2;			
char*	s3;			
} komplex = {0x3421,0xFF,0x3333};		_komplex:	FDB	\$3421
			FCB	\$FF
			FDB	\$3333

```
// Deklaration av lokala variabler.
// Alla lokala variabler läggs på stacken.
// Alla funktioner med lokala variabler inleds med Prolog och avslutas med Epilog.
// Prolog skapar utrymme för de lokala variablerna på stacken genom att minska SP.
// Epilog återställer stacken så att SP pekar på återhoppadressen.
```

```
void main(void) {      _main:      LEAS      -19,SP      Prolog
    short shortint;
    long  longint;
    int   justint;
    int   intvec[3];

    struct {
        int s1;
        char s2;
        char* s3;
    } komplex;

    justint = 0;

                                CLRA
                                CLRB
                                STD      11,SP      justint = 0 till stack

    }                        LEAS      19,SP      Epilog
                                RTS
```

//Hur lokala variabler tilldelas värden på stacken

void main(void) {	_main:	LEAS	-19,SP	Prolog
short shortint;				
long longint;				
int justint;				
int intvec[3];				
struct {				
int s1;				
char s2;				
char *s3;				
} komplex;				
shortint = 1;		LDD	#1	
		STD	17,SP	
longint = 2;		LDY	#0x0000	
		LDD	#0x0002	
		STY	13,SP	
		STD	15,SP	
justint = 3;		LDD	#3	
		STD	11,SP	
intvec[0] = 4;		LDD	#4	
		STD	5,SP	
intvec[1] = 5;		LDD	#5	
		STD	7,SP	
intvec[2] = 6;		LDD	#6	
		STD	9,SP	
komplex.s1 = 7;		LDD	#7	
		STD	0,SP	
komplex.s2 = 8;		LDAB	#8	
		STAB	2,SP	
komplex.s3 = (char *) 9;		LDD	#9	
		STD	3,SP	
}		LEAS	19,SP	Epilog
		RTS		

Inparametrar vid funktionsanrop i XCC.

// Funktionsanrop med två globala inparametrar
 // Inga lokala parametrar i den anropade funktionen

void callfunc(int aa, int bb);

int alfa = 0x1234;	_alfa:	FDB	\$1234	
int beta = 0x5678;	_beta:	FDB	\$5678	
void main(){	_main:			
callfunc(alfa,beta);		LDD	_beta	
		PSHD		beta till stack
		LDD	_alfa	
		PSHD		alfa till stack
		JSR	_callfunc	
}		LEAS	4,SP	Justera stack
		RTS		

void callfunc(aa,bb){	_callfunc:	RTS	
}			

// Funktionsanrop med två globala inparametrar
 // Två lokala parametrar i den anropade funktionen

void callfunc(int aa, int bb);

int alfa = 0x1111;	_alfa:	FDB	\$1111	Tilldelningar
int beta = 0x5678;	_beta:	FDB	\$5678	
void main(void){	_main:			
callfunc(alfa,beta);		LDD	_beta	
		PSHD		beta till stack
		LDD	_alfa	
		PSHD		alfa till stack
		JSR	_callfunc	
}		LEAS	4,SP	Justera stack
		RTS		
void callfunc(aa,bb){	_callfunc:	LEAS	-3,SP	Prolog
char gamma = 0x55;		LDAB	#\$55	Tilldelningar
		STAB	2,SP	gamma till stack
int delta = 0x9999;		LDD	#\$9999	
		STD	0,SP	delta till stack
}		LEAS	3,SP	Epilog
		RTS		

```
// Funktionsanrop med två lokala inparametrar från main
// Två lokala parametrar i den anropade funktionen
```

```
void callfunc(int aa, int bb);
```

void main(void){	_main:	LEAS	-4,SP	Prolog
int alfa = 0x1111;		LDD	#\$1111	Tilldelningar
		STD	2,SP	alfa till stack
int beta = 0x5678;		LDD	#\$5678	
		STD	0,SP	beta till stack
callfunc(alfa,beta);		PSHD		beta finns i D-reg
		LDD	4,SP	alfa
		PSHD		
		JSR	_callfunc	
		LEAS	4,SP	Justera stack
}		LEAS	4,SP	Epilog
		RTS		
void callfunc(aa,bb){	_callfunc:	LEAS	-3,SP	Prolog
char gamma = 0x55;		LDAB	#\$55	Tilldelningar
		STAB	2,SP	gamma till stack
int delta = 0x9999;		LDD	#\$9999	
		STD	0,SP	delta till stack
}		LEAS	3,SP	Epilog
		RTS		

Returvärden från funktioner i XCC.

char, int, short, long och float returneras i register.

char i B-registret.

int och short i D-registret.

long och float i Y- och D-registret, med mest signifikant del i Y-registret.