```c
/* ************************************************************************** */
/*                                                                            */
/*                                                        ::::::::            */
/*   ft_printf.c                                        :+:    :+:            */
/*                                                     +:+                    */
/*   By: dvan-kri <dvan-kri@student.codam.nl>         +#+                     */
/*                                                   +#+                      */
/*   Created: 2021/03/14 21:08:31 by dvan-kri      #+#    #+#                 */
/*   Updated: 2021/03/27 14:36:59 by dvan-kri      ########   odam.nl         */
/*                                                                            */
/* ************************************************************************** */

#include "../includes/ft_printf.h"
#include "../libft/libft.h"

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

/* zet de conv_specs naar initiele waardes want dat is handig als ze geen waarde toegewezen krijgen */
int init_convspecs(t_convert *conv_specs)
{
 conv_specs->width = 0;
 conv_specs->precision = -1; /* precision 0 is anders dan geen precision voor strings bijvoorbeeld */
 conv_specs->minus = 0;
 conv_specs->zero = 0;
 conv_specs->type = 'z';
 return (0);
}


void ft_argtostruct(t_convert *conv_specs)
{
 if (conv_specs->type == 'c')
  conv_specs->c = va_arg(conv_specs->ap, int);
 if (conv_specs->type == 's')
  conv_specs->s = va_arg(conv_specs->ap, char *);
 if (conv_specs->type == 'p')
  conv_specs->p = va_arg(conv_specs->ap, char *);
 if (conv_specs->type == 'd')
  conv_specs->d = va_arg(conv_specs->ap, int);
 if (conv_specs->type == 'i')
  conv_specs->i = va_arg(conv_specs->ap, int);
 if (conv_specs->type == 'u')
  conv_specs->u = va_arg(conv_specs->ap, unsigned int);
 if (conv_specs->type == 'x')
  conv_specs->u = va_arg(conv_specs->ap, unsigned int);
 if (conv_specs->type == 'X')
  conv_specs->u = va_arg(conv_specs->ap, unsigned int);
}

/* deze functie wordt aangeroepen als er een procent teken gevonden is in de format string.
De functie gaat de conversie specificatie analyzeren en de gegevens in de t_convert struct zetten.
De functie returnt het aantal karakters van de format string dat verwerkt is, zodat de parse functie verde
int check_conversion(char *format, t_convert *conv_specs)
{
 int i;

 i = 0;
 while (ft_checkflag(&format[i], conv_specs))
  i++;
 if (ft_checkasterisk(&format[i], conv_specs))
  i++;
 /* check for digits for the width */
 if (ft_checkwidthdigit(&format[i], conv_specs))
  i += ft_strlen(ft_itoa(conv_specs->width));
 if (ft_checkprecision(&format[i], conv_specs))
 {
  if (conv_specs->precision)
   i += ft_strlen(ft_itoa(conv_specs->precision));
  else
   i++;
 }
 while (!ft_checktype(&format[i], conv_specs))
  i++;
 if (ft_checktype(&format[i], conv_specs))
  i++;
 ft_argtostruct(conv_specs);
 return (i);
}


int ft_parse(char *format, t_convert conv_specs)
{

 char  *converted_argument;
 int  i;
 int  ret;

 init_convspecs(&conv_specs);
 i = 0;
 ret = 0;
 while (i < ft_strlen(format))
 {
  if (format[i] == '%')
  {
   i++;
   i += check_conversion(&format[i], &conv_specs);
   ft_putconversion(&conv_specs);
  }
  else
  {
   ft_putchar_fd(format[i], 1);
   i++;
   ret++;
  }
 }
 printspecs(conv_specs);
 return (ret);
}

int ft_printf(const char *format, ...)
 {
 t_convert conv_specs;
 int  written_bytes;

 va_start(conv_specs.ap, format);
 written_bytes = ft_parse((char *)format, conv_specs);
 va_end(conv_specs.ap);
 return (written_bytes);
}
```