/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*/

/\* ::: :::::::: \*/

/\* ft.h :+: :+: :+: \*/

/\* +:+ +:+ +:+ \*/

/\* By: jongpark <jongpark@student.42seoul.kr> +#+ +:+ +#+ \*/

/\* +#+#+#+#+#+ +#+ \*/

/\* Created: 2020/12/06 11:24:27 by jongpark #+# #+# \*/

/\* Updated: 2020/12/06 13:28:45 by jongpark ### ########.fr \*/

/\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

#ifndef FT\_H

# define FT\_H

# include <stdlib.h>

# include <fcntl.h>

# include <unistd.h>

int ft\_is\_valid(char \*str);

void print\_dict\_err(void);

void ft\_puttc(char tc, int ws);

int ft\_clear\_buf(void);

void ft\_make\_zero(int \*i1, int \*i2);

int dict\_trim(int fd);

void ft\_nextline(int fd);

int ft\_strlen(char \*str, int \*total);

int dict\_printval(char c);

int dict\_printval2(char c1, char c2);

int dict\_printval3(char c1, char c2, char c3);

int dict\_printpos(int len);

void dict\_printvaln(int total, int len, char \*key);

void dict\_print(char \*key);

void ft\_putchar(char tc);

#endif

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*/

/\* ::: :::::::: \*/

/\* main.c :+: :+: :+: \*/

/\* +:+ +:+ +:+ \*/

/\* By: sehyan <sehyan@student.42seoul.kr> +#+ +:+ +#+ \*/

/\* +#+#+#+#+#+ +#+ \*/

/\* Created: 2020/12/05 10:21:29 by sehyan #+# #+# \*/

/\* Updated: 2020/12/06 13:29:26 by jongpark ### ########.fr \*/

/\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

#include "ft.h"

char \*g\_dictname = "numbers.dict";

char g\_buf[1024];

int g\_wasspace = 0;

int g\_isfirst = 1;

int g\_error = 0;

int dict\_trim(int fd)

{

char tc;

int has\_delim;

tc = 0;

ft\_make\_zero(&has\_delim, &g\_wasspace);

g\_isfirst = (g\_isfirst == 0) ? write(1, " ", 1) == 0 : 0;

read(fd, &tc, 1);

while (tc == ' ' || (tc == ':' && !has\_delim))

{

has\_delim = (tc == ':') ? 1 : has\_delim;

if (!read(fd, &tc, 1))

break ;

}

while (tc == ' ' && read(fd, &tc, 1))

;

while (tc != '\n' && tc != 0)

{

if (tc == ' ' && g\_wasspace == 0)

g\_wasspace = 1;

else if (tc != ' ')

ft\_putchar(tc);

if (!read(fd, &tc, 1))

break ;

}

return (1);

}

void ft\_nextline(int fd)

{

char tc;

tc = 0;

while (tc != '\n')

if (!read(fd, &tc, 1))

{

g\_error = 1;

return ;

}

}

int ft\_strlen(char \*str, int \*total)

{

int flag;

int len;

int i;

i = 0;

flag = 1;

len = 0;

while (str[i])

{

if (flag == 1 && str[i] != '0')

flag = 0;

else if (flag == 1 && str[i] == '0')

i++;

else

{

i++;

len++;

}

}

\*total = i;

return (len);

}

void dict\_print(char \*key)

{

int total;

int len;

if (ft\_strlen(key, &total) > 37 + 2)

return (print\_dict\_err());

len = total;

while (len > 0)

{

if (key[total - len] != '0')

{

if (len == 1)

dict\_printval(key[total - len]);

else if (len == 2)

dict\_printval2(key[total - len], key[total - len + 1]);

else if (len == 3)

dict\_printval3(key[total - len],

key[total - len + 1], key[total - len + 2]);

else

dict\_printvaln(total, len, key);

len = (len % 3 == 0) ? len - 3 : len - (len % 3);

}

else if (key[total - len--] == '0' && len + 1 == 1 && g\_isfirst)

dict\_printval('0');

}

}

int main(int argc, char \*\*argv)

{

if (argc == 2)

{

if (!ft\_is\_valid(argv[1]))

write(1, "Error", 5);

else

dict\_print(argv[1]);

}

else if (argc == 3)

{

if (!ft\_is\_valid(argv[2]))

write(1, "Error", 5);

else

{

g\_dictname = argv[1];

dict\_print(argv[2]);

}

}

else

write(1, "Error", 5);

if (g\_error)

print\_dict\_err();

write(1, "\n", 1);

}

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*/

/\* ::: :::::::: \*/

/\* printval.c :+: :+: :+: \*/

/\* +:+ +:+ +:+ \*/

/\* By: jongpark <jongpark@student.42seoul.kr> +#+ +:+ +#+ \*/

/\* +#+#+#+#+#+ +#+ \*/

/\* Created: 2020/12/06 11:32:00 by jongpark #+# #+# \*/

/\* Updated: 2020/12/06 13:27:56 by jongpark ### ########.fr \*/

/\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

#include "ft.h"

extern char \*g\_dictname;

extern char g\_buf[1024];

extern int g\_wasspace;

extern int g\_isfirst;

extern int g\_error;

int dict\_printval(char c)

{

int fd;

int i;

if ((fd = open(g\_dictname, O\_RDONLY)) == -1)

return (g\_error = 1);

while (g\_error != 1)

{

ft\_clear\_buf();

i = -1;

while (read(fd, &g\_buf[++i], 1))

{

if (g\_buf[i] == '\n')

i--;

if (g\_buf[i] == ' ' | g\_buf[i] == ':')

break ;

}

if (g\_buf[0] == c && (g\_buf[1] == ' ' || g\_buf[1] == ':'))

{

dict\_trim(fd);

break ;

}

else

ft\_nextline(fd);

}

return (close(fd));

}

int dict\_printval2(char c1, char c2)

{

int fd;

int i;

if ((fd = open(g\_dictname, O\_RDONLY)) == -1)

return (g\_error = 1);

while (g\_error != 1 && ft\_clear\_buf())

{

i = -1;

while (read(fd, &g\_buf[++i], 1))

if (g\_buf[i] == ' ' || g\_buf[i] == ':')

break ;

if (g\_buf[0] == c1 && (g\_buf[2] == ' ' || g\_buf[2] == ':'))

{

if ((c1 != '1' && g\_buf[1] != '0') ||

(g\_buf[0] == '1' && g\_buf[1] != c2))

continue;

dict\_trim(fd);

if (c1 != '1' && c2 != '0')

dict\_printval(c2);

break ;

}

else

ft\_nextline(fd);

}

return (close(fd));

}

int dict\_printval3(char c1, char c2, char c3)

{

int fd;

int i;

if ((fd = open(g\_dictname, O\_RDONLY)) == -1)

return (g\_error = 1);

while (g\_error != 1 && ft\_clear\_buf())

{

i = -1;

while (read(fd, &g\_buf[++i], 1))

if (g\_buf[i] == ' ' || g\_buf[i] == ':')

break ;

if (g\_buf[0] == '1' && g\_buf[1] == '0' && g\_buf[2] == '0'

&& (g\_buf[3] == ' ' || g\_buf[3] == ':'))

{

dict\_printval(c1);

if (dict\_trim(fd) && c2 != '0')

dict\_printval2(c2, c3);

else if (c2 == '0' && c3 != '0')

dict\_printval(c3);

break ;

}

else

ft\_nextline(fd);

}

return (close(fd));

}

int dict\_printpos(int len)

{

int fd;

int i;

if ((fd = open(g\_dictname, O\_RDONLY)) == -1)

return (g\_error = 1);

while (g\_error != 1)

{

ft\_clear\_buf();

i = -1;

while (read(fd, &g\_buf[++i], 1))

if (g\_buf[i] == ' ' || g\_buf[i] == ':')

break ;

if (g\_buf[0] == '1')

{

i = 1;

while (g\_buf[i] == '0')

i++;

if (i == len && dict\_trim(fd))

break ;

}

else

ft\_nextline(fd);

}

return (close(fd));

}

void dict\_printvaln(int total, int len, char \*key)

{

if (len % 3 == 0)

dict\_printval3(key[total - len],

key[total - len + 1], key[total - len + 2]);

else if (len % 3 == 1)

dict\_printval(key[total - len]);

else if (len % 3 == 2)

dict\_printval2(key[total - len], key[total - len + 1]);

if (len % 3 == 0)

dict\_printpos(len - 2);

else

dict\_printpos((len / 3) \* 3 + 1);

}

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*/

/\* ::: :::::::: \*/

/\* ft\_putchar.c :+: :+: :+: \*/

/\* +:+ +:+ +:+ \*/

/\* By: jongpark <jongpark@student.42seoul.kr> +#+ +:+ +#+ \*/

/\* +#+#+#+#+#+ +#+ \*/

/\* Created: 2020/12/06 13:26:18 by jongpark #+# #+# \*/

/\* Updated: 2020/12/06 13:28:17 by jongpark ### ########.fr \*/

/\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

#include "ft.h"

extern char \*g\_dictname;

extern char g\_buf[1024];

extern int g\_wasspace;

extern int g\_isfirst;

extern int g\_error;

void ft\_putchar(char tc)

{

if (g\_wasspace)

write(1, " ", 1);

ft\_puttc(tc, 0);

}

========MAKEFILE=========

CC = gcc

CF = -Wall -Wextra -Werror

INC = -I ./includes/

TARGET = rush-02

SRC = main.c util.c printval.c ft\_putchar.c

SRC\_DIR = ./srcs/

SRCS = $(addprefix $(SRC\_DIR), $(SRC))

OBJ = main.o util.o printval.o ft\_putchar.o

OBJ\_DIR = ./objs/

OBJS = $(addprefix $(OBJ\_DIR), $(OBJ))

all : $(TARGET)

$(OBJ\_DIR)%.o : $(SRC\_DIR)%.c

$(CC) $(CF) $(INC) -c $< -o $@

$(TARGET) : $(OBJS)

$(CC) $(CF) -o $(TARGET) $(OBJS)

clean :

rm -f $(OBJS)

fclean : clean

rm -f $(TARGET)

=========================