```c
/* ************************************************************************** */
/*                                                                            */
/*                                                        :::::::::           */
/*   helperfunctions.c                                   :+:      :+:         */
/*                                                      +:+                    */
/*   By: dvan-kri <dvan-kri@student.codam.nl>          +#+                     */
/*                                                    +#+                      */
/*   Created: 2021/03/14 19:08:34 by dvan-kri        #+#    #+#                */
/*   Updated: 2021/03/20 18:44:41 by dvan-kri       ########   odam.nl         */
/*                                                                            */
/* ************************************************************************** */

#include <stdarg.h>
#include <stdio.h> /* dit is alleen voor de originele printf verwijder later */
#include "../includes/ft_printf.h"
#include "../libft/libft.h"

/* deze functie wordt aangeroepen als er een procent teken gevonden is in de format string.
De functie gaat de conversie specificatie analyzeren, de gegevens in de t_convert struct zetten en ten slo
De functie returnt het aantal karakters van de format string dat verwerkt is, zodat de parse functie verde


int ft_checkflag(char *format, t_convert *conv_specs)
{
	int i;

	i = 0;
	if (format[i] == '0')
	{
		conv_specs->zero = 1;
		i++;
	}
	else if (format[i] == '-')
	{
		conv_specs->minus = 1;
		i++;
	}
	return (i);
}

int ft_checkprecision(char *format, t_convert *conv_specs, va_list ap)
{
	int i;
	int j;

	i = 0;
	j = 0;
	if (format[i] == '.')
	{
		i++;
		if (format[i] == '*')
		{
			conv_specs->precision = va_arg(ap, int);
			conv_specs->precision_asterisk = 1;
			return (1);
		}
		else
		{
			while (ft_isdigit(format[i + j]))
			{
				j++;
			}
			if (j)
				conv_specs->precision = ft_atoi(ft_substr(format, i, j));
			return (1);
		}
	}
	return (0);
}
```

```c
int ft_checkasterisk(char *format, t_convert *conv_specs, va_list ap)
{
 if (*format == '*')
 {
  conv_specs->width_asterisk = va_arg(ap, int);
  return (1);
 }
 return (0);
}

int ft_checkwidthdigit(char *format, t_convert *conv_specs)
{
 int i;

 i = 0;
 while (ft_isdigit(format[i]))
  i++;
 if (i)
  conv_specs->width_digit = ft_atoi(ft_substr(format, 0, i));
 return (i);
}

int ft_checktype(char *format, t_convert *conv_specs)
{
 if (*format == 'd' || *format == 'i' || *format == 'u' || *format == 'x' || *format == 'p' || *format ==
 {
  conv_specs->type = *format;
  return (1);
 }
 return (0);
}
```