```
/* ************************************************************************** */
/*                                                                            */
/*                                                        ::::::::            */
/*   ft_printf.c                                        :+:    :+:            */
/*                                                     +:+                    */
/*   By: dvan-kri <dvan-kri@student.codam.nl>         +#+                     */
/*                                                   +#+                      */
/*   Created: 2021/03/14 21:08:31 by dvan-kri      #+#    #+#                 */
/*   Updated: 2021/03/21 09:48:09 by dvan-kri      ########   odam.nl         */
/*                                                                            */
/* ************************************************************************** */

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

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

int printspecs(t_convert conv_specs)
{
 printf("width-digit: _%d_\n", conv_specs.width_digit);
 printf("width asterisk (*): _%d_\n", conv_specs.width_asterisk);
 printf("precision (.): _%d_\n", conv_specs.precision);
 printf("left-justified (-): _%d_\n", conv_specs.minus);
 printf("zero-padding (0): _%d_\n", conv_specs.zero);
 printf("conversion-type: _%c_\n", conv_specs.type);
 return (0);
}

int init_convspecs(t_convert *conv_specs)
{
 conv_specs->width_asterisk = 0;
 conv_specs->width_digit = 0;
 conv_specs->precision = 0;
 conv_specs->precision_asterisk = 0;
 conv_specs->minus = 0;
 conv_specs->zero = 0;
 conv_specs->type = 'z';
 return (0);
}

int ft_convtostring(t_convert conv_specs, va_list ap)
{
 char *string;

 string = va_arg(ap, char*);
 ft_putstr_fd(string, 1);
 return (0);
}

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

 i = 0;
 while (ft_checkflag(&format[i], conv_specs) && i < 2)
  i++;
 if (ft_checkasterisk(&format[i], conv_specs, ap))
  i++;
 /* check for digits for the width */
 if (ft_checkwidthdigit(&format[i], conv_specs))
  i += ft_strlen(ft_itoa(conv_specs->width_digit));
 if (ft_checkprecision(&format[i], conv_specs, ap))
 {
  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++;
 return (i);
}


int ft_parse(char *format, va_list ap)
{
 t_convert conv_specs;
 int i;
 int ret;

 init_convspecs(&conv_specs);
// printspecs(conv_specs);
 i = 0;
 ret = 0;
 while (i < ft_strlen(format))
 {
  if (format[i] == '%')
  {
   i++;
   i += check_conversion(&format[i], ap, &conv_specs);
   ret += ft_convtostring(conv_specs, ap);
  }
  else
  {
   ft_putchar_fd(format[i], 1);
   i++;
   ret++;
  }
 }
  /* printf("de conversion type: |%c| en i is nu: |%d|\n", conv_specs.type, i); */
// printspecs(conv_specs);
 return (ret);
}

int ft_printf(const char *format, ...)
 {
 va_list ap;
 int  written_bytes;

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