

Gitcon

Generated by Doxygen 1.9.5

1 Data Structure Index	1
1.1 Data Structures	1
2 File Index	3
2.1 File List	3
3 Data Structure Documentation	5
3.1 fft_config_t Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Field Documentation	5
3.1.2.1 direction	5
3.1.2.2 flags	6
3.1.2.3 input	6
3.1.2.4 output	6
3.1.2.5 size	6
3.1.2.6 twiddle_factors	6
3.1.2.7 type	6
3.2 gitcon_context_t Struct Reference	7
3.2.1 Detailed Description	7
3.2.2 Field Documentation	7
3.2.2.1 midi_handle	7
3.2.2.2 midi_queue	7
3.2.2.3 sampler	8
3.3 i2s_sampler_t Struct Reference	8
3.3.1 Detailed Description	8
3.3.2 Field Documentation	8
3.3.2.1 buffer	8
3.3.2.2 buffer_pos	9
3.3.2.3 buffer_size	9
3.3.2.4 dma_queue	9
3.3.2.5 dsp_queue	9
3.4 mcp3201_config_t Struct Reference	9
3.4.1 Detailed Description	10
3.4.2 Field Documentation	10
3.4.2.1 cs_io	10
3.4.2.2 dma_chan	10
3.4.2.3 host	10
3.4.2.4 miso_io	10
3.4.2.5 mosi_io	10
3.5 mcp3201_context_t Struct Reference	11
3.5.1 Detailed Description	11
3.5.2 Field Documentation	11
3.5.2.1 cfg	11

3.5.2.2 ongoing_transaction	11
3.5.2.3 spi	12
3.6 mcp3201_sampler_t Struct Reference	12
3.6.1 Detailed Description	12
3.6.2 Field Documentation	12
3.6.2.1 buffer	12
3.6.2.2 buffer_pos	12
3.6.2.3 buffer_size	13
3.6.2.4 dma_queue	13
3.6.2.5 dsp_queue	13
3.6.2.6 mcp_handle	13
3.7 midi_config_t Struct Reference	13
3.7.1 Detailed Description	13
3.7.2 Field Documentation	14
3.7.2.1 baudrate	14
3.7.2.2 rx_io	14
3.7.2.3 tx_io	14
3.7.2.4 uart_num	14
3.8 midi_context_t Struct Reference	15
3.8.1 Detailed Description	15
3.8.2 Field Documentation	16
3.8.2.1 cfg	16
3.9 midi_message_t Struct Reference	16
3.9.1 Detailed Description	16
3.9.2 Field Documentation	17
3.9.2.1 channel	17
3.9.2.2 param1	17
3.9.2.3 param2	17
3.9.2.4 status	17
4 File Documentation	19
4.1 include/config.h File Reference	19
4.1.1 Detailed Description	20
4.1.2 Macro Definition Documentation	20
4.1.2.1 ADC_RES	20
4.1.2.2 ADC_RES_BITS	21
4.1.2.3 AUDIO_BUFFER_SIZE	21
4.1.2.4 DMA_CHAN	21
4.1.2.5 F_SAMPLE_HZ	21
4.1.2.6 FFT_SIZE	21
4.1.2.7 FFT_WINDOW_SIZE	21
4.1.2.8 INTERNAL_ADC_CHANNEL	22

4.1.2.9 INTERNAL_ADC_IO	22
4.1.2.10 INTERNAL_ADC_UNIT	22
4.1.2.11 MIDI_BAUD	22
4.1.2.12 MIDI_RX	22
4.1.2.13 MIDI_TX	22
4.1.2.14 MIDI_UART	23
4.1.2.15 SPI_CS	23
4.1.2.16 SPI_DEV	23
4.1.2.17 SPI_MISO	23
4.1.2.18 SPI_MOSI	23
4.1.2.19 SPI_SCLK	23
4.2 config.h	24
4.3 include/processed-data.h File Reference	24
4.3.1 Variable Documentation	41
4.3.1.1 test_buffer	41
4.4 processed-data.h	58
4.5 lib/audio/i2s_sampler.c File Reference	58
4.5.1 Detailed Description	58
4.5.2 Macro Definition Documentation	59
4.5.2.1 READER_TIMEOUT_MS	59
4.5.2.2 READER_TIMEOUT_TICKS	59
4.5.2.3 RESAMPLE_DENOMINATOR	59
4.5.3 Function Documentation	59
4.5.3.1 i2s_sampler_start()	59
4.5.3.2 i2s_sampler_stop()	60
4.6 i2s_sampler.c	60
4.7 lib/audio/i2s_sampler.h File Reference	61
4.7.1 Detailed Description	62
4.7.2 Function Documentation	62
4.7.2.1 i2s_sampler_start()	62
4.7.2.2 i2s_sampler_stop()	63
4.8 i2s_sampler.h	63
4.9 lib/fft/fft.c File Reference	64
4.9.1 Macro Definition Documentation	64
4.9.1.1 LARGE_BASE_CASE	64
4.9.1.2 TWO_PI	65
4.9.1.3 USE_SPLIT_RADIX	65
4.9.2 Function Documentation	65
4.9.2.1 fft()	65
4.9.2.2 fft4()	65
4.9.2.3 fft8()	65
4.9.2.4 fft_destroy()	66

4.9.2.5 <code>fft_execute()</code>	66
4.9.2.6 <code>fft_init()</code>	66
4.9.2.7 <code>fft_primitive()</code>	66
4.9.2.8 <code>ifft()</code>	66
4.9.2.9 <code>ifft_primitive()</code>	67
4.9.2.10 <code>irfft()</code>	67
4.9.2.11 <code>rfft()</code>	67
4.9.2.12 <code>split_radix_fft()</code>	67
4.10 <code>fft.c</code>	68
4.11 <code>lib/fft/fft.h</code> File Reference	75
4.11.1 Macro Definition Documentation	76
4.11.1.1 <code>FFT_OWN_INPUT_MEM</code>	76
4.11.1.2 <code>FFT_OWN_OUTPUT_MEM</code>	76
4.11.2 Enumeration Type Documentation	76
4.11.2.1 <code>fft_direction_t</code>	76
4.11.2.2 <code>fft_type_t</code>	77
4.11.3 Function Documentation	77
4.11.3.1 <code>fft()</code>	78
4.11.3.2 <code>fft4()</code>	78
4.11.3.3 <code>fft8()</code>	78
4.11.3.4 <code>fft_destroy()</code>	78
4.11.3.5 <code>fft_execute()</code>	78
4.11.3.6 <code>fft_init()</code>	79
4.11.3.7 <code>fft_primitive()</code>	79
4.11.3.8 <code>ifft()</code>	79
4.11.3.9 <code>ifft_primitive()</code>	79
4.11.3.10 <code>irfft()</code>	80
4.11.3.11 <code>rfft()</code>	80
4.11.3.12 <code>split_radix_fft()</code>	80
4.12 <code>fft.h</code>	80
4.13 <code>lib/mcp3201/mcp3201.c</code> File Reference	81
4.13.1 Detailed Description	81
4.13.2 Typedef Documentation	82
4.13.2.1 <code>mcp3201_context_t</code>	82
4.13.3 Function Documentation	82
4.13.3.1 <code>mcp3201_exit()</code>	82
4.13.3.2 <code>mcp3201_init()</code>	82
4.13.3.3 <code>mcp3201_read()</code>	83
4.14 <code>mcp3201.c</code>	83
4.15 <code>lib/mcp3201/mcp3201.h</code> File Reference	84
4.15.1 Detailed Description	85
4.15.2 Macro Definition Documentation	85

4.15.2.1 ADC_CLK	85
4.15.3 Typedef Documentation	86
4.15.3.1 mcp3201_handle_t	86
4.15.4 Function Documentation	86
4.15.4.1 mcp3201_exit()	86
4.15.4.2 mcp3201_init()	86
4.15.4.3 mcp3201_read()	87
4.15.4.4 mcp3201_sampler_start()	87
4.15.4.5 mcp3201_sampler_stop()	88
4.16 mcp3201.h	88
4.17 lib/mcp3201/mcp3201_sampler.c File Reference	88
4.17.1 Detailed Description	89
4.17.2 Macro Definition Documentation	89
4.17.2.1 READER_TIMEOUT_MS	90
4.17.2.2 READER_TIMEOUT_TICKS	90
4.17.2.3 RESAMPLE_DENOMINATOR	90
4.17.3 Typedef Documentation	90
4.17.3.1 mcp3201_context_t	90
4.17.4 Function Documentation	90
4.17.4.1 mcp3201_sampler_start()	90
4.17.4.2 mcp3201_sampler_stop()	91
4.18 mcp3201_sampler.c	91
4.19 lib/midi/midi.c File Reference	93
4.19.1 Detailed Description	93
4.19.2 Macro Definition Documentation	94
4.19.2.1 MIDI_BYTE_SIZE_DEFAULT	94
4.19.2.2 MIDI_BYTE_SIZE_SHORT	94
4.19.3 Typedef Documentation	94
4.19.3.1 midi_context_t	94
4.19.4 Function Documentation	94
4.19.4.1 midi_exit()	94
4.19.4.2 midi_init()	95
4.19.4.3 midi_read()	95
4.19.4.4 midi_write()	95
4.20 midi.c	96
4.21 lib/midi/midi.h File Reference	98
4.21.1 Detailed Description	99
4.21.2 Macro Definition Documentation	99
4.21.2.1 MIDI_BYTE_SIZE_DEFAULT	99
4.21.2.2 MIDI_BYTE_SIZE_SHORT	99
4.21.2.3 MIDI_LOG_LEVEL	99
4.21.2.4 MIDI_PITCH_BEND_CENTER	100

4.21.2.5 MIDI_PITCH_BEND_MAX	100
4.21.2.6 MIDI_PITCH_BEND_MIN	100
4.21.3 Typedef Documentation	100
4.21.3.1 midi_handle_t	100
4.21.4 Enumeration Type Documentation	100
4.21.4.1 midi_status_t	100
4.21.5 Function Documentation	101
4.21.5.1 channel_pressure()	101
4.21.5.2 ctrl_change()	101
4.21.5.3 midi_exit()	101
4.21.5.4 midi_init()	102
4.21.5.5 midi_read()	102
4.21.5.6 midi_write()	103
4.21.5.7 note_off()	103
4.21.5.8 note_on()	103
4.21.5.9 pitch_bend()	103
4.21.5.10 poly_key_pressure()	104
4.21.5.11 prg_change()	104
4.22 midi.h	104
4.23 lib/midi/midi_utils.c File Reference	105
4.23.1 Detailed Description	105
4.23.2 Function Documentation	106
4.23.2.1 channel_pressure()	106
4.23.2.2 ctrl_change()	106
4.23.2.3 note_off()	106
4.23.2.4 note_on()	106
4.23.2.5 pitch_bend()	107
4.23.2.6 poly_key_pressure()	107
4.23.2.7 prg_change()	107
4.24 midi_utils.c	107
4.25 src/gitcon.c File Reference	108
4.25.1 Detailed Description	109
4.25.2 Macro Definition Documentation	109
4.25.2.1 CONCERT_A	109
4.25.2.2 CONCERT_A_NOTE	109
4.25.2.3 FLOAT_TO_UINT16	110
4.25.2.4 MIDI_HIGHEST_NOTE	110
4.25.2.5 MIDI_KEY_BOUNDARY	110
4.25.2.6 MIDI_LOWEST_NOTE	110
4.25.2.7 SENSITIVITY	110
4.25.2.8 UINT16_TO_FLOAT	110
4.25.3 Function Documentation	110

4.25.3.1 gitcon_exit()	110
4.25.3.2 gitcon_init()	111
4.26 gitcon.c	111
4.27 src/gitcon.h File Reference	114
4.27.1 Detailed Description	115
4.27.2 Macro Definition Documentation	115
4.27.2.1 GITCON_LOG_LEVEL	115
4.27.3 Typedef Documentation	116
4.27.3.1 gitcon_handle_t	116
4.27.4 Function Documentation	116
4.27.4.1 gitcon_exit()	116
4.27.4.2 gitcon_init()	116
4.28 gitcon.h	117
4.29 src/main.c File Reference	117
4.29.1 Detailed Description	118
4.29.2 Macro Definition Documentation	118
4.29.2.1 PROTOTYPE	118
4.29.2.2 USER_LOCAL_LEVEL	118
4.29.3 Function Documentation	118
4.29.3.1 app_main()	119
4.30 main.c	119
4.31 test/test_fft/test_fft.c File Reference	120
4.31.1 Detailed Description	121
4.31.2 Macro Definition Documentation	121
4.31.2.1 AUDIO_BUFFER_SIZE	121
4.31.2.2 F_SAMPLE_HZ	121
4.31.2.3 FFT_SIZE	121
4.31.2.4 FFT_WINDOW_SIZE	122
4.31.3 Function Documentation	122
4.31.3.1 app_main()	122
4.31.3.2 setUp()	122
4.31.3.3 tearDown()	122
4.31.3.4 test_processed_data()	122
4.32 test_fft.c	123
4.33 test/test_midi/test_midi.c File Reference	123
4.33.1 Detailed Description	124
4.33.2 Macro Definition Documentation	124
4.33.2.1 MIDI_BAUD	124
4.33.2.2 MIDI_RX	124
4.33.2.3 MIDI_TX	125
4.33.2.4 MIDI_UART	125
4.33.3 Function Documentation	125

4.33.3.1 app_main()	125
4.33.3.2 setUp()	125
4.33.3.3 tearDown()	125
4.33.3.4 test_midi_bend()	126
4.33.3.5 test_midi_bent_note()	126
4.33.3.6 test_midi_blink()	126
4.34 test_midi.c	126
Index	129

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

fft_config_t	5
gitcon_context_t	
Gitcon Configuration	7
i2s_sampler_t	
Sampler Configuration	8
mcp3201_config_t	9
mcp3201_context_t	
MCP3201 Context struct for internal use	11
mcp3201_sampler_t	12
midi_config_t	
MIDI UART Configuration	13
midi_context_t	
MIDI Context (internal! not to be accessed externally, use midi_handle_t instead)	15
midi_message_t	
MIDI Message	16

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

include/ config.h	
Gitcon Driver Configuration and Pin-Out File and dependencies	19
include/ processed-data.h	24
lib/audio/ i2s_sampler.c	
I2S Sampler Driver Source for ESP32	58
lib/audio/ i2s_sampler.h	
I2S Sampler Driver for ESP32	61
lib/fft/ fft.c	64
lib/fft/ fft.h	75
lib/mcp3201/ mcp3201.c	81
lib/mcp3201/ mcp3201.h	84
lib/mcp3201/ mcp3201_sampler.c	
MCP3201 Sampler	88
lib/midi/ midi.c	
MIDI UART Driver Source for ESP32	93
lib/midi/ midi.h	
MIDI UART Driver for ESP32	98
lib/midi/ midi_utils.c	
MIDI Message Utilities	105
src/ gitcon.c	
Gitcon Driver Source	108
src/ gitcon.h	
Gitcon Driver Header	114
src/ main.c	
Main File for Gitcon Project	117
test/test_fft/ test_fft.c	
Unit Test for FFT	120
test/test_midi/ test_midi.c	
Unit Tests for MIDI Driver	123

Chapter 3

Data Structure Documentation

3.1 `fft_config_t` Struct Reference

```
#include <fft.h>
```

Data Fields

- int [size](#)
- float * [input](#)
- float * [output](#)
- float * [twiddle_factors](#)
- [fft_type_t](#) type
- [fft_direction_t](#) direction
- unsigned int [flags](#)

3.1.1 Detailed Description

Definition at line [43](#) of file [fft.h](#).

3.1.2 Field Documentation

3.1.2.1 direction

```
fft\_direction\_t direction
```

Definition at line [50](#) of file [fft.h](#).

3.1.2.2 flags

`unsigned int flags`

Definition at line 51 of file [fft.h](#).

3.1.2.3 input

`float* input`

Definition at line 46 of file [fft.h](#).

3.1.2.4 output

`float* output`

Definition at line 47 of file [fft.h](#).

3.1.2.5 size

`int size`

Definition at line 45 of file [fft.h](#).

3.1.2.6 twiddle_factors

`float* twiddle_factors`

Definition at line 48 of file [fft.h](#).

3.1.2.7 type

[fft_type_t](#) type

Definition at line 49 of file [fft.h](#).

The documentation for this struct was generated from the following file:

- [lib/fft/fft.h](#)

3.2 gitcon_context_t Struct Reference

Gitcon Configuration.

```
#include <gitcon.h>
```

Data Fields

- [i2s_sampler_t](#) * [sampler](#)
- [midi_handle_t](#) [midi_handle](#)
- [QueueHandle_t](#) [midi_queue](#)

3.2.1 Detailed Description

Gitcon Configuration.

Parameters

<i>sampler</i>	Sampler Handler (MCP3201 or I2S)
<i>midi_handle</i>	MIDI Driver Context (MIDI over UART)
<i>midi_queue</i>	MIDI Queue Handler

Definition at line 25 of file [gitcon.h](#).

3.2.2 Field Documentation

3.2.2.1 midi_handle

```
midi\_handle\_t midi_handle
```

Definition at line 32 of file [gitcon.h](#).

3.2.2.2 midi_queue

```
QueueHandle\_t midi_queue
```

Definition at line 33 of file [gitcon.h](#).

3.2.2.3 sampler

`i2s_sampler_t* sampler`

Definition at line 30 of file [gitcon.h](#).

The documentation for this struct was generated from the following file:

- [src/gitcon.h](#)

3.3 i2s_sampler_t Struct Reference

Sampler Configuration.

```
#include <i2s_sampler.h>
```

Data Fields

- `QueueHandle_t dma_queue`
- `QueueHandle_t dsp_queue`
- `size_t * buffer`
- `size_t buffer_pos`
- `size_t buffer_size`

3.3.1 Detailed Description

Sampler Configuration.

Parameters

<i>dma_queue</i>	Samples are sent to this queue by the DMA
<i>dsp_queue</i>	Sampling result is sent to this queue
<i>buffer</i>	Buffer to store samples in
<i>buffer_pos</i>	Current position in buffer
<i>buffer_size</i>	Size of the buffer in samples

Definition at line 30 of file [i2s_sampler.h](#).

3.3.2 Field Documentation

3.3.2.1 buffer

`size_t* buffer`

Definition at line 34 of file [i2s_sampler.h](#).

3.3.2.2 buffer_pos

```
size_t buffer_pos
```

Definition at line 35 of file [i2s_sampler.h](#).

3.3.2.3 buffer_size

```
size_t buffer_size
```

Definition at line 36 of file [i2s_sampler.h](#).

3.3.2.4 dma_queue

```
QueueHandle_t dma_queue
```

Definition at line 32 of file [i2s_sampler.h](#).

3.3.2.5 dsp_queue

```
QueueHandle_t dsp_queue
```

Definition at line 33 of file [i2s_sampler.h](#).

The documentation for this struct was generated from the following file:

- [lib/audio/i2s_sampler.h](#)

3.4 mcp3201_config_t Struct Reference

```
#include <mcp3201.h>
```

Data Fields

- [spi_host_device_t](#) [host](#)
- [gpio_num_t](#) [cs_io](#)
- [gpio_num_t](#) [miso_io](#)
- [gpio_num_t](#) [mosi_io](#)
- [int](#) [dma_chan](#)

3.4.1 Detailed Description

Definition at line 36 of file [mcp3201.h](#).

3.4.2 Field Documentation

3.4.2.1 cs_io

```
gpio_num_t cs_io
```

Definition at line 39 of file [mcp3201.h](#).

3.4.2.2 dma_chan

```
int dma_chan
```

Definition at line 42 of file [mcp3201.h](#).

3.4.2.3 host

```
spi_host_device_t host
```

Definition at line 38 of file [mcp3201.h](#).

3.4.2.4 miso_io

```
gpio_num_t miso_io
```

Definition at line 40 of file [mcp3201.h](#).

3.4.2.5 mosi_io

```
gpio_num_t mosi_io
```

Definition at line 41 of file [mcp3201.h](#).

The documentation for this struct was generated from the following file:

- [lib/mcp3201/mcp3201.h](#)

3.5 mcp3201_context_t Struct Reference

MCP3201 Context struct for internal use.

Data Fields

- [mcp3201_config_t](#) *cfg*
- [spi_device_handle_t](#) *spi*
- [spi_transaction_t](#) * *ongoing_transaction*

3.5.1 Detailed Description

MCP3201 Context struct for internal use.

Parameters

<i>cfg</i>	MCP3201 Configuration
<i>spi</i>	SPI Device Handle
<i>ongoing_transaction</i>	Ongoing SPI Transaction

Definition at line 22 of file [mcp3201.c](#).

3.5.2 Field Documentation

3.5.2.1 *cfg*

[mcp3201_config_t](#) *cfg*

Definition at line 24 of file [mcp3201.c](#).

3.5.2.2 *ongoing_transaction*

[spi_transaction_t](#) * *ongoing_transaction*

Definition at line 26 of file [mcp3201.c](#).

3.5.2.3 spi

```
spi_device_handle_t spi
```

Definition at line 25 of file [mcp3201.c](#).

The documentation for this struct was generated from the following files:

- [lib/mcp3201/mcp3201.c](#)
- [lib/mcp3201/mcp3201_sampler.c](#)

3.6 mcp3201_sampler_t Struct Reference

```
#include <mcp3201.h>
```

Data Fields

- [mcp3201_handle_t](#) [mcp_handle](#)
- [QueueHandle_t](#) [dma_queue](#)
- [QueueHandle_t](#) [dsp_queue](#)
- [size_t](#) * [buffer](#)
- [size_t](#) [buffer_pos](#)
- [size_t](#) [buffer_size](#)

3.6.1 Detailed Description

Definition at line 58 of file [mcp3201.h](#).

3.6.2 Field Documentation

3.6.2.1 buffer

```
size_t* buffer
```

Definition at line 63 of file [mcp3201.h](#).

3.6.2.2 buffer_pos

```
size_t buffer_pos
```

Definition at line 64 of file [mcp3201.h](#).

3.6.2.3 buffer_size

size_t buffer_size

Definition at line 65 of file [mcp3201.h](#).

3.6.2.4 dma_queue

QueueHandle_t dma_queue

Definition at line 61 of file [mcp3201.h](#).

3.6.2.5 dsp_queue

QueueHandle_t dsp_queue

Definition at line 62 of file [mcp3201.h](#).

3.6.2.6 mcp_handle

[mcp3201_handle_t](#) mcp_handle

Definition at line 60 of file [mcp3201.h](#).

The documentation for this struct was generated from the following file:

- [lib/mcp3201/mcp3201.h](#)

3.7 midi_config_t Struct Reference

MIDI UART Configuration.

```
#include <midi.h>
```

Data Fields

- uart_port_t [uart_num](#)
- uint [baudrate](#)
- gpio_num_t [rx_io](#)
- gpio_num_t [tx_io](#)

3.7.1 Detailed Description

MIDI UART Configuration.

Parameters

<i>uart_num</i>	UART Port
<i>baudrate</i>	UART Baudrate
<i>rx_io</i>	UART RX Pin
<i>tx_io</i>	UART TX Pin

Definition at line 76 of file [midi.h](#).

3.7.2 Field Documentation

3.7.2.1 baudrate

```
uint baudrate
```

Definition at line 79 of file [midi.h](#).

3.7.2.2 rx_io

```
gpio_num_t rx_io
```

Definition at line 80 of file [midi.h](#).

3.7.2.3 tx_io

```
gpio_num_t tx_io
```

Definition at line 81 of file [midi.h](#).

3.7.2.4 uart_num

```
uart_port_t uart_num
```

Definition at line 78 of file [midi.h](#).

The documentation for this struct was generated from the following file:

- lib/midi/[midi.h](#)

3.8 midi_context_t Struct Reference

MIDI Context (internal! not to be accessed externally, use midi_handle_t instead)

Data Fields

- [midi_config_t](#) *cfg*

3.8.1 Detailed Description

MIDI Context (internal! not to be accessed externally, use midi_handle_t instead)

Parameters

<i>cfg</i>	MIDI Config
------------	-------------

Definition at line 24 of file [midi.c](#).

3.8.2 Field Documentation

3.8.2.1 *cfg*

[midi_config_t](#) *cfg*

Definition at line 26 of file [midi.c](#).

The documentation for this struct was generated from the following file:

- [lib/midi/midi.c](#)

3.9 [midi_message_t](#) Struct Reference

MIDI Message.

```
#include <midi.h>
```

Data Fields

- [uint8_t](#) [param1](#)
- [midi_status_t](#) [status](#)
- [uint8_t](#) [channel](#)
- [uint8_t](#) [param2](#)

3.9.1 Detailed Description

MIDI Message.

Parameters

<i>status</i>	MIDI Status Byte
<i>channel</i>	MIDI Channel
<i>param1</i>	MIDI Parameter 1
<i>param2</i>	MIDI Parameter 2

Definition at line 60 of file [midi.h](#).

3.9.2 Field Documentation

3.9.2.1 channel

```
uint8_t channel
```

Definition at line 64 of file [midi.h](#).

3.9.2.2 param1

```
uint8_t param1
```

Definition at line 62 of file [midi.h](#).

3.9.2.3 param2

```
uint8_t param2
```

Definition at line 65 of file [midi.h](#).

3.9.2.4 status

```
midi_status_t status
```

Definition at line 63 of file [midi.h](#).

The documentation for this struct was generated from the following file:

- [lib/midi/midi.h](#)

Chapter 4

File Documentation

4.1 include/config.h File Reference

Gitcon Driver Configuration and Pin-Out File and dependencies.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/queue.h"
#include "freertos/timers.h"
#include "driver/gpio.h"
#include "driver/spi_master.h"
#include "driver/uart.h"
#include "driver/adc.h"
#include "driver/i2s.h"
#include "esp_adc_cal.h"
#include "esp_log.h"
#include "fft.h"
#include "i2s_sampler.h"
#include "mcp3201.h"
#include "midi.h"
```

Macros

- #define [SPI_MOSI](#) (GPIO_NUM_23)
- #define [SPI_MISO](#) (GPIO_NUM_19)
- #define [SPI_SCLK](#) (GPIO_NUM_18)
- #define [SPI_CS](#) (GPIO_NUM_5)
- #define [SPI_DEV](#) (VSPI_HOST)
- #define [MIDI_UART](#) (UART_NUM_1)
- #define [MIDI_BAUD](#) (115200)
- #define [MIDI_TX](#) (GPIO_NUM_26)
- #define [MIDI_RX](#) (GPIO_NUM_27)
- #define [DMA_CHAN](#) 1

- `#define ADC_RES_BITS 12`
- `#define ADC_RES (1 << ADC_RES_BITS)`
- `#define INTERNAL_ADC_UNIT (ADC_UNIT_1)`
- `#define INTERNAL_ADC_CHANNEL (ADC_CHANNEL_5)`
- `#define INTERNAL_ADC_IO (GPIO_NUM_33)`
- `#define AUDIO_BUFFER_SIZE 512`
- `#define F_SAMPLE_HZ 15000`
- `#define FFT_WINDOW_SIZE 4`
- `#define FFT_SIZE (AUDIO_BUFFER_SIZE * FFT_WINDOW_SIZE)`

4.1.1 Detailed Description

Gitcon Driver Configuration and Pin-Out File and dependencies.

Author

@s-grundner

Version

0.1

Date

2022-12-24

Copyright

Copyright (c) 2022

Definition in file [config.h](#).

4.1.2 Macro Definition Documentation

4.1.2.1 ADC_RES

```
#define ADC_RES (1 << ADC_RES_BITS)
```

Definition at line 50 of file [config.h](#).

4.1.2.2 ADC_RES_BITS

```
#define ADC_RES_BITS 12
```

Definition at line 49 of file [config.h](#).

4.1.2.3 AUDIO_BUFFER_SIZE

```
#define AUDIO_BUFFER_SIZE 512
```

Definition at line 55 of file [config.h](#).

4.1.2.4 DMA_CHAN

```
#define DMA_CHAN 1
```

Definition at line 48 of file [config.h](#).

4.1.2.5 F_SAMPLE_HZ

```
#define F_SAMPLE_HZ 15000
```

Definition at line 56 of file [config.h](#).

4.1.2.6 FFT_SIZE

```
#define FFT_SIZE (AUDIO_BUFFER_SIZE * FFT_WINDOW_SIZE)
```

Definition at line 58 of file [config.h](#).

4.1.2.7 FFT_WINDOW_SIZE

```
#define FFT_WINDOW_SIZE 4
```

Definition at line 57 of file [config.h](#).

4.1.2.8 INTERNAL_ADC_CHANNEL

```
#define INTERNAL_ADC_CHANNEL (ADC_CHANNEL_5)
```

Definition at line 52 of file [config.h](#).

4.1.2.9 INTERNAL_ADC_IO

```
#define INTERNAL_ADC_IO (GPIO_NUM_33)
```

Definition at line 53 of file [config.h](#).

4.1.2.10 INTERNAL_ADC_UNIT

```
#define INTERNAL_ADC_UNIT (ADC_UNIT_1)
```

Definition at line 51 of file [config.h](#).

4.1.2.11 MIDI_BAUD

```
#define MIDI_BAUD (115200)
```

Definition at line 44 of file [config.h](#).

4.1.2.12 MIDI_RX

```
#define MIDI_RX (GPIO_NUM_27)
```

Definition at line 46 of file [config.h](#).

4.1.2.13 MIDI_TX

```
#define MIDI_TX (GPIO_NUM_26)
```

Definition at line 45 of file [config.h](#).

4.1.2.14 MIDI_UART

```
#define MIDI_UART (UART_NUM_1)
```

Definition at line 43 of file [config.h](#).

4.1.2.15 SPI_CS

```
#define SPI_CS (GPIO_NUM_5)
```

Definition at line 40 of file [config.h](#).

4.1.2.16 SPI_DEV

```
#define SPI_DEV (VSPI_HOST)
```

Definition at line 41 of file [config.h](#).

4.1.2.17 SPI_MISO

```
#define SPI_MISO (GPIO_NUM_19)
```

Definition at line 38 of file [config.h](#).

4.1.2.18 SPI_MOSI

```
#define SPI_MOSI (GPIO_NUM_23)
```

Definition at line 37 of file [config.h](#).

4.1.2.19 SPI_SCLK

```
#define SPI_SCLK (GPIO_NUM_18)
```

Definition at line 39 of file [config.h](#).

4.2 config.h

[Go to the documentation of this file.](#)

```

00001
00011 #ifndef CONFIG_H
00012 #define CONFIG_H
00013
00014 #include <stdio.h>
00015 #include <stdlib.h>
00016 #include <string.h>
00017 #include <math.h>
00018
00019 #include "freertos/FreeRTOS.h"
00020 #include "freertos/task.h"
00021 #include "freertos/queue.h"
00022 #include "freertos/timers.h"
00023
00024 #include "driver/gpio.h"
00025 #include "driver/spi_master.h"
00026 #include "driver/uart.h"
00027 #include "driver/adc.h"
00028 #include "driver/i2s.h"
00029 #include "esp_adc_cal.h"
00030 #include "esp_log.h"
00031
00032 #include "fft.h"
00033 #include "i2s_sampler.h"
00034 #include "mcp3201.h"
00035 #include "midi.h"
00036
00037 #define SPI_MOSI (GPIO_NUM_23)
00038 #define SPI_MISO (GPIO_NUM_19)
00039 #define SPI_SCLK (GPIO_NUM_18)
00040 #define SPI_CS (GPIO_NUM_5)
00041 #define SPI_DEV (VSPI_HOST)
00042
00043 #define MIDI_UART (UART_NUM_1)
00044 #define MIDI_BAUD (115200)
00045 #define MIDI_TX (GPIO_NUM_26)
00046 #define MIDI_RX (GPIO_NUM_27)
00047
00048 #define DMA_CHAN 1
00049 #define ADC_RES_BITS 12
00050 #define ADC_RES (1 << ADC_RES_BITS)
00051 #define INTERNAL_ADC_UNIT (ADC_UNIT_1)
00052 #define INTERNAL_ADC_CHANNEL (ADC_CHANNEL_5)
00053 #define INTERNAL_ADC_IO (GPIO_NUM_33)
00054
00055 #define AUDIO_BUFFER_SIZE 512 // Size of buffer for FFT and sampler
00056 #define F_SAMPLE_HZ 15000 // Sample rate of FFT and sampler
00057 #define FFT_WINDOW_SIZE 4 // Amount of buffers to take for FFT
00058 #define FFT_SIZE (AUDIO_BUFFER_SIZE * FFT_WINDOW_SIZE) // Amount of samples to take for FFT
00059
00060 // leave this commented out to use internal ADC
00061 // #define USE_MCP3201
00062
00063 #endif // CONFIG_H

```

4.3 include/processed-data.h File Reference

Variables

- float [test_buffer](#) [] = {5.656499999999999,5.667,5.6739999999999995,5.672499999999999,5.6655,5.↵
6525,5.638,5.6259999999999994,5.616750000000001,5.6122499999999995,5.61375,5.62275,5.641249999999999,5.↵
671,5.70675,5.7465,5.78375,5.82125,5.854,5.88375,5.90975,5.93175,5.952500000000001,5.9715,5.↵
99125,6.008249999999999,6.0264999999999995,6.0424999999999995,6.0577499999999995,6.0714999999999995,6.↵
0815,6.086749999999999,6.0815,6.063000000000001,6.0287500000000005,5.9784999999999995,5.↵
906000000000001,5.81425,5.7015,5.5665,5.407,5.22625,5.02475,4.80275,4.56775,4.31675,4.06025,3.↵
79725,3.5325,3.2715,3.01525,2.7702500000000003,2.5345,2.31175,2.10125,1.9035,1.725,1.56475,1.↵
419,1.291,1.17575,1.072,0.9820000000000001,0.9032500000000001,0.8370000000000001,0.78425,0.↵
73775,0.6965,0.66075,0.6295000000000001,0.6035,0.583,0.56375,0.54625,0.53025,0.5105000000000001,0.↵
4904999999999994,0.4700000000000003,0.4502500000000004,0.4349999999999994,0.42725,0.↵
4265,0.43575,0.4462500000000004,0.4592500000000005,0.4715,0.4845000000000004,0.5035000000000001,0.↵

52425,0.54775,0.5745,0.59825,0.6195,0.6385,0.65625,0.6737500000000001,0.6935,0.7125,0.734,0.↵
756,0.77675,0.79725,0.8109999999999999,0.8202499999999999,0.8255,0.8292499999999999,0.↵
8307500000000001,0.827,0.8170000000000001,0.8002499999999999,0.776,0.7422500000000001,0.↵
70275,0.66075,0.615,0.5660000000000001,0.51725,0.46775,0.42275,0.3807500000000003,0.34475,0.↵
3135,0.28525,0.267,0.25325,0.24725,0.2494999999999997,0.261,0.2739999999999997,0.28675,0.↵
296,0.30525,0.31425,0.32275,0.3295,0.33025,0.3242500000000004,0.312,0.2975000000000004,0.↵
28075,0.26175,0.2395,0.212,0.18625,0.1487500000000002,0.1129999999999999,0.0854999999999999,0.↵
0510000000000004,0.00775,-0.03125,-0.06025,-0.084,-0.1052500000000001,-0.122,-0.135,-0.14275,-
0.14425,-0.1420000000000002,-0.13725,-0.135,-0.1335,-0.135,-0.1420000000000002,-0.15325,-0.↵
164,-0.17625,-0.18375,-0.187,-0.18375,-0.1785,-0.1677499999999998,-0.15325,-0.1335,-0.106,-0.↵
0717500000000001,-0.032,0.00925,0.048,0.0854999999999999,0.11825,0.148,0.1770000000000002,0.↵
2014999999999998,0.22425,0.248,0.27075,0.2945000000000004,0.3212500000000004,0.34625,0.↵
37,0.3952500000000005,0.41275,0.43325,0.447,0.4669999999999997,0.48075,0.4935,0.50275,0.↵
5135,0.5235,0.53875,0.5615,0.5935,0.63775,0.69275,0.7567499999999999,0.82775,0.9039999999999999,0.↵
9820000000000001,1.06275,1.13825,1.2077499999999999,1.2710000000000001,1.323,1.36325,1.↵
39475,1.41825,1.4375,1.448,1.4535,1.4565,1.4565,1.4602499999999998,1.46475,1.47025,1.47925,1.↵
49,1.503,1.5145,1.52425,1.52975,1.52675,1.513,1.484,1.4405,1.381,1.30475,1.2115,1.1025,0.972,0.↵
82625,0.66525,0.492,0.30525,0.1129999999999999,-0.08925,-0.3045,-0.52575,-0.7567499999999999,-
0.9932500000000001,-1.23525,-1.4777500000000001,-1.72125,-1.96075,-2.1919999999999997,-2.↵
4147499999999997,-2.6252500000000003,-2.82525,-3.009,-3.1655,-3.33025,-3.47675,-3.57675,-3.↵
6734999999999998,-3.7835,-3.90325,-4.0322499999999994,-4.1595,-4.27325,-4.3685,-4.453250000000001,-
4.5295000000000005,-4.59825,-4.66375,-4.72575,-4.78275,-4.83925,-4.8950000000000005,-4.94775,-
5.0040000000000004,-5.06275,-5.1262500000000001,-5.20025,-5.2802500000000006,-5.3664999999999999,-
5.4635000000000001,-5.564,-5.667,-5.77175,-5.87,-5.96225,-6.0432500000000005,-6.112,-6.173,-6.↵
22625,-6.2767500000000001,-6.3270000000000001,-6.37675,-6.43475,-6.50175,-6.5772499999999999,-6.↵
6665,-6.7650000000000001,-6.8747500000000001,-6.9907499999999999,-7.1067500000000001,-7.219,-7.↵
32425,-7.4179999999999999,-7.5020000000000001,-7.57,-7.6255000000000001,-7.669,-7.7017500000000005,-
7.7299999999999995,-7.7552499999999999,-7.7827499999999999,-7.8155,-7.85825,-7.91175,-7.978000000000001,-
8.0605,-8.1572500000000001,-8.2702499999999999,-8.3985,-8.53875,-8.68375,-8.825750000000001,-8.↵
9545,-9.0645,-9.147,-9.195,-9.2039999999999999,-9.16525,-9.07825,-8.93925,-8.74625,-8.50225,-8.↵
2047499999999999,-7.8589999999999999,-7.46775,-7.02975,-6.5574999999999999,-6.05175,-5.523,-4.↵
9735000000000005,-4.4075000000000001,-3.83225,-3.25325,-2.67575,-2.10425,-1.54725,-1.0085,-0.↵
49525,-0.004500000000000005,0.45325,0.88425,1.28775,1.6632500000000001,2.00875,2.3307499999999997,2.↵
6237500000000002,2.8937500000000003,3.1425,3.3715,3.58275,3.7735,3.94675,4.0985000000000005,4.↵
22825,4.33725,4.42725,4.50125,4.5625,4.6157499999999999,4.6629999999999999,4.7035,4.743250000000001,4.↵
7837499999999995,4.828,4.879,4.9355,4.9965,5.0575,5.117,5.1727500000000001,5.22375,5.268,5.↵
3062499999999995,5.3405000000000005,5.36875,5.39025,5.40475,5.4129999999999999,5.4185,5.↵
42,5.426,5.4330000000000001,5.4427499999999999,5.45425,5.46725,5.481,5.497,5.51225,5.529,5.↵
545,5.5565000000000001,5.5665,5.5755,5.584,5.59225,5.6022500000000001,5.6099999999999999,5.↵
6175000000000001,5.6274999999999995,5.6427499999999995,5.66625,5.7,5.7435,5.7945,5.851,5.↵
91125,5.97525,6.03725,6.1034999999999995,6.1644999999999999,6.22175,6.26525,6.2964999999999999,6.↵
3095,6.30025,6.273,6.2247499999999999,6.157,6.0699999999999999,5.9647499999999996,5.838,5.↵
6907499999999995,5.5245000000000001,5.33975,5.1400000000000001,4.9262500000000005,4.702,4.↵
4692500000000001,4.2305,3.9855,3.743,3.50125,3.2655000000000003,3.0395,2.82375,2.6222499999999997,2.↵
4307499999999997,2.25075,2.08425,1.92875,1.786,1.6587500000000002,1.54025,1.43425,1.3375,1.↵
2475,1.162,1.085,1.0125,0.9497500000000001,0.8920000000000001,0.8415,0.7979999999999999,0.↵
7567499999999999,0.72025,0.6859999999999999,0.6537499999999999,0.62325,0.59975,0.57825,0.↵
56075,0.5485,0.54025,0.5355000000000001,0.5309999999999999,0.53025,0.534,0.5377500000000001,0.↵
54775,0.55925,0.5737500000000001,0.59125,0.6094999999999999,0.63175,0.65525,0.6819999999999999,0.↵
7095,0.7422500000000001,0.77675,0.8109999999999999,0.8444999999999999,0.8750000000000001,0.↵
898,0.91475,0.924,0.9299999999999999,0.9285,0.92175,0.91025,0.8920000000000001,0.869,0.8422499999999999,0.↵
81025,0.77675,0.74,0.7005,0.65925,0.621,0.5805,0.5425,0.505,0.4692499999999994,0.4440000000000006,0.↵
42275,0.409,0.402,0.4035,0.415,0.4295,0.4485,0.4692499999999994,0.4882499999999996,0.505,0.↵
518,0.521,0.518,0.5035000000000001,0.4822500000000007,0.4525,0.418,0.3807500000000003,0.↵
3455000000000003,0.3105,0.27775,0.25025,0.2227499999999998,0.1992499999999998,0.1807500000000002,0.↵
1685,0.16025,0.154,0.145,0.1342499999999998,0.11975,0.1015,0.0785,0.052,0.0212499999999998,-
0.0085,-0.0375,-0.061,-0.0825,-0.0992499999999999,-0.1115000000000002,-0.11975,-0.1242499999999999,-
0.1235,-0.1242499999999999,-0.11975,-0.11525,-0.109,-0.1,-0.08925,-0.07475,-0.0595000000000004,-

0.0375,-0.00925,0.02225,0.058750000000000004,0.097,0.13725,0.175500000000000002,0.21125,0.↵
2425,0.27,0.296,0.31825,0.33575,0.34875,0.35625,0.351000000000000003,0.331,0.3205,0.31825,0.↵
3175,0.3165,0.32575,0.345500000000000003,0.37374999999999997,0.4075,0.45175,0.502,0.55925,0.↵
618,0.6805,0.7454999999999999,0.808,0.8645,0.9169999999999999,0.966,1.0125,1.0545,1.0925,1.↵
1245,1.155,1.184,1.21625,1.2489999999999999,1.28625000000000001,1.3215,1.358,1.39225,1.42825,1.↵
46175,1.49525,1.525,1.55575,1.58,1.603,1.62575,1.6432499999999999,1.661,1.6740000000000002,1.↵
68,1.68,1.67,1.645,1.60525,1.54275,1.45725,1.349,1.2185,1.065,0.89350000000000001,0.7035,0.5005,0.↵
2885,0.06874999999999999,-0.15025,-0.3715,-0.5904999999999999,-0.808,-1.023,-1.236,-1.4435,-1.↵
6495,-1.85625,-2.0629999999999997,-2.27125,-2.47425,-2.67175000000000003,-2.8625000000000003,-
3.03800000000000003,-3.2005,-3.34775,-3.4775,-3.59875,-3.707,-3.804,-3.8925,-3.97025,-4.04575,-4.↵
12225,-4.20075,-4.28475,-4.3725,-4.4632499999999995,-4.557,-4.65475,-4.74925,-4.84775,-4.94075,-5.↵
03225,-5.11925,-5.1979999999999995,-5.26725,-5.33075,-5.38625,-5.43825,-5.49175,-5.54275,-5.59925,-
5.6572499999999994,-5.72275000000000004,-5.795249999999999,-5.88,-5.973,-6.0760000000000005,-
6.1845,-6.29875,-6.41475,-6.529999999999999,-6.638249999999999,-6.7475,-6.84975,-6.9215,-6.99775,-
7.08625,-7.1480000000000001,-7.1754999999999995,-7.17850000000000005,-7.1677500000000001,-7.↵
16175,-7.170749999999999,-7.1975,-7.2425000000000001,-7.30350000000000005,-7.3800000000000001,-
7.46925,-7.57450000000000005,-7.691249999999999,-7.821,-7.967499999999999,-8.12,-8.28775,-8.↵
464,-8.64475,-8.8265,-9.002,-9.163,-9.30475,-9.41925,-9.501,-9.5435,-9.54125,-9.49325,-9.39325,-9.↵
2415,-9.034,-8.7775,-8.4725,-8.126,-7.74075,-7.32575,-6.8839999999999995,-6.420249999999999,-5.↵
93650000000000006,-5.436,-4.92700000000000005,-4.41125,-3.891,-3.373,-2.8535,-2.336,-1.825,-1.32375,-
0.8332499999999999,-0.35925,0.09375,0.525,0.92925,1.307,1.6555,1.9775,2.27125,2.53675,2.7725,2.↵
9884999999999997,3.18,3.35475,3.5132499999999998,3.6575,3.7895,3.91475,4.0305,4.140499999999999,4.↵
245,4.34725,4.444,4.53725,4.628,4.7135000000000001,4.7905000000000001,4.8607499999999995,4.↵
922499999999999,4.97425000000000005,5.02325,5.0652500000000001,5.104,5.13925,5.1720000000000001,5.↵
20625,5.239,5.27125,5.303999999999999,5.336749999999999,5.365,5.396249999999999,5.423,5.↵
4465,5.4695,5.481,5.489999999999999,5.4947500000000001,5.4925,5.489249999999999,5.48325,5.↵
4765,5.47175,5.471,5.46725,5.46875,5.47325,5.48325,5.49925,5.5237500000000001,5.55725,5.59775,5.↵
6480000000000001,5.70750000000000005,5.7725,5.8450000000000001,5.9220000000000001,6.005,6.↵
089,6.17525,6.256,6.33250000000000005,6.398,6.4475000000000001,6.482,6.49575,6.4879999999999995,6.↵
4612500000000001,6.4094999999999995,6.337,6.240749999999999,6.12250000000000005,5.9845000000000001,5.↵
827999999999999,5.655749999999999,5.46575,5.26425000000000005,5.05225,4.8355000000000001,4.↵
614999999999999,4.39525,4.17625000000000005,3.95825,3.7445,3.5385,3.3385,3.1454999999999997,2.↵
9579999999999997,2.774,2.59775,2.42925,2.2697499999999997,2.1165,1.97225,1.83725,1.709,1.↵
58925,1.48075,1.377,1.284,1.19925,1.12525,1.05825,0.99875,0.94225,0.88725,0.83625,0.78575,0.↵
73775,0.6905,0.6477499999999999,0.60725,0.57375000000000001,0.5455,0.5225,0.50275,0.492,0.↵
483,0.4815,0.483,0.48975,0.5005,0.51725,0.53475,0.557,0.57975,0.60575,0.63625,0.6697500000000001,0.↵
7072499999999999,0.7437499999999999,0.78125,0.8125,0.8422499999999999,0.86300000000000001,0.↵
87825000000000001,0.8865,0.88275,0.8734999999999999,0.85225000000000001,0.82175000000000001,0.↵
78275,0.73625,0.6842499999999999,0.631,0.58125,0.5369999999999999,0.502,0.473,0.45175,0.↵
4425,0.440250000000000003,0.4425,0.44925,0.45475000000000004,0.45925000000000005,0.4607499999999994,0.↵
4599999999999996,0.45625,0.45025000000000004,0.4394999999999995,0.4287499999999996,0.↵
4157499999999995,0.4004999999999997,0.38675,0.3707499999999997,0.35925,0.34550000000000003,0.↵
3334999999999996,0.32275,0.30825,0.29675,0.28300000000000003,0.26775000000000004,0.25025,0.↵
23275,0.21125,0.1885,0.16475,0.14125,0.11825,0.09375,0.07175000000000001,0.05100000000000004,0.↵
029750000000000002,0.00925,-0.01075,-0.03275,-0.05425,-0.0785,-0.10225,-0.12825,-0.158,-0.1855,-
0.216750000000000003,-0.24424999999999997,-0.26625,-0.28300000000000003,-0.29450000000000004,-
0.297500000000000004,-0.29524999999999996,-0.2845,-0.26549999999999996,-0.2395,-0.20525000000000002,-
0.1655,-0.122750000000000001,-0.07625,-0.0335,0.0015,0.03424999999999996,0.059500000000000004,0.↵
0785,0.093,0.1,0.1015,0.1,0.097,0.0915,0.09075,0.09225,0.0985,0.10975,0.126,0.14575,0.1677499999999998,0.↵
19525,0.22499999999999998,0.2525,0.28075,0.31275000000000003,0.34550000000000003,0.38525,0.↵
42874999999999996,0.47600000000000003,0.5287499999999999,0.586,0.6455,0.70575,0.7675000000000001,0.↵
82625,0.882,0.9285,0.96900000000000001,1.001,1.02475,1.04225,1.0605,1.08025,1.101,1.13075,1.↵
167250000000000001,1.21525,1.2725,1.3405,1.4160000000000001,1.496,1.57625,1.654,1.7235,1.78675,1.↵
8365,1.87225,1.89675000000000002,1.905,1.9064999999999999,1.90575,1.879,1.83175,1.77,1.702,1.↵
6219999999999999,1.52675,1.42125,1.3015,1.1657499999999998,1.014,0.84525,0.6629999999999999,0.↵
47000000000000003,0.2595,0.039,-0.18475,-0.4135,-0.644,-0.8704999999999999,-1.0955,-1.3085,-1.↵
5105,-1.70450000000000001,-1.88525,-2.05525,-2.21475,-2.3635,-2.5025,-2.636,-2.765,-2.8907499999999997,-
3.0175,-3.14325,-3.27,-3.39725,-3.52475,-3.65375,-3.77875,-3.8985,-4.01075,-4.1122499999999995,-

4.20925,-4.293,-4.36775,-4.4350000000000005,-4.4952499999999995,-4.55175,-4.60425,-4.654,-4.↵
708749999999999,-4.76675,-4.8294999999999995,-4.901,-4.9805,-5.0637500000000001,-5.156000000000001,-
5.25275,-5.3572500000000005,-5.463500000000001,-5.57625,-5.6907499999999995,-5.806,-5.9205,-
6.0302500000000006,-6.1355,-6.229500000000001,-6.321000000000001,-6.398,-6.462,-6.5125,-6.↵
5482499999999995,-6.575749999999999,-6.59475,-6.6055,-6.61175,-6.61625,-6.62225,-6.639749999999999,-
6.668749999999999,-6.717,-6.789499999999999,-6.88175,-7.00375,-7.14425,-7.309,-7.492749999999999,-
7.695,-7.906249999999999,-8.1275,-8.3495,-8.56925,-8.7775,-8.9745,-9.146,-9.298,-9.41625,-9.50475,-
9.5605,-9.581000000000001,-9.5665,-9.50925,-9.414,-9.278749999999999,-9.1035,-8.892750000000001,-
8.644,-8.3595,-8.0405,-7.6875,-7.30125,-6.8839999999999995,-6.43625,-5.954000000000001,-5.471,-4.↵
96675,-4.42725,-3.89325,-3.3845,-2.8945,-2.427,-1.9737500000000001,-1.52675,-1.081,-0.648500000000001,-
0.2350000000000001,0.1565,0.52425,0.865999999999999,1.18725,1.48775,1.76925,2.03325,2.↵
279,2.51225,2.729749999999999,2.93275,3.1205,3.2975,3.45925,3.6065,3.743,3.868,3.984,4.093249999999999,4.↵
19475,4.29,4.38225,4.4677500000000006,4.548,4.62025,4.6905,4.754,4.8125,4.87125,4.9285000000000005,4.↵
98125,5.033,5.079000000000001,5.124750000000001,5.16825,5.20775,5.242249999999999,5.272749999999999,5.↵
30025,5.319249999999999,5.33,5.335249999999999,5.33,5.3214999999999995,5.3062499999999995,5.↵
288749999999999,5.2705,5.25125,5.234500000000001,5.220000000000001,5.2124999999999995,5.↵
20925,5.2170000000000005,5.2330000000000005,5.2620000000000005,5.304749999999999,5.362,5.↵
4305,5.51225,5.60375,5.7052499999999995,5.811999999999999,5.921250000000001,6.0295000000000005,6.↵
1355,6.2355,6.32175,6.39575,6.4515,6.4902500000000005,6.508750000000001,6.50775,6.4872499999999995,6.↵
44525,6.386500000000001,6.3125,6.218,6.108750000000001,5.986,5.846500000000001,5.6937500000000005,5.↵
52975,5.35425,5.1705000000000005,4.9775,4.7805,4.57925,4.37325,4.16875,3.96725,3.765749999999999,3.↵
57125,3.379,3.1945,3.016,2.8457500000000002,2.68475,2.5315,2.38275,2.2407500000000002,2.↵
10425,1.973,1.8485,1.7295,1.61975,1.516,1.4205,1.3335,1.25125,1.17875,1.11075,1.049,0.99025,0.↵
936999999999999,0.882,0.82625,0.7735,0.718,0.66525,0.614250000000001,0.56675,0.5295,0.↵
50425,0.48975,0.486,0.4935,0.5135,0.54475,0.58375,0.629500000000001,0.6775,0.72475,0.77125,0.↵
81325,0.846749999999999,0.875749999999999,0.894250000000001,0.903999999999999,0.↵
9055000000000001,0.901749999999999,0.890250000000001,0.872,0.852250000000001,0.82775,0.↵
8057500000000001,0.7835,0.763,0.741499999999999,0.72175,0.70425,0.686750000000001,0.↵
6697500000000001,0.653,0.63775,0.624749999999999,0.612,0.6005,0.590499999999999,0.5805,0.↵
573,0.566000000000001,0.5615,0.557,0.553999999999999,0.552249999999999,0.547,0.54325,0.↵
53625,0.5295,0.518,0.5065,0.492749999999999,0.4760000000000003,0.45775,0.4402500000000003,0.↵
42125,0.4012499999999994,0.3845,0.36775,0.3540000000000004,0.344,0.3365,0.331,0.32575,0.↵
322,0.312,0.296,0.27625,0.2487500000000003,0.2145000000000002,0.17475,0.12975,0.080749999999999,0.↵
0305,-0.0205,-0.068749999999999,-0.108249999999999,-0.14275,-0.16475,-0.18,-0.187,-0.1855,-
0.18,-0.17175,-0.16025,-0.148,-0.136499999999999,-0.125,-0.114499999999999,-0.1015,-0.08625,-
0.0710000000000001,-0.0495,-0.028249999999999,-0.004500000000000005,0.01975,0.0427499999999996,0.↵
064,0.08,0.09075,0.0945,0.09225,0.084,0.07325,0.061,0.0465,0.0357500000000004,0.0282499999999997,0.↵
02975000000000002,0.039,0.0572499999999995,0.085499999999999,0.122,0.1677499999999998,0.↵
21825,0.27225,0.3265,0.3777500000000003,0.42725,0.4715,0.508,0.536999999999999,0.5645,0.↵
58525,0.60125,0.61875,0.63625,0.65625,0.67975,0.7095,0.742250000000001,0.78275,0.82775,0.↵
875749999999999,0.9285,0.9865,1.043,1.10625,1.17025,1.23525,1.30225,1.37025,1.436499999999999,1.↵
5015,1.56775,1.632750000000001,1.69675,1.7585,1.818749999999999,1.873,1.91725,1.95,1.968500000000001,1.↵
97,1.9515,1.9105,1.84625,1.7607500000000003,1.6555,1.5305,1.384,1.2215,1.046,0.859,0.6715,0.↵
4775,0.28775,0.097,-0.09375,-0.2845,-0.47075,-0.657750000000001,-0.84075,-1.02225,-1.20075,-1.↵
37875,-1.55575,-1.73425,-1.9105,-2.085,-2.2560000000000002,-2.4245,-2.5895,-2.748750000000002,-
2.9059999999999997,-3.05475,-3.19525,-3.32875,-3.456,-3.57425,-3.6835,-3.785,-3.878,-3.95975,-4.↵
039750000000001,-4.123,-4.186249999999999,-4.2365,-4.29075,-4.348,-4.406750000000001,-4.47225,-
4.5425,-4.6235,-4.71575,-4.81725,-4.9255,-5.043,-5.171250000000001,-5.3062499999999995,-5.44575,-
5.5785,-5.7082500000000005,-5.830500000000001,-5.94025,-6.03725,-6.1187499999999995,-6.1835,-
6.2325,-6.26375,-6.278250000000001,-6.2835,-6.279,-6.273,-6.261500000000001,-6.255249999999999,-
6.255249999999999,-6.26375,-6.289,-6.32925,-6.392,-6.4750000000000005,-6.58025,-6.704000000000001,-
6.845750000000001,-7.002999999999999,-7.1762500000000005,-7.364750000000001,-7.56525,-7.↵
77275,-7.9857499999999995,-8.200000000000001,-8.413,-8.62725,-8.830250000000001,-9.027000000000001,-
9.20875,-9.36975,-9.50925,-9.62,-9.701500000000001,-9.7465,-9.75275,-9.7205,-9.64575,-9.52825,-9.↵
3735,-9.17575,-8.941,-8.67,-8.36325,-8.0285,-7.66675,-7.274750000000001,-6.861000000000001,-6.↵
4285,-5.9784999999999995,-5.51825,-5.0475,-4.5714999999999995,-4.0969999999999995,-3.627,-3.↵
15775,-2.697,-2.24525,-1.80675,-1.384,-0.975,-0.58525,-0.21125,0.1465,0.4845000000000004,0.↵
80875,1.11625,1.4075,1.684499999999999,1.9455,2.19125,2.42375,2.6397500000000003,2.84275,3.↵

02975,3.20575,3.3715,3.5255,3.6705,3.8055,3.9322500000000002,4.0497499999999995,4.1602500000000004,4.↵
263999999999999,4.364,4.46025,4.553999999999999,4.64475,4.7325,4.818,4.8987500000000001,4.↵
967499999999999,5.0500000000000001,5.114,5.15125,5.20025,5.2604999999999995,5.323,5.3765,5.↵
4145,5.426,5.41925,5.3955,5.36125,5.3245000000000005,5.285,5.2459999999999996,5.2147499999999996,5.↵
1887500000000001,5.17425,5.1735000000000001,5.185,5.20925,5.2467500000000005,5.2962500000000001,5.↵
36125,5.4345,5.5175,5.6114999999999995,5.71125,5.815,5.92425,6.03325,6.14325,6.24925,6.35,6.↵
4430000000000005,6.52775,6.5972500000000001,6.650499999999999,6.6872500000000001,6.70775,6.↵
70775,6.6850000000000005,6.64525,6.587999999999999,6.514,6.4247499999999995,6.3225000000000001,6.↵
209499999999999,6.0845,5.95025,5.806,5.655,5.4932500000000001,5.3245000000000005,5.1475,4.↵
9597500000000005,4.7692499999999995,4.569249999999999,4.3685,4.165,3.96125,3.7605,3.5675,3.↵
37975,3.2035,3.0364999999999998,2.88325,2.7390000000000003,2.60475,2.4802500000000003,2.↵
366,2.253,2.1445,2.0385,1.931,1.822,1.7105,1.59525,1.48075,1.36575,1.2565,1.1520000000000001,1.↵
05125,0.9575,0.8734999999999999,0.8024999999999999,0.7437499999999999,0.6965,0.6645,0.↵
6425000000000001,0.63025,0.62625,0.63175,0.644,0.66,0.68125,0.7012499999999999,0.724,0.747,0.↵
7667499999999999,0.78275,0.79725,0.808,0.8177500000000001,0.8217500000000001,0.82475,0.↵
8255,0.8202499999999999,0.81475,0.8049999999999999,0.7927500000000001,0.7775,0.759,0.737,0.↵
71475,0.692,0.6675,0.64475,0.6224999999999999,0.6035,0.59125,0.58125,0.5737500000000001,0.↵
573,0.5715,0.573,0.5715,0.5685,0.55925,0.5455,0.528,0.5065,0.4815,0.45625,0.434,0.41125,0.39375,0.↵
3815,0.3737499999999999,0.37225,0.3752500000000003,0.3807500000000003,0.38975,0.39675,0.↵
39975,0.3989999999999999,0.393,0.38225,0.3655,0.3425,0.3175,0.29075,0.2595,0.2305,0.2022499999999999,0.↵
174,0.1487500000000002,0.1227500000000001,0.09525,0.06475,0.032,-0.00375,-0.0435,-0.0847499999999999,-
0.125,-0.1662499999999998,-0.2022499999999999,-0.2350000000000001,-0.25875,-0.2739999999999997,-
0.2799999999999997,-0.2755,-0.2625,-0.2425,-0.2145000000000002,-0.18375,-0.15025,-0.11525,-0.↵
08625,-0.06175,-0.0442500000000005,-0.03275,-0.03125,-0.0357500000000004,-0.04575,-0.06025,-
0.074,-0.09075,-0.1052500000000001,-0.1175000000000001,-0.122,-0.11825,-0.1082499999999999,-
0.093,-0.0694999999999999,-0.04125,-0.01,0.0245,0.0572499999999995,0.08925,0.1227500000000001,0.↵
1525,0.1807500000000002,0.2060000000000002,0.2312499999999998,0.254,0.2785,0.3045,0.↵
33025,0.35925,0.39225,0.4250000000000004,0.4599999999999996,0.4927499999999997,0.5287499999999999,0.↵
55925,0.5875,0.61575,0.64325,0.6697500000000001,0.6965,0.7255,0.76,0.7995,0.8507499999999999,0.↵
911,0.9820000000000001,1.06575,1.15975,1.25875,1.3665,1.47175,1.57925,1.6844999999999999,1.↵
7815,1.86925,1.9485,2.01425,2.0660000000000003,2.102,2.1195,2.12175,2.105,2.0705,2.0187500000000003,1.↵
9522500000000003,1.8685,1.7722499999999999,1.6595,1.53725,1.4045,1.262,1.11225,0.9575,0.↵
7965,0.63325,0.46775,0.29675,0.126,-0.04725,-0.22125,-0.3989999999999997,-0.57975,-0.76225,-0.↵
9490000000000001,-1.139,-1.32975,-1.52275,-1.7157499999999999,-1.90425,-2.09275,-2.2735,-2.↵
4475000000000002,-2.6100000000000003,-2.758,-2.896,-3.019,-3.128,-3.225,-3.3105,-3.38275,-3.4485,-
3.50725,-3.5645000000000002,-3.62325,-3.68425,-3.753,-3.8315,-3.92225,-4.02675,-4.1427499999999995,-
4.271,-4.40825,-4.551,-4.69525,-4.8415,-4.9827499999999999,-5.12,-5.2497500000000001,-5.3695,-5.↵
47725,-5.56875,-5.6457500000000005,-5.71125,-5.7625,-5.80375,-5.83275,-5.85325,-5.86325,-5.86625,-
5.86775,-5.864,-5.864,-5.86625,-5.8770000000000001,-5.8930000000000001,-5.9220000000000001,-5.↵
9655,-6.0280000000000005,-6.1065000000000005,-6.20725,-6.32625,-6.46825,-6.6292500000000001,-
6.8092500000000005,-7.009,-7.21125,-7.4179999999999999,-7.6439999999999999,-7.8865000000000001,-
8.12825,-8.3625,-8.593,-8.81875,-9.0317499999999999,-9.224,-9.39875,-9.5475000000000001,-9.6695000000000001,-
9.7595000000000001,-9.81675,-9.8419999999999999,-9.8267499999999999,-9.77325,-9.6847500000000001,-
9.5565,-9.39175,-9.19425,-8.9599999999999999,-8.6945,-8.397,-8.0735,-7.7225,-7.3539999999999999,-6.↵
96575,-6.56125,-6.1455,-5.7189999999999999,-5.2887499999999999,-4.8522500000000001,-4.4152499999999995,-
3.9787500000000002,-3.5455,-3.1157500000000002,-2.6932500000000004,-2.27825,-1.87225,-1.4785,-
1.0925,-0.7224999999999999,-0.36475,-0.01975,0.30825,0.6195,0.9155,1.191,1.4495,1.6885000000000001,1.↵
9120000000000001,2.1195,2.314,2.497,2.6687499999999997,2.8335,2.99375,3.148,3.29975,3.4515,3.↵
5995,3.7475,3.89325,4.03125,4.1640000000000001,4.29225,4.4105,4.51975,4.621,4.71125,4.79125,4.↵
8585,4.9117500000000005,4.9522499999999999,4.9827499999999999,5.0025,5.01325,5.0147499999999999,5.↵
00575,4.988,4.9645,4.9362499999999999,4.90275,4.8705,4.8425,4.8202500000000001,4.8065,4.8050000000000001,4.↵
8149999999999995,4.8415,4.885,4.947,5.0225,5.1095000000000001,5.2109999999999999,5.3199999999999999,5.↵
43375,5.54875,5.6640000000000001,5.77325,5.87925,5.9784999999999995,6.0714999999999995,6.↵
1585,6.23775,6.3072500000000001,6.3705,6.424,6.46675,6.49725,6.5185,6.53375,6.5407499999999999,6.↵
5095,6.4797499999999999,6.4537499999999999,6.3905,6.28825,6.1530000000000005,5.98975,5.81825,5.↵
6465000000000005,5.4795000000000001,5.3122500000000001,5.1422500000000001,4.9674999999999999,4.↵
79275,4.6164999999999999,4.44325,4.27325,4.1085,3.949,3.794,3.6475,3.50425,3.366,3.22875,3.↵
09225,2.95475,2.8205,2.681,2.5382499999999997,2.39575,2.253,2.1125000000000003,1.976,1.84325,1.↵

7175,1.6,1.4869999999999999,1.3847500000000001,1.28925,1.20175,1.123,1.04975,0.988,0.9322499999999999,0.↵
88575,0.8507499999999999,0.82625,0.81175,0.8064999999999999,0.8109999999999999,0.8210000000000001,0.↵
84075,0.8619999999999999,0.88575,0.9095000000000001,0.9285,0.94225,0.9530000000000001,0.↵
9545,0.9497500000000001,0.93925,0.9247500000000001,0.9095000000000001,0.8942500000000001,0.↵
8789999999999999,0.8675,0.85825,0.8574999999999999,0.8612500000000001,0.8637499999999999,0.↵
869,0.8734999999999999,0.8765000000000001,0.8765000000000001,0.87125,0.8652500000000001,0.↵
85375,0.8370000000000001,0.8162499999999999,0.79425,0.7705,0.75075,0.73175,0.7165,0.70575,0.↵
698,0.6935,0.69125,0.68825,0.6859999999999999,0.6842499999999999,0.68125,0.67675,0.6697500000000001,0.↵
66,0.65225,0.64475,0.63475,0.6287499999999999,0.62325,0.6195,0.62025,0.62175,0.624,0.6247499999999999,0.↵
62175,0.6165,0.60275,0.58375,0.55775,0.5225,0.48075,0.43325,0.3807500000000003,0.32875,0.↵
27225,0.2167500000000003,0.167,0.122,0.084,0.055,0.03275,0.0205,0.0174999999999999,0.↵
0212499999999999,0.029,0.03825,0.0495,0.06025,0.06725,0.07325,0.074,0.0725,0.0687499999999999,0.↵
06325,0.0564999999999999,0.05025,0.04725,0.0442500000000005,0.04125,0.03825,0.0357500000000004,0.↵
0349999999999999,0.0357500000000004,0.0349999999999999,0.0335,0.0275,0.0245,0.02225,0.↵
019,0.0212499999999999,0.0282499999999999,0.03975,0.0572499999999999,0.0755,0.10225,0.↵
132,0.16475,0.2,0.2365,0.2739999999999999,0.30525,0.33575,0.35775,0.377,0.38975,0.39975,0.↵
40425,0.4075,0.40825,0.41125,0.4135,0.41875,0.4287499999999999,0.4454999999999999,0.4684999999999999,0.↵
4975000000000005,0.534,0.5805,0.6355000000000001,0.7035,0.779,0.8619999999999999,0.9575,1.↵
05825,1.162,1.27175,1.381,1.493,1.6007500000000001,1.703,1.79825,1.8815,1.9522500000000003,2.↵
01025,2.053,2.0805000000000002,2.092,2.08825,2.06825,2.0355000000000003,1.9875,1.9295,1.↵
8599999999999999,1.78,1.6945000000000001,1.60375,1.503,1.3992499999999999,1.2885,1.17025,1.↵
04375,0.9055000000000001,0.7567499999999999,0.5975,0.4235,0.241,0.0465,-0.1565,-0.36775,-0.583,-
0.8017500000000001,-1.0170000000000001,-1.22825,-1.429,-1.62125,-1.8005,-1.9675000000000002,-
2.121,-2.25975,-2.38575,-2.501,-2.60475,-2.70075,-2.78925,-2.8725,-2.9517499999999997,-3.02975,-
3.1075,-3.18975,-3.279,-3.373,-3.476,-3.58725,-3.707,-3.83675,-3.97175,-4.1145000000000005,-4.↵
26175,-4.40825,-4.55775,-4.705,-4.8462499999999999,-4.9805,-5.1069999999999999,-5.22075,-5.↵
3207499999999999,-5.40775,-5.4795000000000001,-5.5397500000000001,-5.58325,-5.6175000000000001,-
5.635,-5.6442499999999999,-5.645,-5.6404999999999999,-5.62975,-5.62125,-5.6129999999999999,-
5.60925,-5.61375,-5.6282499999999999,-5.6525,-5.6922500000000005,-5.748,-5.82425,-5.918,-6.↵
0295000000000005,-6.1622499999999999,-6.3147499999999999,-6.485,-6.6750000000000001,-6.88025,-
7.1,-7.33025,-7.5707499999999999,-7.8155,-8.0635,-8.30925,-8.55025,-8.7815000000000001,-8.9997499999999999,-
9.2024999999999999,-9.3865,-9.54275,-9.6755,-9.7755,-9.8434999999999999,-9.87475,-9.87325,-9.835,-
9.7640000000000001,-9.658,-9.5215,-9.352,-9.156,-8.9294999999999999,-8.6785,-8.4029999999999999,-
8.10325,-7.7842499999999999,-7.44475,-7.08625,-6.7100000000000001,-6.32175,-5.91575,-5.49925,-
5.07425,-4.6425,-4.20925,-3.778,-3.35,-2.929,-2.52075,-2.1239999999999997,-1.73725,-1.377,-1.↵
0345,-0.6995,-0.3715,-0.06475,0.2235,0.50575,0.77525,1.0285,1.27175,1.5015,1.72575,1.9395,2.↵
1462499999999999,2.34525,2.5382499999999999,2.7305,2.9212499999999999,3.11125,3.2952500000000002,3.↵
4805,3.6605,3.83825,4.00475,4.1617500000000005,4.3075,4.4372500000000001,4.55175,4.64925,4.↵
7287500000000001,4.7882500000000001,4.8294999999999999,4.8507500000000001,4.8599999999999999,4.↵
85675,4.84325,4.82625,4.8042500000000001,4.78275,4.76525,4.75075,4.7425000000000001,4.7432500000000001,4.↵
7485,4.763,4.7844999999999999,4.8142499999999999,4.8515000000000001,4.8965000000000005,4.↵
947,5.008,5.0757499999999999,5.14825,5.2292499999999999,5.31625,5.40775,5.50775,5.6122499999999999,5.↵
722,5.83575,5.9532500000000001,6.0699999999999999,6.18675,6.2979999999999999,6.4025,6.4935000000000001,6.↵
5705,6.6307500000000001,6.6720000000000001,6.6917500000000001,6.694,6.6720000000000001,6.↵
6307500000000001,6.572,6.49875,6.4094999999999999,6.3057500000000001,6.192,6.0722499999999999,5.↵
9455,5.815,5.6785000000000005,5.53825,5.3925,5.2437499999999999,5.0917499999999999,4.9399999999999999,4.↵
78525,4.63025,4.4769999999999999,4.322,4.1695,4.02075,3.87125,3.727,3.582,3.4385,3.29275,3.↵
1425,2.9884999999999999,2.8320000000000003,2.6710000000000003,2.5085,2.3467499999999997,2.↵
1835,2.02875,1.87825,1.73875,1.61525,1.49925,1.4000000000000001,1.316,1.24275,1.17875,1.1215,1.↵
08725,1.043,0.98725,0.9575,0.94525,0.94525,0.9497500000000001,0.9497500000000001,0.9369999999999999,0.↵
91775,0.898,0.87975,0.8645,0.85375,0.8492500000000001,0.8467499999999999,0.8484999999999999,0.↵
8500000000000001,0.853,0.8522500000000001,0.8507499999999999,0.8492500000000001,0.84,0.↵
83,0.8162499999999999,0.8035000000000001,0.7872499999999999,0.7735,0.7615000000000001,0.↵
75,0.74075,0.7325,0.7270000000000001,0.72025,0.71575,0.70875,0.69875,0.68975,0.67675,0.65925,0.↵
6385,0.61575,0.59125,0.5685,0.54325,0.5225,0.50725,0.4967499999999997,0.4945,0.4989999999999999,0.↵
509,0.52575,0.541,0.56075,0.5775,0.58975,0.59825,0.599,0.592,0.57975,0.55925,0.53025,0.4982499999999997,0.↵
4584999999999999,0.41875,0.377,0.3334999999999999,0.293,0.25325,0.2145000000000002,0.↵
1807500000000002,0.1472500000000002,0.116,0.08625,0.058,0.0297500000000002,0.00375,-0.↵

019,-0.04125,-0.058,-0.0725,-0.08074999999999999,-0.08625,-0.08625,-0.0825,-0.0785,-0.07025,-0.↵
06325,-0.05575000000000001,-0.052,-0.04874999999999999,-0.05100000000000004,-0.05425,-0.↵
06325,-0.07475,-0.08850000000000001,-0.106,-0.1205,-0.13649999999999998,-0.15175,-0.161,-0.↵
167,-0.1725,-0.1685,-0.161,-0.148,-0.12675,-0.1045,-0.07475,-0.042,-0.00675,0.0275,0.064,0.0985,0.↵
12975,0.158,0.183,0.20375,0.2205,0.232,0.2395,0.24025000000000002,0.24175,0.23800000000000002,0.↵
23124999999999998,0.22499999999999998,0.219,0.216,0.21675000000000003,0.21974999999999997,0.↵
232,0.25175000000000003,0.28375,0.32725,0.3845,0.4555,0.5369999999999999,0.631,0.7355,0.↵
84925000000000001,0.96675000000000001,1.088,1.20925000000000002,1.3245,1.43425,1.53875,1.↵
6305,1.7157499999999999,1.79225000000000001,1.85775,1.915,1.9645,2.008,2.04475,2.07525,2.098,2.↵
11875000000000004,2.12775000000000003,2.12775000000000003,2.12025,2.0995,2.0675,2.02325,1.↵
9645,1.8912499999999999,1.80125,1.69825,1.57550000000000001,1.4364999999999999,1.2855,1.↵
12075000000000001,0.94000000000000001,0.75225,0.5555,0.35325,0.14725000000000002,-0.05875000000000004,-
0.2625,-0.46074999999999994,-0.6537499999999999,-0.83700000000000001,-1.00925,-1.171,-1.323,-1.↵
46325,-1.5915,-1.70675,-1.815,-1.91425,-2.008,-2.10125,-2.1905,-2.27975,-2.3720000000000003,-2.↵
46975,-2.5735,-2.6839999999999997,-2.80225,-2.929,-3.06175,-3.20125,-3.34925,-3.5020000000000002,-
3.6575,-3.81475,-3.97275,-4.12975000000000005,-4.2807499999999999,-4.425,-4.5607500000000005,-
4.6875,-4.802,-4.8995,-4.9842499999999999,-5.05375,-5.108,-5.15125,-5.18025,-5.1995,-5.207,-5.↵
20775,-5.2010000000000005,-5.19325,-5.1819999999999995,-5.1705000000000005,-5.162,-5.159,-5.↵
16575,-5.1827499999999995,-5.2139999999999995,-5.2620000000000005,-5.3267500000000005,-5.↵
4115,-5.5175,-5.6480000000000001,-5.799,-5.9700000000000001,-6.16525,-6.374249999999999,-6.↵
60025000000000001,-6.8382499999999995,-7.084,-7.337249999999999,-7.59125,-7.846,-8.0955,-8.33675,-
8.5685,-8.78675,-8.98975,-9.17425,-9.33925,-9.48025,-9.59775,-9.6885,-9.751999999999999,-9.78475,-
9.788499999999999,-9.76025,-9.7015000000000001,-9.6100000000000001,-9.48475,-9.33225,-9.1445,-
8.9340000000000001,-8.6945,-8.432,-8.146749999999999,-7.8407500000000001,-7.515,-7.17175,-6.↵
8115,-6.4384999999999994,-6.05175,-5.6525,-5.2467500000000005,-4.8355000000000001,-4.425,-4.↵
0184999999999995,-3.61475,-3.22025,-2.83575,-2.4665,-2.114,-1.7785,-1.4565,-1.1520000000000001,-
0.859,-0.5805,-0.31124999999999997,-0.04874999999999995,0.20825000000000002,0.46375,0.↵
71725,0.9705,1.22375000000000001,1.47925,1.7335,1.98525,2.23475,2.4795000000000003,2.7175000000000002,2.↵
94425,3.157,3.35475,3.5332500000000002,3.695,3.83825,3.9635,4.0702500000000001,4.1595,4.234999999999999,4.↵
29925000000000001,4.35175,4.39375,4.429749999999999,4.4595,4.4830000000000005,4.50275,4.↵
51500000000000001,4.5265,4.5310000000000001,4.5295000000000005,4.5265,4.51725,4.508249999999999,4.↵
5104999999999995,4.519,4.522,4.5325,4.554749999999999,4.59525,4.64925,4.71575,4.79575,4.↵
8912499999999999,4.99725,5.1117500000000001,5.2337500000000001,5.3605,5.491,5.61975,5.748749999999999,5.↵
8715,5.9852500000000001,6.0975,6.20425,6.29875,6.38575,6.4582500000000005,6.517,6.558999999999999,6.↵
58200000000000001,6.587999999999999,6.5765,6.545249999999999,6.49575,6.433,6.3515000000000001,6.↵
26150000000000001,6.162249999999999,6.0577499999999995,5.951,5.841749999999999,5.732,5.↵
6245,5.51525,5.407,5.29775,5.185,5.06825,4.9445000000000001,4.8134999999999994,4.67225,4.↵
5234999999999999,4.36475,4.201499999999999,4.0329999999999995,3.8575,3.68125,3.50575,3.↵
3295,3.1555,2.9892499999999997,2.82675,2.67325,2.5284999999999997,2.39025,2.259,2.1340000000000003,2.↵
015,1.90125,1.796,1.6975,1.603,1.51525,1.4375,1.3679999999999999,1.30625,1.24975,1.20325,1.↵
16574999999999998,1.1352499999999999,1.114,1.09775,1.08325,1.0705,1.05975,1.0475,1.03375,1.↵
01775000000000002,1.001,0.98575,0.96675000000000001,0.94900000000000001,0.9315,0.91850000000000001,0.↵
90950000000000001,0.90550000000000001,0.9047499999999999,0.90950000000000001,0.91475,0.924,0.↵
9322499999999999,0.93775000000000001,0.9415,0.94075,0.9322499999999999,0.91850000000000001,0.↵
8995,0.87500000000000001,0.8484999999999999,0.81550000000000001,0.78425,0.75525,0.73025,0.↵
70575,0.68675000000000001,0.673,0.67375000000000001,0.68125,0.6689999999999999,0.6875,0.↵
72175,0.73925,0.74225000000000001,0.731,0.71025,0.695,0.69275,0.69875,0.70875,0.71875,0.72475,0.↵
72325000000000001,0.718,0.7072499999999999,0.68900000000000001,0.66525,0.63775,0.60575,0.↵
57075,0.53550000000000001,0.4935,0.45325,0.41050000000000003,0.367,0.325,0.28525,0.24650000000000002,0.↵
21125,0.18075000000000002,0.15325,0.1335,0.11525,0.10525000000000001,0.09625,0.09225,0.↵
09225,0.09075,0.09,0.08925,0.08549999999999999,0.08074999999999999,0.07475,0.06325,0.05025,0.↵
03424999999999996,0.01749999999999998,-0.00075,-0.019,-0.03975,-0.05875000000000004,-0.↵
08074999999999999,-0.10074999999999999,-0.11674999999999999,-0.1335,-0.14725000000000002,-
0.158,-0.1625,-0.161,-0.1525,-0.13725,-0.11449999999999999,-0.084,-0.0495,-0.00925,0.0305,0.07025,0.↵
11075,0.14350000000000002,0.16925,0.1885,0.19675,0.1975,0.18625,0.1685,0.14575,0.11674999999999999,0.↵
08474999999999999,0.0535,0.026,0.00075,-0.01375,-0.01975,-0.01675,-0.0015,0.02525,0.064,0.↵
10975,0.1685,0.23424999999999999,0.3075,0.3845,0.46775,0.5515,0.64,0.72700000000000001,0.↵
81775000000000001,0.9079999999999999,0.9979999999999999,1.0865,1.17275,1.25875,1.34425,1.↵

429,1.5105,1.59225,1.67075,1.7495,1.82425,1.89275,1.9585000000000001,2.0157499999999997,2.↵
0652500000000003,2.10425,2.1277500000000003,2.13775,2.13175,2.1087499999999997,2.0675,2.↵
005,1.9265,1.828,1.7105,1.57925,1.43275,1.27325,1.0987500000000001,0.91625,0.7292500000000001,0.↵
53875,0.34725,0.1565,-0.032,-0.2135000000000002,-0.3875,-0.5499999999999999,-0.705,-0.8500000000000001,-
0.98725,-1.114,-1.2315,-1.34125,-1.4449999999999998,-1.5457500000000002,-1.64575,-1.7472499999999997,-
1.8502500000000002,-1.957,-2.0652500000000003,-2.179,-2.3009999999999997,-2.42775,-2.562,-2.↵
7045,-2.85175,-3.0052499999999998,-3.164,-3.32325,-3.48575,-3.65,-3.8125,-3.97575,-4.1335,-4.2855,-
4.4265,-4.557,-4.6745,-4.779,-4.8660000000000005,-4.9392499999999995,-4.9935,-5.029999999999999,-
5.05375,-5.06275,-5.059749999999999,-5.0500000000000001,-5.031499999999999,-5.011,-4.9895000000000005,-
4.968999999999999,-4.9544999999999995,-4.94625,-4.94925,-4.966,-4.995,-5.0385,-5.09875,-5.1765,-
5.2705,-5.38475,-5.51825,-5.67025,-5.840999999999999,-6.0317500000000001,-6.2385,-6.4582500000000005,-
6.6925,-6.935,-7.18525,-7.43875,-7.6905,-7.93925,-8.181,-8.4145,-8.63425,-8.83875,-9.0270000000000001,-
9.195,-9.343,-9.4712500000000001,-9.57575,-9.6587500000000001,-9.716,-9.7487500000000001,-9.↵
7549999999999999,-9.7320000000000001,-9.6795,-9.59775,-9.48475,-9.3415,-9.16525,-8.958499999999999,-
8.72125,-8.45425,-8.161999999999999,-7.845249999999999,-7.50875,-7.15325,-6.7847500000000001,-
6.40725,-6.0272499999999996,-5.6472500000000005,-5.2705,-4.8987500000000001,-4.53425,-4.18175,-
3.8375,-3.5035,-3.1785,-2.8594999999999997,-2.54525,-2.2315,-1.9195,-1.60825,-1.29625,-0.9804999999999999,-
0.6675,-0.3555,-0.045,0.26175,0.5599999999999999,0.8522500000000001,1.1344999999999998,1.↵
4092500000000001,1.6740000000000002,1.92875,2.172,2.4085,2.633,2.8465000000000003,3.0495,3.↵
241,3.421,3.58575,3.7359999999999998,3.87125,3.9895,4.09,4.1755,4.2465,4.3015,4.3457500000000001,4.↵
3770000000000001,4.4014999999999995,4.4174999999999995,4.4265,4.4319999999999995,4.4365000000000006,4.↵
4387500000000001,4.4380000000000001,4.4387500000000001,4.4387500000000001,4.44325,4.4510000000000005,4.↵
4677500000000006,4.48925,4.522,4.567,4.62575,4.70125,4.7897500000000001,4.8942499999999995,5.↵
011,5.1407500000000001,5.2795000000000005,5.42525,5.57775,5.7275,5.87075,6.005,6.128,6.2385,6.↵
32925,6.4049999999999999,6.4659999999999999,6.5087500000000001,6.5392500000000001,6.55375,6.↵
5545000000000001,6.5482499999999995,6.53525,6.514,6.4895,6.4582500000000005,6.4232499999999995,6.↵
38425,6.3377500000000001,6.2875,6.227,6.15775,6.0775000000000001,5.98975,5.89,5.783,5.66625,5.↵
52975,5.38625,5.2475000000000005,5.10875,4.95675,4.802,4.6477499999999999,4.4914999999999999,4.↵
3290000000000001,4.1665,3.99925,3.83225,3.6637500000000003,3.4935,3.31875,3.14875,2.9855,2.↵
82675,2.6717500000000003,2.5229999999999997,2.3872500000000003,2.26125,2.14775,2.04075,1.↵
94475,1.8585,1.7760000000000002,1.70375,1.63575,1.5709999999999997,1.50825,1.4465000000000001,1.↵
3885,1.33275,1.2802499999999999,1.23225,1.188,1.149,1.1155,1.088,1.06425,1.0475,1.03375,1.↵
0255,1.0185,1.014,1.0125,1.00925,1.0070000000000001,1.0017500000000001,0.9957499999999999,0.↵
98875,0.9787499999999999,0.9674999999999999,0.9522499999999999,0.9352499999999999,0.↵
9155,0.8965000000000001,0.87425,0.8500000000000001,0.82775,0.8024999999999999,0.7805000000000001,0.↵
759,0.7385,0.718,0.7005,0.6859999999999999,0.67225,0.66075,0.653,0.64925,0.64925,0.6537499999999999,0.↵
6637500000000001,0.676,0.6859999999999999,0.7012499999999999,0.7132499999999999,0.7270000000000001,0.↵
73775,0.74475,0.7462500000000001,0.74075,0.7284999999999999,0.711,0.6890000000000001,0.66,0.↵
6295000000000001,0.59425,0.5555,0.51725,0.47825,0.4394999999999995,0.4035,0.3655,0.3295,0.↵
296,0.264,0.23275,0.2045,0.18,0.15725,0.13875,0.1242499999999999,0.11075,0.106,0.10225,0.↵
1015,0.0992499999999999,0.1,0.097,0.09225,0.07625,0.061,0.0442500000000005,0.00525,-0.↵
04425000000000005,-0.07775,-0.1007499999999999,-0.11525,-0.13125,-0.15025,-0.17775,-0.203,-
0.22200000000000003,-0.2380000000000002,-0.24175,-0.2342499999999999,-0.219,-0.19525,-0.↵
16325,-0.125,-0.08325,-0.04125,-0.003,0.032,0.0625,0.087,0.10225,0.11075,0.1115000000000002,0.↵
10525000000000001,0.0915,0.07325,0.0495,0.02124999999999998,-0.00375,-0.032,-0.058,-0.07925,-
0.0945,-0.10375,-0.10525000000000001,-0.0985,-0.0847499999999999,-0.064,-0.0342499999999996,0.↵
002250000000000003,0.04274999999999996,0.09075,0.1420000000000002,0.196,0.2562500000000003,0.↵
319,0.38225,0.4502500000000004,0.51875,0.5935,0.6705,0.753,0.84075,0.92925,1.02625,1.1245,1.↵
229,1.3345,1.44275,1.55475,1.66475,1.7760000000000002,1.8807500000000001,1.98125,2.07225,2.↵
15075,2.2155,2.26525,2.29875,2.30875,2.3032500000000002,2.2765,2.2285,2.1582500000000002,2.↵
0715,1.9645,1.84475,1.71125,1.56775,1.41375,1.25425,1.0902500000000002,0.9247500000000001,0.↵
7607499999999999,0.59825,0.4365,0.2815,0.129,-0.01749999999999998,-0.15725,-0.29075,-0.418,-
0.53875,-0.6537499999999999,-0.7645,-0.86975,-0.972,-1.0727499999999999,-1.175,-1.27875,-1.387,-
1.5022499999999999,-1.6235,-1.7555,-1.8960000000000001,-2.0425,-2.20325,-2.3682499999999997,-
2.5405,-2.7175000000000002,-2.89775,-3.0785,-3.25775,-3.4354999999999998,-3.6065,-3.7675,-3.↵
923,-4.0665,-4.19625,-4.316,-4.419,-4.5082499999999999,-4.58225,-4.64325,-4.68675,-4.718,-4.7325,-4.↵
73625,-4.73175,-4.7165,-4.699,-4.6767499999999999,-4.654,-4.6317499999999999,-4.62025,-4.61275,-4.↵
6164999999999999,-4.63325,-4.66525,-4.71575,-4.786,-4.8759999999999994,-4.9849999999999999,-5.↵

11325,-5.2604999999999995,-5.426,-5.606,-5.80375,-6.0075,-6.223999999999999,-6.44825,-6.679499999999999,-
6.914499999999999,-7.15325,-7.393750000000001,-7.634,-7.875,-8.11225,-8.346499999999999,-8.5725,-
8.792,-8.996500000000001,-9.188749999999999,-9.36275,-9.513,-9.636,-9.730500000000001,-9.79375,-
9.82125,-9.817499999999999,-9.774,-9.69625,-9.587,-9.44675,-9.277249999999999,-9.08425,-8.86525,-
8.62875,-8.37475,-8.107000000000001,-7.827000000000001,-7.53625,-7.23425,-6.926,-6.61,-6.28975,-
5.9632499999999995,-5.635,-5.304749999999999,-4.9742500000000005,-4.646249999999999,-4.319,-
3.99475,-3.67125,-3.35075,-3.02975,-2.7115,-2.3924999999999996,-2.07225,-1.754,-1.43275,-1.11225,-
0.792,-0.473,-0.15725,0.154,0.4555,0.75225,1.03675,1.312250000000001,1.57325,1.82025,2.05775,2.↵
28125,2.49325,2.6932500000000004,2.8815,3.05625,3.22275,3.3715,3.51025,3.63475,3.743,3.83675,3.↵
91,3.968,4.008500000000001,4.0322499999999994,4.042,4.040500000000001,4.02825,4.0115,3.↵
99025,3.9735,3.959,3.9572499999999997,3.96725,3.99475,4.0375000000000005,4.0969999999999995,4.↵
1732499999999995,4.264749999999999,4.37,4.48525,4.609,4.7355,4.867500000000001,4.998749999999999,5.↵
127000000000001,5.25275,5.3725000000000005,5.489249999999999,5.59775,5.70075,5.79825,5.↵
88925,5.973,6.05175,6.1225000000000005,6.186,6.2385,6.282749999999999,6.3165,6.3415,6.358999999999999,6.↵
36525,6.363,6.35225,6.3317499999999995,6.3065,6.276,6.23625,6.192,6.1385000000000005,6.07975,6.↵
0104999999999995,5.9334999999999996,5.846500000000001,5.748749999999999,5.639,5.51825,5.↵
3870000000000005,5.2475000000000005,5.0995,4.9422500000000005,4.779,4.61275,4.445749999999999,4.↵
278499999999999,4.1129999999999995,3.949,3.7872500000000002,3.6285,3.47525,3.32575,3.1815,3.↵
04175,2.9082500000000002,2.77775,2.65125,2.52675,2.40625,2.291,2.179,2.073,1.9745,1.880000000000001,1.↵
78975,1.7075,1.6320000000000001,1.56325,1.5015,1.44575,1.397,1.3565,1.31825,1.28475,1.253499999999999,1.↵
22675,1.20325,1.18025,1.15975,1.1520000000000001,1.1475000000000002,1.134499999999999,1.↵
1215,1.11325,1.1085,1.101,1.0925,1.08425,1.0705,1.057499999999999,1.039,1.02,0.9972500000000001,0.↵
9705,0.9415,0.9125,0.88575,0.8612500000000001,0.8347500000000001,0.81475,0.7965,0.7865000000000001,0.↵
779,0.77525,0.77525,0.779749999999999,0.787249999999999,0.79575,0.808,0.8210000000000001,0.↵
8347500000000001,0.8460000000000001,0.85825,0.869,0.87975,0.8880000000000001,0.8942500000000001,0.↵
8965000000000001,0.8942500000000001,0.885,0.87125,0.850749999999999,0.82325,0.78975,0.↵
7515,0.711,0.666,0.62175,0.57825,0.53475,0.4967499999999997,0.4629999999999997,0.4365,0.↵
4135,0.396,0.3845,0.37625,0.3737499999999997,0.373,0.3737499999999997,0.3745,0.3715,0.↵
367,0.35775,0.34325,0.32575,0.30125,0.27075,0.2372500000000002,0.20075,0.16025,0.11975,0.↵
0755,0.03275,-0.006,-0.04125,-0.068,-0.08925,-0.1030000000000001,-0.109,-0.106,-0.09775,-0.08,-0.↵
05875000000000004,-0.03275,-0.002250000000000003,0.02975000000000002,0.06025,0.09,0.↵
11525,0.13575,0.155,0.16325,0.167,0.1662499999999998,0.15875,0.1465,0.129,0.10975,0.08625,0.↵
06325,0.039,0.01749999999999998,-0.00375,-0.01975,-0.0342499999999996,-0.045,-0.0495,-0.↵
04874999999999995,-0.045,-0.03575000000000004,-0.023,0.0,0.02375,0.0349999999999996,0.↵
0725,0.11975,0.151,0.174,0.18775,0.2045,0.2305,0.2755,0.33725,0.4135,0.4989999999999994,0.↵
59425,0.69575,0.80725,0.9247500000000001,1.0475,1.176499999999999,1.30625,1.4375,1.5685,1.↵
6945000000000001,1.81425,1.9265,2.028,2.1187500000000004,2.1950000000000003,2.2567500000000003,2.↵
3009999999999997,2.3285,2.3369999999999997,2.327,2.2965,2.2485,2.182,2.0995,1.9997500000000001,1.↵
8875,1.76,1.62425,1.48475,1.33825,1.191,1.04375,0.89875,0.756749999999999,0.621,0.48975,0.↵
3655,0.24725,0.1342499999999998,0.02525,-0.0825,-0.183,-0.28525,-0.3889999999999996,-0.↵
49125,-0.599,-0.70875,-0.82325,-0.943,-1.072,-1.20625,-1.34725,-1.496,-1.65175,-1.818,-1.9875,-2.↵
16675,-2.3497500000000002,-2.53675,-2.72675,-2.9175,-3.106,-3.292,-3.47125,-3.64,-3.801,-3.94525,-
4.07725,-4.1884999999999994,-4.283250000000001,-4.361000000000001,-4.4167499999999995,-4.↵
4555,-4.4792499999999995,-4.48825,-4.484500000000001,-4.473,-4.45475,-4.4334999999999996,-4.↵
4152499999999995,-4.3975,-4.3885,-4.3862499999999995,-4.3885,-4.406000000000001,-4.4319999999999995,-
4.4670000000000005,-4.5135000000000005,-4.569249999999999,-4.63875,-4.71725,-4.8095,-4.91875,-
5.04075,-5.18025,-5.335999999999999,-5.51075,-5.702249999999999,-5.9105,-6.13625,-6.3735,-6.↵
62525,-6.8839999999999995,-7.14875,-7.41575,-7.67975,-7.9415,-8.193999999999999,-8.434999999999999,-
8.66325,-8.87225,-9.062249999999999,-9.22925,-9.375,-9.494,-9.5885,-9.654250000000001,-9.69,-
9.697,-9.674000000000001,-9.6245,-9.54425,-9.439,-9.30475,-9.147,-8.96825,-8.7715,-8.5595,-8.↵
331249999999999,-8.0925,-7.840750000000001,-7.581250000000001,-7.315,-7.0435,-6.76575,-6.482,-
6.1905,-5.890750000000001,-5.585500000000001,-5.274249999999999,-4.95675,-4.63475,-4.30675,-
3.9765,-3.6445,-3.3105,-2.9785,-2.64675,-2.32,-1.9975,-1.6785,-1.36325,-1.052,-0.743,-0.4365,-0.↵
1305,0.174,0.4760000000000003,0.77525,1.072,1.36325,1.645,1.91725,2.176,2.4154999999999998,2.↵
6374999999999997,2.83425,3.0082500000000003,3.15775,3.2845,3.3875,3.4705,3.5332500000000002,3.↵
579,3.61325,3.63625,3.6545,3.66825,3.67975,3.691,3.7055,3.72325,3.746,3.7735,3.80925,3.85425,3.↵
90625,3.968,4.036,4.117500000000001,4.2082500000000005,4.3075,4.4135,4.5265,4.64475,4.766,4.↵
88975,5.01175,5.13375,5.25125,5.3635,5.4695,5.570250000000001,5.664000000000001,5.7517499999999995,5.↵

8312500000000001,5.9045,5.9715,6.0325,6.0912500000000005,6.14325,6.1905,6.23625,6.276,6.3095,6.↵
3377500000000001,6.358999999999999,6.369,6.3727500000000001,6.3645,6.3445,6.313249999999999,6.↵
26825,6.210249999999999,6.14025,6.0585,5.964749999999999,5.8625,5.750249999999999,5.6305000000000005,5.↵
504499999999999,5.3735,5.2375,5.09725,4.954499999999999,4.8065,4.6585,4.506,4.3525,4.19775,4.↵
04275,3.8895,3.73375,3.582,3.43325,3.28675,3.144,3.005249999999999,2.87025,2.7375,2.6100000000000003,2.↵
48875,2.3720000000000003,2.2567500000000003,2.1492500000000003,2.0425,1.94475,1.85325,1.↵
7670000000000001,1.68675,1.61275,1.5495,1.49225,1.44275,1.4007500000000002,1.36575,1.3345,1.↵
30925,1.2862500000000001,1.265,1.24425,1.22525,1.20475,1.184,1.1635,1.1405,1.119249999999999,1.↵
0987500000000001,1.07575,1.0545,1.031499999999999,1.004749999999999,0.9795,0.9537500000000001,0.↵
9247500000000001,0.89875,0.872,0.84375,0.816249999999999,0.78975,0.76525,0.745499999999999,0.↵
731,0.722499999999999,0.71725,0.71875,0.7255,0.73775,0.75375,0.7735,0.79875,0.8225,0.8460000000000001,0.↵
865999999999999,0.8805000000000001,0.890999999999999,0.8965000000000001,0.895,0.8835000000000001,0.↵
86975,0.84525,0.81475,0.7805000000000001,0.743749999999999,0.69425,0.647,0.605,0.5715,0.↵
53475,0.50425,0.47925,0.458499999999999,0.445499999999999,0.43875,0.43325,0.4325,0.↵
43325,0.434999999999999,0.438,0.4365,0.43175,0.42125,0.406,0.3837500000000004,0.35625,0.↵
322,0.2815,0.2402500000000002,0.19225,0.1472500000000002,0.099249999999999,0.0535,0.↵
00925,-0.029,-0.06325,-0.08925,-0.11225,-0.124249999999999,-0.132,-0.1335,-0.125,-0.112999999999999,-
0.0985,-0.077,-0.055,-0.029,-0.00675,0.01375,0.03275,0.0495,0.058,0.0655,0.06725,0.064,0.057249999999999,0.↵
0495,0.03825,0.0275,0.01675,0.00525,-0.003,-0.01,-0.01825,-0.02525,-0.032,-0.0357500000000004,-
0.042,-0.048749999999999,0.05425,-0.064,-0.0717500000000001,-0.0825,-0.0915,-0.097,-0.↵
1030000000000001,-0.1030000000000001,-0.10375,-0.097,-0.084749999999999,-0.0665,-0.↵
0465,-0.01675,0.01975,0.0665,0.125,0.19375,0.27075,0.35925,0.45775,0.5660000000000001,0.6875,0.↵
8170000000000001,0.95525,1.097,1.244999999999999,1.39325,1.5395,1.6815,1.8135,1.9355,2.↵
0455,2.1385,2.214,2.27125,2.311,2.32925,2.33,2.311,2.2735,2.2225,2.155249999999999,2.076,1.↵
9875,1.88975,1.78825,1.67925,1.56775,1.4465000000000001,1.332,1.22,1.0895,0.96425,0.856,0.↵
76225,0.67675,0.59125,0.502,0.406,0.3105,0.21125,0.114499999999999,0.016,-0.08175,-0.183,-0.↵
28675,-0.39975,-0.52025,-0.653749999999999,-0.79725,-0.9530000000000001,-1.12225,-1.30375,-1.↵
49525,-1.6945000000000001,-1.899,-2.105,-2.311,-2.513,-2.7085000000000004,-2.89525,-3.07225,-3.↵
2365,-3.38675,-3.521,-3.64375,-3.750499999999999,-3.84375,-3.92225,-3.987,-4.04275,-4.0840000000000005,-
4.1137500000000005,-4.1327500000000001,-4.1457500000000005,-4.152,-4.14875,-4.141249999999999,-
4.1290000000000004,-4.1167500000000001,-4.1045,-4.093999999999999,-4.08775,-4.091,-4.0992500000000005,-
4.12225,-4.152,-4.19775,-4.2565,-4.332,-4.4227500000000001,-4.5302500000000006,-4.65625,-4.80275,-
4.9645,-5.14375,-5.3420000000000005,-5.558,-5.7885,-6.0325,-6.28975,-6.555999999999999,-6.8275000000000001,-
7.1022500000000001,-7.376,-7.643999999999999,-7.9047500000000001,-8.1557500000000001,-8.3885,-8.↵
603,-8.7967500000000001,-8.96375,-9.106499999999999,-9.22475,-9.317,-9.38275,-9.4245,-9.443,-9.↵
439,-9.414,-9.3705,-9.3085,-9.227,-9.127,-9.0087500000000001,-8.87225,-8.718,-8.5435,-8.3495,-8.13675,-
7.909499999999999,7.6645,-7.4075000000000001,-7.1387500000000001,-6.862749999999999,-6.5795,-
6.2935000000000001,-6.00425,-5.716,-5.427499999999999,-5.1385,-4.8485,-4.55775,-4.265499999999999,-
3.96725,-3.6605,-3.35075,-3.03125,-2.7045,-2.3705,-2.03025,-1.68525,-1.3345,-0.98575,-0.63775,-0.↵
295249999999999,0.04125,0.367,0.681999999999999,0.980499999999999,1.264249999999999,1.↵
52975,1.7760000000000002,2.001249999999999,2.20575,2.391,2.55575,2.7015000000000002,2.↵
826,2.93425,3.0235000000000003,3.0945,3.157,3.20575,3.24775,3.2815,3.3105,3.3385,3.3645,3.↵
39275,3.427,3.4675,3.51725,3.57275,3.63775,3.7125,3.7925,3.8795,3.97275,4.0695000000000001,4.↵
16875,4.2725,4.3747500000000001,4.476999999999999,4.583,4.686,4.7882500000000001,4.88975,4.↵
99125,5.0935,5.19475,5.2940000000000005,5.39475,5.4925,5.58925,5.684,5.77325,5.8602500000000001,5.↵
9395000000000001,6.0127500000000005,6.08225,6.14325,6.194249999999999,6.239999999999999,6.↵
2752500000000001,6.300999999999999,6.32175,6.33075,6.331749999999999,6.3210000000000001,6.↵
299499999999999,6.26825,6.22625,6.16925,6.102749999999999,6.023499999999999,5.93425,5.↵
83725,5.72975,5.6145,5.4932500000000001,5.367999999999999,5.24075,5.1110000000000001,4.976,4.↵
8385,4.69975,4.5592500000000005,4.419,4.27175,4.125249999999999,3.978,3.8285,3.6805,3.5325,3.↵
38275,3.23725,3.09225,2.9525,2.819,2.68925,2.56725,2.44975,2.3385,2.238499999999999,2.14075,2.↵
0500000000000003,1.9692500000000002,1.88825,1.81725,1.75175,1.69075,1.6365,1.5885,1.54275,1.↵
50525,1.4695,1.43575,1.4085,1.383249999999999,1.36025,1.3390000000000002,1.3145,1.291,1.↵
2665,1.238249999999999,1.21,1.176499999999999,1.14125,1.10325,1.0635000000000001,1.02475,0.↵
985,0.94675,0.9087500000000001,0.875749999999999,0.846749999999999,0.8225,0.804249999999999,0.↵
795,0.78975,0.79425,0.8057500000000001,0.8225,0.842249999999999,0.865999999999999,0.↵
892749999999999,0.916999999999999,0.9415,0.963499999999999,0.978749999999999,0.98875,0.↵
991749999999999,0.98725,0.978749999999999,0.9605,0.934499999999999,0.904749999999999,0.↵

87125,0.8354999999999999,0.8009999999999999,0.7645,0.73025,0.69875,0.673,0.65075,0.63325,0.↵
62025,0.6102500000000001,0.605,0.6035,0.60725,0.612,0.615,0.61725,0.6134999999999999,0.608,0.↵
59425,0.5745,0.547,0.515,0.4760000000000003,0.4325,0.3845,0.331,0.27775,0.2265,0.18,0.13575,0.↵
097,0.06325,0.0349999999999999,0.01525,0.0,-0.0085,-0.013,-0.01525,-0.01525,-0.01,-0.004500000000000005,0.↵
0,0.006,0.01,0.0145,0.019,0.0245,0.0365,0.04725,0.0487499999999999,0.0487499999999999,0.↵
051000000000000004,0.05425,0.055,0.055,0.05425,0.0535,0.05425,0.051000000000000004,0.0435,0.↵
0365,0.0335,0.02675,0.01825,0.0085,-0.004500000000000005,-0.0174999999999999,0.032,-0.↵
05025,-0.0694999999999999,-0.0885000000000001,-0.10975,-0.132,-0.154,-0.17625,-0.1975,-0.21525,-
0.22725,-0.2342499999999999,-0.2350000000000001,-0.2249999999999999,-0.2052500000000002,-
0.17175,-0.122,-0.061,0.016,0.1075,0.21275,0.331,0.4615,0.59825,0.7437499999999999,0.8909999999999999,1.↵
0415,1.191,1.336,1.474,1.60525,1.72725,1.8365,1.934,2.0187500000000003,2.086,2.1392499999999999,2.↵
176,2.19875,2.20325,2.1927499999999998,2.166,2.12325,2.0652500000000003,1.9975,1.91875,1.↵
8325,1.73725,1.641,1.54275,1.448,1.352,1.2605,1.17275,1.0895,1.01175,0.9384999999999999,0.↵
8704999999999999,0.8049999999999999,0.73625,0.66525,0.58975,0.509,0.4205000000000004,0.↵
32275,0.2167500000000003,0.1007499999999999,-0.0297500000000002,-0.1677499999999999,-
0.3165,-0.4745000000000003,-0.644,-0.8187500000000001,-1.0017500000000001,-1.1925,-1.38625,-
1.58225,-1.78,-1.97675,-2.1767499999999997,-2.36975,-2.5597499999999997,-2.74125,-2.9152500000000003,-
3.07775,-3.228,-3.3674999999999997,-3.49425,-3.60575,-3.7039999999999997,-3.7872500000000002,-
3.8552500000000003,-3.9114999999999998,-3.94525,-3.97725,-4.0070000000000001,-4.00075,-3.↵
98625,-3.98325,-3.98875,-4.0015,-4.01375,-4.016,-4.00625,-3.9895,-3.9735,-3.96425,-3.96425,-3.97725,-
4.0077500000000001,-4.0550000000000001,-4.1244999999999999,-4.21375,-4.3282500000000001,-4.↵
4647499999999999,-4.6235,-4.80275,-5.00575,-5.223,-5.45425,-5.69925,-5.9517500000000005,-6.↵
2087499999999999,-6.47125,-6.73225,-6.9915,-7.2440000000000001,-7.489,-7.7255,-7.9545,-8.16875,-8.↵
3739999999999999,-8.56325,-8.741,-8.89975,-9.0407500000000001,-9.16375,-9.26525,-9.346,-9.4055,-9.↵
43975,-9.44975,-9.43525,-9.39625,-9.33375,-9.2482499999999999,-9.1414999999999999,-9.0164999999999999,-
8.87525,-8.7165,-8.54575,-8.3625,-8.1695000000000001,-7.9674999999999999,-7.7545,-7.5370000000000001,-
7.312,-7.0792499999999999,-6.8412500000000005,-6.5955,-6.343,-6.083,-5.8127499999999999,-5.536,-
5.2475000000000005,-4.9529999999999999,-4.6469999999999999,-4.3282500000000001,-4.0032499999999999,-
3.66825,-3.325,-2.9732499999999997,-2.616,-2.2544999999999997,-1.88975,-1.52425,-1.1605,-0.↵
8009999999999999,-0.4502500000000004,-0.10975,0.216,0.52425,0.8162499999999999,1.08575,1.↵
3335,1.55875,1.7607500000000003,1.944,2.1035000000000004,2.2445,2.3689999999999998,2.4787500000000002,2.↵
5757499999999998,2.662,2.7405,2.8129999999999997,2.8815,2.948,3.01125,3.074,3.13725,3.19975,3.↵
263,3.32875,3.39725,3.4645,3.5385,3.614,3.6942500000000003,3.77725,3.86425,3.9565,4.052,4.↵
1505,4.251,4.354,4.45775,4.56325,4.6685,4.7730000000000001,4.8805000000000005,4.9857499999999999,5.↵
0924999999999999,5.20025,5.3062499999999999,5.4122499999999999,5.51825,5.62125,5.7227500000000004,5.↵
82275,5.91975,6.012,6.0975,6.17375,6.24225,6.29725,6.3437499999999999,6.3805,6.4064999999999999,6.↵
4232499999999999,6.4292500000000001,6.4255,6.411,6.3894999999999999,6.3560000000000001,6.↵
3155,6.2630000000000001,6.20275,6.1339999999999999,6.0554999999999999,5.9677500000000006,5.↵
8729999999999999,5.7700000000000005,5.6610000000000005,5.54575,5.42375,5.2970000000000001,5.↵
16575,5.0292499999999999,4.8919999999999999,4.75225,4.61125,4.4662500000000005,4.322,4.↵
1762500000000005,4.0329999999999999,3.88725,3.74225,3.601,3.457,3.31875,3.17925,3.045,2.↵
9145,2.787,2.6687499999999997,2.555,2.44975,2.3529999999999998,2.262,2.179,2.1065,2.0385,1.↵
9805,1.92875,1.88375,1.841,1.80425,1.77,1.7365,1.703,1.667,1.62975,1.58925,1.54425,1.497,1.4465000000000001,1.↵
3915,1.3375,1.28325,1.22825,1.17725,1.1275,1.081,1.0415,1.0070000000000001,0.9804999999999999,0.↵
9575,0.9445,0.9377500000000001,0.9369999999999999,0.9415,0.9530000000000001,0.9667500000000001,0.↵
985,1.00325,1.0215,1.039,1.05275,1.065,1.07125,1.0705,1.06275,1.04975,1.03,1.00325,0.97425,0.↵
94225,0.9087500000000001,0.8750000000000001,0.84525,0.8177500000000001,0.79425,0.77525,0.↵
76225,0.75375,0.74925,0.75,0.75075,0.75525,0.7575000000000001,0.75825,0.75525,0.74925,0.74,0.↵
72475,0.70275,0.67825,0.647,0.6102500000000001,0.5737500000000001,0.53025,0.4890000000000005,0.↵
44325,0.39675,0.35175,0.3067499999999997,0.264,0.22425,0.187,0.1525,0.1242499999999999,0.↵
09525,0.07625,0.058,0.04125,0.0305,0.02225,0.01375,0.0085,0.00525,0.006,0.004500000000000005,0.↵
004500000000000005,0.00525,0.00675,0.00675,0.00675,0.00675,0.006,0.00775,0.00525,0.↵
0085,0.0115,0.0174999999999999,0.02375,0.0297500000000002,0.0357500000000004,0.↵
044250000000000005,0.05025,0.055,0.0587500000000004,0.058,0.0487499999999999,0.0342499999999999,0.↵
0145,-0.01375,-0.0465,-0.08625,-0.12825,-0.1732500000000002,-0.2197499999999997,-0.26475,-0.↵
3075,-0.344,-0.37625,-0.39825,-0.41125,-0.4165000000000004,-0.40675,-0.38825,-0.357,-0.30975,-
0.2517500000000003,-0.1815,-0.09525,0.0,0.106,0.2279999999999998,0.3585,0.48675,0.624,0.↵
779,0.9377500000000001,1.0887499999999999,1.2375,1.3832499999999999,1.5182499999999999,1.↵

64425,1.7585,1.854,1.934,1.99575,2.037,2.0622499999999997,2.069,2.06075,2.0425,2.00875,1.9685000000000001,1.↵
91875,1.863,1.805,1.74475,1.6844999999999999,1.6272499999999999,1.56625,1.50675,1.451,1.397,1.↵
34425,1.29325,1.24125,1.1895,1.1352499999999999,1.07725,1.0162499999999999,0.9482499999999999,0.↵
8757499999999999,0.79425,0.7035,0.6042500000000001,0.4935,0.37,0.2350000000000001,0.0885000000000001,-
0.0687499999999999,-0.2365,-0.41275,-0.59825,-0.7872499999999999,-0.98125,-1.17575,-1.37025,-
1.56475,-1.75775,-1.947,-2.13175,-2.31175,-2.48575,-2.6565,-2.819,-2.9732499999999997,-3.1189999999999998,-
3.25475,-3.37825,-3.4905,-3.58725,-3.67125,-3.7359999999999998,-3.7865,-3.82,-3.8375,-3.8415,-3.↵
83075,-3.81,-3.785,-3.75375,-3.7215,-3.691,-3.66125,-3.643,-3.63475,-3.64,-3.659,-3.69575,-3.74825,-3.↵
8185,-3.90475,-4.01,-4.1320000000000001,-4.27025,-4.4265,-4.596,-4.78125,-4.9805,-5.1925,-5.41675,-
5.651,-5.8930000000000001,-6.14175,-6.3935,-6.6489999999999999,-6.903,-7.1572499999999994,-7.4035,-
7.6439999999999999,-7.8727500000000001,-8.091,-8.2925,-8.47625,-8.64475,-8.79375,-8.9310000000000001,-
9.0392500000000001,-9.1247499999999999,-9.21025,-9.2675,-9.28875,-9.2787499999999999,-9.24225,-9.↵
1902499999999999,-9.127,-9.05375,-8.9722499999999999,-8.87675,-8.7685,-8.6457500000000001,-8.509,-
8.36325,-8.2030000000000001,-8.03525,-7.85975,-7.67675,-7.4852500000000001,-7.2852500000000004,-
7.0762500000000001,-6.855,-6.6229999999999999,-6.37675,-6.1164999999999999,-5.8395,-5.5465,-5.↵
23675,-4.9102500000000004,-4.5707499999999999,-4.2205,-3.859,-3.48975,-3.1165000000000003,-2.↵
7405,-2.36675,-1.9975,-1.63575,-1.2825,-0.94075,-0.6142500000000001,-0.303,-0.006,0.27,0.53025,0.↵
77275,0.9957499999999999,1.20175,1.39075,1.5610000000000002,1.7175,1.8585,1.98525,2.10425,2.↵
2102500000000003,2.31175,2.407,2.494,2.58175,2.665,2.74575,2.8245,2.9045,2.98075,3.06025,3.↵
13725,3.2135,3.28975,3.36675,3.44625,3.524,3.605,3.68725,3.7765,3.86425,3.9605,4.058,4.1610000000000005,4.↵
267,4.37775,4.49,4.60425,4.7195,4.8355000000000001,4.94925,5.06275,5.175,5.285,5.3925,5.49775,5.↵
5962499999999995,5.6922500000000005,5.783,5.86925,5.9495,6.02275,6.0882499999999995,6.↵
1477499999999999,6.1965,6.23775,6.2689999999999999,6.289,6.30025,6.3009999999999999,6.2905000000000001,6.↵
2704999999999999,6.2385,6.1965,6.14325,6.08075,6.012,5.9334999999999996,5.84875,5.75725,5.↵
6594999999999995,5.558,5.45275,5.3405000000000005,5.2245,5.104,4.97975,4.8500000000000005,4.↵
71575,4.57925,4.4365000000000006,4.2915,4.1427499999999995,3.99025,3.83825,3.685,3.53625,3.↵
3875,3.24325,3.10375,2.9732499999999997,2.848,2.73125,2.6252500000000003,2.53,2.4445,2.3682499999999997,2.↵
2995,2.2355,2.179,2.1277500000000003,2.0774999999999997,2.031,1.986,1.9377499999999999,1.↵
88975,1.8365,1.7822500000000001,1.72575,1.6632500000000001,1.6007500000000001,1.532,1.↵
46175,1.394,1.323,1.25725,1.194,1.13825,1.0902500000000002,1.052,1.02075,0.99875,0.98725,0.↵
9804999999999999,0.98125,0.9894999999999999,0.99875,1.01175,1.027,1.04075,1.05275,1.05975,1.↵
065,1.068,1.06275,1.0545,1.0415,1.0255,1.01025,0.99025,0.9712500000000001,0.9514999999999999,0.↵
9337500000000001,0.9169999999999999,0.9032500000000001,0.8902500000000001,0.882,0.8765000000000001,0.↵
87125,0.86975,0.869,0.87125,0.872,0.8734999999999999,0.8750000000000001,0.8750000000000001,0.↵
87425,0.8704999999999999,0.8659999999999999,0.856,0.84375,0.827,0.80725,0.782,0.753,0.72025,0.↵
6842499999999999,0.6455,0.6042500000000001,0.5615,0.518,0.4752499999999995,0.434,0.396,0.↵
364,0.331,0.303,0.27925,0.2602500000000004,0.2435000000000002,0.232,0.2205,0.2105000000000002,0.↵
2,0.18775,0.17625,0.16325,0.1465,0.1327499999999998,0.1167499999999999,0.10375,0.0915,0.↵
08175,0.077,0.07625,0.0785,0.0885000000000001,0.1015,0.1175000000000001,0.1364999999999998,0.↵
1565,0.1785,0.1982499999999998,0.21525,0.2279999999999998,0.2380000000000002,0.2425,0.↵
24175,0.23575,0.2220000000000003,0.2045,0.17925,0.148,0.1129999999999999,0.07325,0.0282499999999997,-
0.01975,-0.0710000000000001,-0.1227500000000001,-0.17625,-0.2279999999999998,-0.2769999999999997,-
0.32275,-0.36175,-0.396,-0.4195,-0.43175,-0.4295,-0.4135,-0.3807500000000003,-0.3334999999999996,-
0.2715,-0.19225,-0.0985,0.01075,0.1342499999999998,0.267,0.40825,0.55325,0.70275,0.85375,1.↵
001,1.142,1.27875,1.4045,1.52125,1.6265,1.7197500000000001,1.8005,1.86775,1.915,1.95,1.97075,1.↵
9775,1.97225,1.95925,1.9377499999999999,1.908,1.87675,1.84175,1.806,1.77075,1.7395,1.71125,1.↵
683,1.657,1.6327500000000001,1.60975,1.58375,1.5565,1.52275,1.48325,1.43575,1.381,1.31525,1.↵
24125,1.1575,1.06275,0.95975,0.8467499999999999,0.7284999999999999,0.612,0.4845000000000004,0.↵
33875,0.1885,0.03499999999999996,-0.129,-0.3005,-0.48075,-0.666,-0.8574999999999999,-1.0545,-
1.2565,-1.464,-1.6715,-1.8775,-2.0805000000000002,-2.27825,-2.4665,-2.64675,-2.81225,-2.964,-3.↵
099,-3.2165,-3.3165,-3.39725,-3.463,-3.511,-3.5437499999999997,-3.5645000000000002,-3.57425,-3.↵
57125,-3.563,-3.5492500000000002,-3.5325,-3.51175,-3.49425,-3.47675,-3.46375,-3.45525,-3.456,-3.↵
46525,-3.48575,-3.5195,-3.56525,-3.62925,-3.7095000000000002,-3.8055,-3.9215,-4.0535000000000005,-
4.2044999999999995,-4.36925,-4.5532499999999999,-4.75,-4.9597500000000005,-5.18425,-5.416,-5.↵
6564999999999999,-5.90525,-6.15475,-6.4094999999999995,-6.6635,-6.916,-7.16175,-7.402,-7.633249999999999,-
7.85,-8.0560000000000001,-8.2457500000000001,-8.4197499999999999,-8.57325,-8.7052500000000001,-8.↵
81725,-8.91025,-8.9835,-9.0392500000000001,-9.07675,-9.0972499999999999,-9.0995,-9.089,-9.069,-9.↵
034,-8.99125,-8.93475,-8.8700000000000001,-8.7975,-8.712,-8.615,-8.5075,-8.387749999999999,-8.↵

25575,-8.1085000000000001,-7.94675,-7.76975,-7.5767500000000005,-7.3662500000000001,-7.141,-
6.8977499999999999,-6.6345,-6.3575000000000001,-6.0615000000000006,-5.7494999999999999,-5.↵
42375,-5.08425,-4.73475,-4.3747500000000001,-4.0085000000000001,-3.63775,-3.2685,-2.89775,-2.↵
53525,-2.179,-1.83175,-1.4985,-1.168,-0.8652500000000001,-0.5805,-0.2860000000000003,-0.0115,0.↵
22975,0.4462500000000004,0.63775,0.8162499999999999,0.99025,1.16425,1.33525,1.49775,1.↵
6495,1.78975,1.92025,2.04,2.153,2.2575000000000003,2.3560000000000003,2.4467499999999998,2.↵
5345,2.62,2.7015000000000002,2.78325,2.864,2.945,3.02575,3.10825,3.1905,3.2775,3.36675,3.45775,3.↵
55225,3.65075,3.7504999999999997,3.85425,3.96125,4.0695000000000001,4.18175,4.2944999999999999,4.↵
40825,4.5234999999999999,4.6395,4.7585000000000001,4.873,4.99125,5.1069999999999999,5.22225,5.↵
33375,5.4465,5.55275,5.6572499999999994,5.7547500000000005,5.8457500000000001,5.9295,6.↵
005,6.0714999999999995,6.12875,6.17525,6.2125,6.2399999999999999,6.2585,6.26825,6.26825,6.↵
2600000000000001,6.2440000000000001,6.2195,6.18825,6.1492499999999999,6.1019999999999999,6.↵
0447500000000005,5.9830000000000005,5.91275,5.8365,5.75325,5.66025,5.5595,5.4565,5.34375,5.↵
22225,5.0935,4.9575000000000005,4.8157499999999995,4.66925,4.519,4.3625,4.20675,4.052,3.↵
89775,3.746,3.6025,3.466,3.3369999999999997,3.2195,3.1105,3.01275,2.9242500000000002,2.8457500000000002,2.↵
7725,2.707,2.64425,2.5825,2.5229999999999997,2.4604999999999997,2.39575,2.32775,2.2560000000000002,2.↵
1774999999999998,2.0972500000000003,2.011,1.925,1.8339999999999999,1.74875,1.6625,1.58475,1.↵
5075,1.43975,1.38025,1.3259999999999998,1.2817500000000002,1.2449999999999999,1.217,1.194,1.↵
17875,1.1672500000000001,1.15825,1.1535,1.14825,1.143,1.139,1.13225,1.126,1.117,1.1055000000000001,1.↵
09325,1.07875,1.065,1.052,1.0375,1.0215,1.0055,0.985,0.96825,0.9497500000000001,0.9337500000000001,0.↵
9192499999999999,0.9087500000000001,0.89725,0.8920000000000001,0.8865,0.88725,0.8880000000000001,0.↵
8927499999999999,0.89725,0.9017499999999999,0.9087500000000001,0.91025,0.9095000000000001,0.↵
9047499999999999,0.8995,0.885,0.869,0.8484999999999999,0.82325,0.792,0.7575000000000001,0.↵
72025,0.6835,0.64625,0.6094999999999999,0.5745,0.53875,0.505,0.4745000000000003,0.4462500000000004,0.↵
4165000000000004,0.3905,0.3632499999999996,0.33425,0.3067499999999997,0.2769999999999997,0.↵
25025,0.2197499999999997,0.19,0.161,0.12975,0.10225,0.07475,0.0487499999999995,0.0275,0.↵
0085,-0.00925,-0.0212499999999998,-0.029,-0.0349999999999996,-0.0349999999999996,-0.↵
0342499999999996,-0.02675,-0.01825,-0.004500000000000005,0.0085,0.026,0.045,0.0655,0.↵
0877500000000001,0.1082499999999999,0.129,0.14575,0.15875,0.1655,0.167,0.15875,0.14275,0.↵
116,0.0825,0.039,-0.013,-0.0725,-0.13725,-0.2060000000000002,-0.27625,-0.348,-0.415,-0.47825,-0.↵
53625,-0.58375,-0.62325,-0.6515,-0.666,-0.6697500000000001,-0.6577500000000001,-0.63025,-0.58975,-
0.53025,-0.4539999999999996,-0.36475,-0.26175,-0.14425,-0.016,0.122,0.2654999999999996,0.↵
41275,0.56075,0.7065,0.8484999999999999,0.9795,1.10175,1.21525,1.316,1.40225,1.4755,1.538,1.↵
587,1.6265,1.657,1.68,1.69825,1.71275,1.722,1.73175,1.73875,1.744,1.74875,1.75325,1.75475,1.↵
7525,1.7495,1.74325,1.73275,1.71825,1.70125,1.68,1.65475,1.62575,1.59375,1.55475,1.513,1.464,1.↵
41075,1.3535,1.28475,1.2085000000000001,1.12225,1.0255,0.9155,0.7965,0.66225,0.515,0.3540000000000004,0.↵
18475,0.00375,-0.187,-0.3845,-0.5904999999999999,-0.7979999999999999,-1.0085,-1.22,-1.4297499999999999,-
1.6335,-1.82875,-2.0172499999999998,-2.19875,-2.365,-2.5185,-2.65875,-2.78325,-2.8930000000000002,-
2.98625,-3.06025,-3.1205,-3.1655,-3.19675,-3.2165,-3.2235,-3.222,-3.21425,-3.20125,-3.18375,-3.↵
1685,-3.15025,-3.1350000000000002,-3.12125,-3.1135,-3.1165000000000003,-3.1220000000000003,-
3.1350000000000002,-3.164,-3.215,-3.273,-3.3425000000000002,-3.4255,-3.5292499999999998,-3.↵
6515,-3.79475,-3.9565,-4.136,-4.33125,-4.5425,-4.766,-5.00325,-5.2497500000000001,-5.49925,-5.↵
7555000000000005,-6.0104999999999995,-6.2615000000000001,-6.50625,-6.7434999999999999,-6.↵
97325,-7.19225,-7.399,-7.5905000000000005,-7.7689999999999999,-7.9307500000000001,-8.0795000000000001,-
8.2167500000000001,-8.33675,-8.442,-8.53275,-8.61125,-8.6762500000000001,-8.72875,-8.7692500000000001,-
8.8005,-8.8195,-8.831,-8.83175,-8.82175,-8.80275,-8.77225,-8.731,-8.677,-8.60975,-8.52575,-8.426,-8.↵
3099999999999999,-8.1742499999999999,-8.0185000000000001,-7.843,-7.6454999999999999,-7.4287500000000001,-
7.1915000000000004,-6.935,-6.662,-6.3705,-6.0645,-5.74575,-5.4177500000000001,-5.0805000000000001,-
4.73625,-4.3892500000000005,-4.0375000000000005,-3.68725,-3.3385,-2.9975,-2.65875,-2.3307499999999997,-
2.00875,-1.69675,-1.3955,-1.1039999999999999,-0.82475,-0.55775,-0.3037499999999996,-0.06325,0.↵
16325,0.3785,0.57825,0.76825,0.9475000000000001,1.11225,1.268,1.413,1.548,1.67925,1.80125,1.↵
915,2.02725,2.1325,2.23225,2.3314999999999997,2.4245,2.51625,2.6054999999999997,2.69175,2.↵
77775,2.861,2.9457500000000003,3.0305,3.1157500000000002,3.205,3.2960000000000003,3.38975,3.↵
4875000000000003,3.58975,3.6965,3.80625,3.92,4.036,4.1557499999999999,4.27325,4.3900000000000001,4.↵
51825,4.6425,4.74475,4.8515000000000001,4.97525,5.108,5.23975,5.365,5.4787500000000001,5.57475,5.↵
6594999999999995,5.7342499999999999,5.803,5.86625,5.92425,5.97225,6.01725,6.0539999999999999,6.↵
0859999999999999,6.11275,6.1317499999999999,6.144,6.15,6.1484999999999999,6.1425,6.125,6.1004999999999999,6.↵
0685,6.0257499999999995,5.97225,5.9082500000000001,5.8295,5.7410000000000005,5.63975,5.526,5.↵

40225,5.2695,5.127000000000001,4.9805,4.828749999999999,4.677499999999999,4.523499999999999,4.↵
374750000000001,4.22975,4.09,3.95825,3.833749999999998,3.71775,3.61175,3.5125,3.4195,3.↵
3325,3.2485,3.16775,3.0885,3.008250000000003,2.926749999999997,2.843499999999997,2.↵
75875,2.671000000000003,2.5795,2.48425,2.3895,2.295000000000004,2.198,2.10275,2.01175,1.↵
921,1.83575,1.75475,1.6785,1.60975,1.545750000000002,1.4855,1.4335,1.38625,1.348,1.3145,1.↵
2855,1.264249999999999,1.2475,1.2315,1.2185,1.2055,1.194,1.18025,1.165749999999998,1.15275,1.↵
135249999999999,1.114,1.094750000000001,1.06975,1.046749999999998,1.02375,1.0025,0.↵
9835,0.966,0.951499999999999,0.940000000000001,0.93075,0.92775,0.9285,0.932249999999999,0.↵
937750000000001,0.94525,0.952249999999999,0.961249999999999,0.96825,0.972,0.97425,0.↵
9735,0.969000000000001,0.962,0.953000000000001,0.9415,0.92925,0.914,0.898,0.87975,0.863000000000001,0.↵
84375,0.824000000000001,0.802499999999999,0.780500000000001,0.756749999999999,0.↵
73175,0.70575,0.6775,0.647749999999999,0.614250000000001,0.5845,0.550750000000001,0.↵
521,0.4890000000000005,0.45775,0.428749999999999,0.398999999999997,0.36625,0.335,0.↵
302,0.27225,0.241,0.20975,0.18,0.147250000000002,0.116749999999999,0.08925,0.068,0.048749999999999,0.↵
034249999999999,0.026,0.02375,0.0275,0.039,0.055750000000001,0.07925,0.106,0.132749999999998,0.↵
1625,0.193,0.22425,0.249499999999997,0.27475,0.295249999999999,0.309,0.315750000000003,0.↵
312750000000003,0.30525,0.28775,0.2595,0.2205,0.174,0.116749999999999,0.05275,-0.019,-0.↵
097,-0.177000000000002,-0.260250000000004,-0.34325,-0.420500000000004,-0.49525,-0.55925,-
0.611,-0.6515,-0.67525,-0.68275,-0.67225,-0.642500000000001,-0.599,-0.537750000000001,-0.↵
459999999999999,-0.370749999999997,-0.268499999999996,-0.15875,-0.042749999999999,0.↵
07775,0.20075,0.322,0.444000000000006,0.559999999999999,0.6745,0.78425,0.888000000000001,0.↵
9865,1.07875,1.1635,1.2435,1.31675,1.383249999999999,1.44425,1.497,1.545,1.586250000000002,1.↵
62125,1.65025,1.674000000000002,1.6915,1.7075,1.7175,1.728,1.735,1.7395,1.74325,1.748,1.75175,1.↵
7555,1.75775,1.75925,1.75925,1.7555,1.748,1.735750000000001,1.719,1.6915,1.6595,1.61825,1.↵
56475,1.503,1.429,1.341999999999999,1.24275,1.13,1.004,0.861999999999999,0.708000000000001,0.↵
5395,0.36075,0.170999999999999,-0.0275,-0.23575,-0.44475,-0.66075,-0.87425,-1.08725,-1.29475,-
1.497,-1.69075,-1.87675,-2.0515,-2.214,-2.3635,-2.50025,-2.623750000000002,-2.733500000000003,-
2.828999999999997,-2.911499999999998,-2.98075,-3.0335,-3.07475,-3.1045,-3.12275,-3.13175,-3.↵
1295,-3.12125,-3.106750000000003,-3.08675,-3.0655,-3.0465,-3.02725,-3.016,-3.0105,-3.01425,-3.↵
03125,-3.05775,-3.100500000000003,-3.160749999999997,-3.23575,-3.331,-3.44475,-3.57525,-3.↵
725500000000003,-3.89175,-4.074,-4.27175,-4.481500000000005,-4.705,-4.936999999999999,-5.175,-
5.4185,-5.662500000000005,-5.902749999999999,-6.139250000000005,-6.3705,-6.593999999999999,-
6.8115,-7.016749999999999,-7.212000000000001,-7.3945,-7.56375,-7.723249999999999,-7.872750000000001,-
8.01075,-8.139000000000001,-8.25875,-8.371,-8.47475,-8.56925,-8.6555,-8.7365,-8.80725,-8.87075,-
8.924,-8.969000000000001,-9.002,-9.020999999999999,-9.027000000000001,-9.016499999999999,-
8.978250000000001,-8.929499999999999,-8.866999999999999,-8.78375,-8.68,-8.554,-8.40825,-8.↵
244250000000001,-8.058250000000001,-7.85375,-7.631,-7.39125,-7.1335,-6.862,-6.574249999999999,-
6.2745,-5.96225,-5.643499999999995,-5.317,-4.983499999999999,-4.6455,-4.30525,-3.9665,-3.63,-3.↵
296000000000003,-2.9685,-2.6475,-2.334500000000002,-2.031749999999997,-1.73875,-1.45425,-
1.1825,-0.920750000000001,-0.6715,-0.43325,-0.205250000000002,0.01225,0.219749999999997,0.↵
416500000000004,0.604250000000001,0.780500000000001,0.947500000000001,1.105500000000001,1.↵
25575,1.399249999999999,1.533499999999999,1.6595,1.78075,1.8935,2.003500000000003,2.↵
1065,2.2025,2.29875,2.39175,2.481,2.571,2.66125,2.752,2.844999999999998,2.942749999999998,3.↵
041,3.14475,3.251,3.36375,3.4835,3.60575,3.73225,3.86275,3.994,4.128249999999995,4.2625,4.↵
396,4.528000000000005,4.65475,4.7775,4.895750000000005,5.00875,5.1185,5.2215,5.319999999999999,5.↵
4115,5.50075,5.584,5.6655,5.7435,5.82275,5.899,5.970000000000001,6.038749999999999,6.10425,6.↵
16525,6.2195,6.265999999999999,6.302000000000005,6.32625,6.337750000000001,6.337,6.321000000000001,6.↵
2865,6.2415,6.176,6.0975,6.010499999999995,5.908250000000001,5.797499999999999,5.68325,5.↵
56025,5.43525,5.30775,5.18025,5.05225,4.927000000000005,4.805000000000001,4.69425,4.57925,4.↵
467000000000005,4.38,4.282500000000001,4.177000000000005,4.0665,3.9505,3.83675,3.73375,3.↵
6415,3.556,3.46975,3.38275,3.292,3.19675,3.096749999999997,2.9945,2.889999999999997,2.784,2.↵
67575,2.5695,2.465,2.365,2.2675,2.176,2.08975,2.011,1.94025,1.875249999999999,1.818,1.76925,1.↵
7235,1.68525,1.651,1.6175,1.585500000000001,1.55575,1.525,1.49225,1.45875,1.42375,1.3885,1.↵
3535,1.32,1.284,1.252,1.2215,1.19475,1.171,1.14975,1.13375,1.120750000000001,1.11,1.10175,1.↵
09625,1.091,1.08725,1.0825,1.08175,1.07875,1.078,1.07575,1.07725,1.07575,1.0765,1.07125,1.068,1.↵
063500000000001,1.057499999999999,1.0475,1.036,1.02375,1.00775,0.992499999999999,0.↵
97425,0.9545,0.935249999999999,0.91325,0.894250000000001,0.87425,0.8545,0.83225,0.814,0.↵
792,0.769,0.742250000000001,0.714,0.6835,0.64925,0.614250000000001,0.573750000000001,0.↵

534,0.4904999999999999,0.44775,0.40275,0.35775,0.31425,0.27075,0.22975,0.19075,0.15175,0.↵
11825,0.08925,0.061,0.042,0.026,0.01375,0.01075,0.0115,0.0205,0.0365,0.0572499999999999,0.↵
084,0.1175000000000000,0.151,0.1915,0.2312499999999999,0.27075,0.306,0.33575,0.357,0.37,0.↵
3707499999999999,0.3585,0.3334999999999999,0.2952499999999999,0.24175,0.17775,0.10225,0.↵
01975,-0.07025,-0.1655,-0.25775,-0.35025,-0.4365,-0.51425,-0.5805,-0.63475,-0.6775,-0.71025,-0.72775,-
0.73625,-0.72775,-0.7072499999999999,-0.673,-0.6295000000000001,-0.573,-0.508,-0.43175,-0.348,-
0.257,-0.15875,-0.058,0.04725,0.154,0.26175,0.36925,0.4775,0.582,0.6842499999999999,0.78275,0.↵
8750000000000000,0.962,1.043,1.11625,1.184,1.24425,1.30075,1.34975,1.39325,1.4312500000000001,1.↵
46625,1.497,1.52275,1.5465,1.5685,1.59225,1.6135,1.6365,1.66175,1.68675,1.7149999999999999,1.↵
744,1.77075,1.7975,1.82275,1.84475,1.8615,1.87075,1.87225,1.867,1.8455,1.81575,1.7722499999999999,1.↵
71275,1.6395,1.55325,1.45025,1.336,1.20625,1.0635000000000001,0.9095000000000001,0.7437499999999999,0.↵
5700000000000000,0.38525,0.19225,-0.004500000000000005,-0.20600000000000002,-0.41125,-0.↵
6142500000000000,-0.8194999999999999,-1.0215,-1.21925,-1.41225,-1.59675,-1.77075,-1.93625,-2.↵
089,-2.23,-2.3575,-2.46975,-2.5687499999999999,-2.6505,-2.71825,-2.77175,-2.8115,-2.83725,-2.85025,-
2.85275,-2.8465000000000003,-2.8320000000000003,-2.81,-2.787,-2.76325,-2.7397500000000004,-2.↵
719,-2.703,-2.69775,-2.7022500000000003,-2.7199999999999998,-2.74975,-2.7977499999999997,-2.↵
8632500000000003,-2.9457500000000003,-3.0465,-3.1655,-3.302,-3.457,-3.627,-3.81,-4.00475,-4.2115,-
4.425,-4.6425,-4.8629999999999995,-5.085,-5.3039999999999999,-5.5215,-5.7349999999999999,-5.944,-
6.1469999999999999,-6.3445,-6.5345,-6.7185,-6.8977499999999999,-7.06875,-7.23725,-7.4027499999999999,-
7.5592500000000005,-7.7109999999999999,-7.85675,-7.99875,-8.13225,-8.26025,-8.381,-8.49375,-8.↵
5975,-8.6915000000000001,-8.7745,-8.84475,-8.9020000000000001,-8.94625,-8.97675,-8.98825,-8.↵
98525,-8.9645000000000001,-8.9232500000000001,-8.863,-8.78375,-8.68,-8.561,-8.4197499999999999,-8.↵
2580000000000001,-8.07725,-7.8772499999999999,-7.6585,-7.425,-7.1762500000000005,-6.91075,-6.↵
6322500000000001,-6.3437499999999999,-6.0455000000000005,-5.7395000000000005,-5.42675,-5.↵
10875,-4.7897500000000001,-4.4677500000000006,-4.1457500000000005,-3.82625,-3.51025,-3.20125,-
2.89925,-2.6025,-2.314,-2.031,-1.75925,-1.493,-1.2382499999999999,-0.9939999999999999,-0.7567499999999999,-
0.528,-0.3112499999999999,-0.0992499999999999,0.1,0.29225,0.4760000000000003,0.6477499999999999,0.↵
8002499999999999,0.9475000000000001,1.094,1.22975,1.35275,1.46625,1.57475,1.67475,1.7722499999999999,1.↵
8655000000000002,1.95475,2.04625,2.13775,2.23075,2.32925,2.4345,2.546,2.6635,2.7847500000000003,2.↵
9105,3.041,3.1745,3.308,3.44,3.57125,3.69725,3.8185,3.936,4.052,4.1602500000000004,4.2695,4.↵
37325,4.4784999999999995,4.58375,4.68675,4.79275,4.901,5.0125,5.12325,5.236,5.34825,5.46025,5.↵
5702500000000001,5.6747499999999995,5.77325,5.86325,5.9455,6.0157500000000001,6.07375,6.↵
1195,6.1530000000000005,6.17,6.17525,6.1644999999999999,6.1392500000000005,6.1019999999999999,6.↵
0539999999999999,5.9929999999999999,5.9235000000000001,5.844,5.7540000000000004,5.66025,5.↵
5595,5.45725,5.3512499999999999,5.2444999999999995,5.136,5.0299999999999999,4.9255,4.8255,4.↵
72575,4.63025,4.53575,4.444,4.354,4.26175,4.16725,4.07175,3.978,3.87875,3.77575,3.6734999999999998,3.↵
5645000000000002,3.45375,3.341,3.22725,3.1135,3.0044999999999997,2.8945,2.79,2.68925,2.5909999999999997,2.↵
50025,2.411,2.33,2.253,2.17825,2.1065,2.04,1.976,1.918,1.8615,1.8082500000000001,1.7607500000000003,1.↵
712,1.667,1.625,1.58475,1.54425,1.50525,1.46625,1.429,1.387,1.34425,1.3145,1.27475,1.22525,1.↵
1895,1.172,1.165,1.1657499999999998,1.1672500000000001,1.1635,1.15425,1.14675,1.14125,1.↵
13675,1.1352499999999999,1.1352499999999999,1.1360000000000001,1.1375,1.1360000000000001,1.↵
1360000000000001,1.13225,1.1275,1.12375,1.11625,1.1085,1.10025,1.09175,1.08425,1.078,1.07125,1.↵
0635000000000001,1.05825,1.052,1.04525,1.03675,1.0285,1.0177500000000002,1.0055,0.98875,0.↵
9735,0.9522499999999999,0.9315,0.9079999999999999,0.8805000000000001,0.85375,0.82325,0.↵
7889999999999999,0.756,0.7195,0.6775,0.63475,0.5890000000000001,0.5425,0.492,0.4394999999999995,0.↵
3889999999999999,0.338,0.2885,0.24175,0.2022499999999999,0.1677499999999998,0.1420000000000002,0.↵
125,0.1190000000000001,0.1242499999999999,0.13575,0.161,0.19375,0.2334999999999999,0.↵
2799999999999999,0.328,0.37925,0.4287499999999999,0.47225,0.51125,0.54475,0.56525,0.5752499999999999,0.↵
5760000000000001,0.5645,0.53875,0.502,0.45325,0.396,0.32725,0.25325,0.1732500000000002,0.↵
08925,0.004500000000000005,-0.08325,-0.167,-0.24725,-0.3212500000000004,-0.3905,-0.4502500000000004,-
0.50125,-0.54175,-0.5700000000000001,-0.5882499999999999,-0.5935,-0.5875,-0.5715,-0.54325,-0.505,-
0.45775,-0.4004999999999997,-0.3334999999999999,-0.26175,-0.1815,-0.09625,-0.004500000000000005,0.↵
09225,0.19075,0.29,0.3915,0.4904999999999999,0.5890000000000001,0.6835,0.77275,0.8574999999999999,0.↵
9352499999999999,1.00625,1.072,1.13,1.1795,1.2245000000000001,1.265,1.30075,1.3390000000000002,1.↵
37175,1.406,1.44125,1.477,1.5174999999999998,1.5565,1.59825,1.6424999999999998,1.6875,1.↵
73725,1.78525,1.8347499999999999,1.88225,1.9295,1.97,2.00275,2.0255,2.04325,2.0500000000000003,2.↵
0455,2.0295,1.999,1.95625,1.899,1.83025,1.7472499999999997,1.6532499999999999,1.54025,1.↵
41675,1.2795,1.12925,0.96825,0.79575,0.612,0.422,0.2227499999999998,0.019,-0.18375,-0.39225,-0.↵

602,-0.8064999999999999,-1.01175,-1.2077499999999999,-1.397,-1.5785,-1.7495,-1.90975,-2.0545,-2.↵
18425,-2.3002499999999997,-2.4017500000000003,-2.48575,-2.55275,-2.60325,-2.6382499999999998,-
2.66025,-2.6687499999999997,-2.66575,-2.6527499999999997,-2.63225,-2.60775,-2.5825,-2.5604999999999998,-
2.53975,-2.5245,-2.5185,-2.5245,-2.5412500000000002,-2.5735,-2.6214999999999997,-2.6832499999999997,-
2.76425,-2.8587499999999997,-2.96925,-3.0945,-3.23175,-3.38375,-3.54625,-3.7155,-3.8962499999999998,-
4.0817499999999995,-4.2725,-4.4692500000000001,-4.66775,-4.8660000000000005,-5.0652500000000001,-
5.2627500000000005,-5.4594999999999999,-5.6557499999999999,-5.85025,-6.0387499999999999,-6.↵
22625,-6.4102500000000004,-6.59175,-6.7710000000000001,-6.94725,-7.1205000000000001,-7.2929999999999999,-
7.4622499999999999,-7.6270000000000001,-7.788,-7.94525,-8.0985,-8.2405,-8.37625,-8.5045,-8.62275,-
8.7295,-8.8235,-8.90425,-8.973,-9.0225,-9.0575000000000001,-9.07375,-9.06975,-9.05075,-9.0102500000000001,-
8.9515,-8.8715000000000001,-8.773,-8.65175,-8.51375,-8.35425,-8.175,-7.97875,-7.76525,-7.5355000000000001,-
7.289,-7.0319999999999999,-6.76125,-6.4797499999999999,-6.192,-5.89825,-5.59925,-5.2970000000000001,-
4.995,-4.6905,-4.3885,-4.087,-3.788,-3.489,-3.1915,-2.89775,-2.6085000000000003,-2.3225,-2.04175,-1.↵
76625,-1.4985,-1.2405,-0.9924999999999999,-0.7567499999999999,-0.5325,-0.3235,-0.126,0.05875000000000004,0,↵
22799999999999998,0.38525,0.534,0.6705,0.8002499999999999,0.924,1.04225,1.159,1.274,1.38925,1.↵
506,1.625,1.7462499999999999,1.87,1.9949999999999999,2.12175,2.25,2.3772499999999996,2.↵
50625,2.63225,2.75875,2.87925,3.0,3.1182499999999997,3.231,3.3455,3.456,3.56825,3.6782500000000002,3.↵
7895,3.9025,4.0169999999999995,4.1320000000000001,4.2535000000000001,4.3762500000000001,4.↵
50125,4.6309999999999999,4.7615,4.8912499999999999,5.02325,5.1545,5.28325,5.40775,5.53125,5.↵
65799999999999995,5.7685,5.86475,5.951,6.0264999999999995,6.0874999999999995,6.1339999999999995,6.↵
16449999999999999,6.1789999999999999,6.1805,6.1675,6.14175,6.1019999999999999,6.0547499999999999,6.↵
0012500000000001,5.94175,5.8762500000000001,5.806,5.7349999999999999,5.6625000000000005,5.↵
58925,5.51525,5.43975,5.362,5.28325,5.20025,5.1185,5.0292499999999999,4.94075,4.84775,4.75475,4.↵
657,4.557,4.4540000000000001,4.34875,4.24275,4.1335,4.02525,3.9155,3.80475,3.6942500000000003,3.↵
582,3.4722500000000003,3.36225,3.2555,3.148,3.045,2.945,2.848,2.7535,2.66425,2.5780000000000003,2.↵
4955000000000003,2.41925,2.343,2.27425,2.205,2.1399999999999997,2.07525,2.015,1.9539999999999997,1.↵
89275,1.8339999999999999,1.7770000000000001,1.7205000000000001,1.664,1.6112499999999998,1.↵
55949999999999999,1.5115,1.46875,1.42825,1.39075,1.3587500000000001,1.3267499999999999,1.↵
29925,1.27725,1.2565,1.2382499999999999,1.2229999999999999,1.2085000000000001,1.1962499999999998,1.↵
18325,1.172,1.162,1.1520000000000001,1.14375,1.1352499999999999,1.1285,1.1215,1.11625,1.↵
11,1.10625,1.0987500000000001,1.09175,1.085,1.075,1.06575,1.059,1.049,1.03675,1.0255,1.0155,1.↵
00775,1.001,0.9932500000000001,0.98875,0.9827499999999999,0.97725,0.9712500000000001,0.↵
9627500000000001,0.9530000000000001,0.9377500000000001,0.9225000000000001,0.9047499999999999,0.↵
869,0.8347500000000001,0.8095,0.7667499999999999,0.705,0.63175,0.5455,0.4615,0.3845,0.31975,0.↵
26175,0.21125,0.16325,0.1235,0.09,0.06325,0.04725,0.04049999999999999,0.04125,0.0557500000000001,0.↵
07625,0.1052500000000001,0.14275,0.18475,0.2305,0.27699999999999997,0.325,0.3707499999999997,0.↵
40974999999999995,0.44325,0.4684999999999997,0.483,0.48900000000000005,0.48525,0.47000000000000003,0.↵
441,0.402,0.35400000000000004,0.29750000000000004,0.23124999999999998,0.16025,0.084,0.00375,-
0.07775,-0.15875,-0.2395,-0.3175,-0.3915,-0.46225,-0.525,-0.582,-0.631,-0.6705,-0.7020000000000001,-
0.72175,-0.73325,-0.73325,-0.72175,-0.695,-0.65925,-0.6142500000000001,-0.55775,-0.4927499999999997,-
0.41875,-0.33725,-0.25325,-0.1625,-0.07025,0.02225,0.11449999999999999,0.20224999999999999,0.↵
29,0.37225,0.451,0.52575,0.5935,0.6585,0.71725,0.7735,0.8255,0.8765000000000001,0.9225000000000001,0.↵
96975,1.0177500000000002,1.0635000000000001,1.11475,1.16425,1.21625,1.2725,1.33125,1.3915,1.↵
45275,1.51975,1.587,1.65475,1.72275,1.786,1.84775,1.908,1.9615,2.008,2.0455,2.07525,2.09275,2.↵
10125,2.0957500000000002,2.0782499999999997,2.0484999999999998,2.00575,1.95,1.879,1.796,1.↵
6975,1.58775,1.46325,1.3259999999999998,1.17425,1.01325,0.8370000000000001,0.6515,0.45775,0.↵
25775,0.05025,-0.161,-0.3715,-0.58125,-0.7865000000000001,-0.9865,-1.17875,-1.358,-1.52425,-1.↵
677,-1.815,-1.9355,-2.03925,-2.1255,-2.1927499999999998,-2.243,-2.27975,-2.3025,-2.3125,-2.314,-
2.308,-2.29575,-2.27825,-2.259,-2.2415000000000003,-2.2270000000000003,-2.21875,-2.2155,-2.↵
22324999999999997,-2.2384999999999997,-2.2675,-2.30725,-2.36275,-2.4299999999999997,-2.5100000000000002,-
2.60075,-2.7055,-2.819,-2.9457500000000003,-3.083,-3.22875,-3.38125,-3.5415,-3.70625,-3.87575,-4.↵
04824999999999995,-4.22525,-4.4052500000000006,-4.5852499999999999,-4.7675,-4.9529999999999999,-
5.13675,-5.3199999999999999,-5.50775,-5.6937500000000005,-5.88075,-6.06775,-6.253,-6.43925,-6.↵
62299999999999999,-6.8062499999999995,-6.987,-7.1647500000000001,-7.3402499999999999,-7.5120000000000005,-
7.67825,-7.8422500000000001,-7.99725,-8.1482499999999999,-8.28925,-8.4197499999999999,-8.538,-8.↵
6457500000000001,-8.7365,-8.8120000000000001,-8.8715000000000001,-8.91025,-8.9325000000000001,-
8.9310000000000001,-8.90875,-8.8677500000000001,-8.805,-8.7205000000000001,-8.61975,-8.4974999999999999,-
8.3602499999999999,-8.207,-8.03675,-7.8545000000000001,-7.6560000000000001,-7.44775,-7.228,-6.↵

99775,-6.7542500000000001,-6.5024999999999995,-6.239999999999999,-5.9677500000000006,-5.↵
6884999999999999,-5.4,-5.10475,-4.8042500000000001,-4.4997500000000001,-4.1922500000000005,-3.↵
8825,-3.57675,-3.27375,-2.9770000000000003,-2.6870000000000003,-2.4085,-2.1392499999999997,-
1.883,-1.63725,-1.403,-1.18475,-0.97575,-0.776,-0.58975,-0.4097499999999995,-0.2380000000000002,-
0.074,0.0854999999999999,0.23875,0.38825,0.5355000000000001,0.679,0.820249999999999,0.↵
95825,1.09625,1.23225,1.36575,1.497,1.627249999999999,1.7555,1.88375,2.00875,2.12925,2.25,2.↵
3689999999999998,2.48175,2.5932500000000003,2.70075,2.8075,2.9114999999999998,3.01425,3.↵
1182499999999997,3.22275,3.331,3.4415,3.56,3.6835,3.81475,3.949,4.091,4.23725,4.3862499999999995,4.↵
5394999999999999,4.692749999999999,4.84325,4.9905,5.1315,5.2620000000000005,5.3870000000000005,5.↵
49925,5.59925,5.6884999999999994,5.763999999999999,5.82425,5.8777500000000001,5.9165,5.↵
9425,5.95775,5.9647499999999996,5.9615,5.9517500000000005,5.9334999999999996,5.9082500000000001,5.↵
87925,5.8450000000000001,5.806,5.761,5.713,5.6617500000000005,5.606,5.548,5.48475,5.41925,5.↵
3512499999999999,5.2787500000000005,5.2025000000000001,5.124,5.03925,4.952249999999999,4.↵
8607499999999995,4.7645,4.6645,4.5615000000000006,4.45625,4.3495,4.239,4.1290000000000004,4.↵
0184999999999995,3.90925,3.8025,3.69875,3.58875,3.4805,3.38125,3.286,3.19225,3.0997500000000002,3.↵
0075000000000003,2.919,2.8320000000000003,2.74425,2.658,2.57425,2.49325,2.40925,2.32925,2.↵
2537499999999997,2.1805000000000003,2.1125000000000003,2.044,1.9805,1.9217499999999998,1.↵
863,1.80675,1.75475,1.702,1.6525,1.60675,1.5610000000000002,1.51675,1.474,1.43575,1.4007500000000002,1.↵
36875,1.3405,1.31675,1.29775,1.2825,1.2710000000000001,1.26125,1.25425,1.24825,1.242,1.23675,1.↵
22975,1.21925,1.207,1.194,1.1795,1.1635,1.14825,1.133,1.1185,1.1055000000000001,1.09625,1.↵
0895,1.085,1.0865,1.0887499999999999,1.0955,1.1055000000000001,1.11625,1.126,1.1375,1.146,1.↵
1520000000000001,1.1535,1.14975,1.13975,1.12375,1.0995,1.06725,1.0285,0.9780000000000001,0.↵
9232499999999999,0.8619999999999999,0.795,0.724,0.64925,0.5737500000000001,0.50275,0.43325,0.↵
3685,0.312,0.2625,0.2227499999999998,0.19075,0.16925,0.16025,0.161,0.1709999999999999,0.↵
18925,0.2145000000000002,0.24725,0.28225,0.3205,0.36075,0.4004999999999997,0.4394999999999995,0.↵
4752499999999995,0.50425,0.53025,0.54925,0.56075,0.56775,0.5645,0.5499999999999999,0.53025,0.↵
49975,0.4599999999999996,0.41275,0.35475,0.2915,0.21825,0.1342499999999998,0.05025,-0.↵
0275,-0.12125,-0.2249999999999998,-0.31425,-0.3875,-0.4462500000000004,-0.496,-0.54475,-0.↵
5875,-0.62625,-0.65625,-0.679,-0.6890000000000001,-0.6875,-0.67225,-0.647,-0.612,-0.56775,-0.51275,-
0.4547500000000004,-0.38825,-0.31975,-0.2465000000000002,-0.1732500000000002,-0.097,-0.↵
02375,0.05025,0.12125,0.1915,0.2562500000000003,0.3205,0.3785,0.4349999999999994,0.4875,0.↵
5369999999999999,0.58675,0.634,0.68275,0.73025,0.77825,0.83,0.88425,0.9415,1.0047499999999998,1.↵
06975,1.1375,1.2085000000000001,1.281,1.35575,1.4335,1.509,1.58375,1.66025,1.7302499999999998,1.↵
79975,1.86375,1.92325,1.97975,2.028,2.06825,2.10025,2.12175,2.13325,2.1325,2.11725,2.08975,2.↵
0484999999999998,1.98975,1.91725,1.8265,1.719,1.5975,1.458,1.30225,1.1344999999999998,0.↵
9514999999999999,0.7607499999999999,0.557,0.34875,0.1364999999999998,-0.07625,-0.29075,-0.↵
49975,-0.7035,-0.89875,-1.081,-1.2534999999999998,-1.4115,-1.5525,-1.67775,-1.78825,-1.8807500000000001,-
1.9585000000000001,-2.018,-2.0645000000000002,-2.0972500000000003,-2.11725,-2.1302499999999998,-
2.1340000000000003,-2.1325,-2.1285,-2.12325,-2.11725,-2.11575,-2.11575,-2.12175,-2.1340000000000003,-
2.15375,-2.18125,-2.21875,-2.26375,-2.32075,-2.3872500000000003,-2.4612499999999997,-2.5475000000000003,-
2.642,-2.748,-2.861,-2.983,-3.1165000000000003,-3.25325,-3.39725,-3.55075,-3.70875,-3.868,-4.↵
03525,-4.2029999999999999,-4.3725,-4.5440000000000005,-4.71725,-4.8912499999999999,-5.06675,-
5.2437499999999995,-5.42525,-5.60675,-5.7915,-5.98075,-6.16925,-6.3622499999999995,-6.55675,-
6.7512500000000001,-6.9465,-7.140249999999999,-7.3342500000000001,-7.523249999999999,-7.↵
7079999999999999,-7.8850000000000001,-8.05125,-8.207,-8.35125,-8.483,-8.5975,-8.696,-8.77675,-8.↵
8410000000000001,-8.88525,-8.9165,-8.927999999999999,-8.92575,-8.9080000000000001,-8.87375,-8.↵
825,-8.76,-8.6785,-8.58,-8.465499999999999,-8.336,-8.18875,-8.0215,-7.8415,-7.6415,-7.4257500000000001,-
7.1952500000000001,-6.9487499999999995,-6.6902500000000001,-6.421,-6.14025,-5.851,-5.56025,-5.↵
2627500000000005,-4.96525,-4.6662500000000001,-4.36925,-4.0765,-3.7872500000000002,-3.5027500000000003,-
3.2265,-2.95725,-2.6992499999999997,-2.4490000000000003,-2.2087499999999998,-1.97975,-1.75625,-
1.54425,-1.33825,-1.1405,-0.9497500000000001,-0.76525,-0.583,-0.40425,-0.22975,-0.0557500000000001,0.↵
11375,0.28225,0.44925,0.6094999999999999,0.769,0.92175,1.06975,1.21,1.34575,1.4755,1.59825,1.↵
71425,1.825,1.93175,2.03325,2.13625,2.23775,2.3415,2.4475000000000002,2.5565,2.6702500000000002,2.↵
7885,2.9145,3.04575,3.18375,3.32325,3.47125,3.6225,3.77725,3.9345,4.09,4.247999999999999,4.↵
4045000000000005,4.5592500000000005,4.7105,4.856,4.99425,5.1247500000000001,5.2467500000000005,5.↵
359,5.462,5.55275,5.6320000000000001,5.7015,5.75875,5.80525,5.8409999999999999,5.8685,5.888249999999999,5.↵
89975,5.9060000000000001,5.909,5.9060000000000001,5.899,5.8874999999999999,5.8715,5.854,5.↵
82875,5.79825,5.76325,5.7189999999999999,5.6695,5.61375,5.552,5.48325,5.40775,5.3267500000000005,5.↵

```
2422499999999999,5.1552500000000006,5.063750000000001,4.97125,4.877499999999995,4.782,4.↵
6875,4.589,4.49,4.390750000000001,4.2885,4.189249999999995,4.08925,3.98475,3.88325,3.7795,3.↵
6765,3.57675,3.476,3.37675,3.280000000000002,3.18375,3.093,3.006,2.91675,2.8335,2.751249999999998,2.↵
6710000000000003,2.591749999999998,2.51225,2.432250000000002,2.35525,2.27725,2.2025,2.↵
131,2.05925,1.99125,1.9265,1.864499999999998,1.809,1.7555,1.7075,1.66475,1.625,1.58925,1.↵
55875,1.528249999999999,1.50675,1.484,1.462500000000001,1.44575,1.42675,1.4075,1.387,1.↵
36425,1.3405,1.3145,1.29175,1.27475,1.25125,1.2215,1.19775,1.17725,1.15825,1.145249999999999,1.↵
133,1.12925,1.1315,1.1375,1.14825,1.162750000000002,1.184,1.207,1.22525,1.24425,1.262,1.2765,1.↵
28325,1.2855,1.27725,1.25875,1.2315,1.196249999999998,1.14975,1.097,1.039,0.972,0.901,0.827,0.↵
753,0.67825,0.6065,0.537750000000001,0.475249999999995,0.41725,0.36475,0.321250000000004,0.↵
28675,0.256250000000003,0.235000000000001,0.22425,0.219749999999997,0.222749999999998,0.↵
2350000000000001,0.25025,0.27325,0.303,0.33575,0.370749999999997,0.40825,0.446250000000004,0.↵
4822500000000007,0.5135,0.5425,0.56775,0.58525,0.595000000000001,0.5975,0.589000000000001,0.↵
57225,0.54325,0.50575,0.457,0.401249999999999,0.3365,0.26475,0.18625,0.1045,0.02225,-0.064,-
0.1487500000000002,-0.234249999999999,-0.315750000000003,-0.3915,-0.460749999999999,-
0.521,-0.573,-0.6165,-0.647,-0.668999999999999,-0.68125,-0.67975,-0.6715,-0.65225,-0.62175,-0.5875,-
0.5455,-0.498249999999997,-0.44775,-0.393,-0.33575,-0.2785,-0.22125,-0.1625,-0.10375,-0.04575,0.↵
01,0.06325,0.116,0.167,0.21825,0.27,0.319,0.3685,0.416500000000004,0.466999999999997,0.↵
521749999999999,}
```

4.3.1 Variable Documentation

4.3.1.1 test_buffer

```
float test_buffer[] = {5.656499999999999,5.667,5.673999999999995,5.672499999999999,5.6655,5.↵
6525,5.638,5.625999999999994,5.616750000000001,5.612249999999995,5.61375,5.62275,5.641249999999999,5.↵
671,5.70675,5.7465,5.78375,5.82125,5.854,5.88375,5.90975,5.93175,5.952500000000001,5.9715,5.↵
99125,6.008249999999999,6.026499999999995,6.042499999999995,6.057749999999995,6.071499999999995,6.↵
0815,6.086749999999999,6.0815,6.063000000000001,6.028750000000005,5.978499999999995,5.↵
906000000000001,5.81425,5.7015,5.5665,5.407,5.22625,5.02475,4.80275,4.56775,4.31675,4.06025,3.↵
79725,3.5325,3.2715,3.01525,2.770250000000003,2.5345,2.31175,2.10125,1.9035,1.725,1.56475,1.↵
419,1.291,1.17575,1.072,0.982000000000001,0.903250000000001,0.837000000000001,0.78425,0.↵
73775,0.6965,0.66075,0.629500000000001,0.6035,0.583,0.56375,0.54625,0.53025,0.510500000000001,0.↵
490499999999999,0.470000000000003,0.450250000000004,0.434999999999999,0.42725,0.↵
4265,0.43575,0.446250000000004,0.459250000000005,0.4715,0.484500000000004,0.503500000000001,0.↵
52425,0.54775,0.5745,0.59825,0.6195,0.6385,0.65625,0.673750000000001,0.6935,0.7125,0.734,0.↵
756,0.77675,0.79725,0.810999999999999,0.820249999999999,0.8255,0.829249999999999,0.830750000000001,0.↵
827,0.817000000000001,0.800249999999999,0.776,0.742250000000001,0.70275,0.66075,0.615,0.↵
566000000000001,0.51725,0.46775,0.42275,0.380750000000003,0.34475,0.3135,0.28525,0.267,0.↵
25325,0.24725,0.249499999999997,0.261,0.273999999999997,0.28675,0.296,0.30525,0.31425,0.↵
32275,0.3295,0.33025,0.324250000000004,0.312,0.297500000000004,0.28075,0.26175,0.2395,0.↵
212,0.18625,0.148750000000002,0.112999999999999,0.085499999999999,0.051000000000004,0.↵
00775,-0.03125,-0.06025,-0.084,-0.105250000000001,-0.122,-0.135,-0.14275,-0.14425,-0.↵
142000000000002,-0.13725,-0.135,-0.1335,-0.135,-0.142000000000002,-0.15325,-0.164,-0.↵
17625,-0.18375,-0.187,-0.18375,-0.1785,-0.167749999999998,-0.15325,-0.1335,-0.106,-0.↵
071750000000001,-0.032,0.00925,0.048,0.085499999999999,0.11825,0.148,0.177000000000002,0.↵
201499999999998,0.22425,0.248,0.27075,0.294500000000004,0.321250000000004,0.34625,0.↵
37,0.395250000000005,0.41275,0.43325,0.447,0.466999999999997,0.48075,0.4935,0.50275,0.↵
5135,0.5235,0.53875,0.5615,0.5935,0.63775,0.69275,0.756749999999999,0.82775,0.903999999999999,0.↵
982000000000001,1.06275,1.13825,1.207749999999999,1.271000000000001,1.323,1.36325,1.↵
39475,1.41825,1.4375,1.448,1.4535,1.4565,1.4565,1.460249999999998,1.46475,1.47025,1.47925,1.↵
49,1.503,1.5145,1.52425,1.52975,1.52675,1.513,1.484,1.4405,1.381,1.30475,1.2115,1.1025,0.↵
972,0.82625,0.66525,0.492,0.30525,0.112999999999999,-0.08925,-0.3045,-0.52575,-0.756749999999999,-0.↵
```

9932500000000001,-1.23525,-1.4777500000000001,-1.72125,-1.96075,-2.1919999999999997,-2.↵
4147499999999997,-2.6252500000000003,-2.82525,-3.009,-3.1655,-3.33025,-3.47675,-3.57675,-3.↵
6734999999999998,-3.7835,-3.90325,-4.0322499999999994,-4.1595,-4.27325,-4.3685,-4.453250000000001,-4.↵
5295000000000005,-4.59825,-4.66375,-4.72575,-4.78275,-4.83925,-4.8950000000000005,-4.94775,-5.↵
0040000000000004,-5.06275,-5.126250000000001,-5.20025,-5.2802500000000006,-5.366499999999999,-5.↵
4635000000000001,-5.564,-5.667,-5.77175,-5.87,-5.96225,-6.0432500000000005,-6.112,-6.173,-6.↵
22625,-6.276750000000001,-6.327000000000001,-6.37675,-6.43475,-6.50175,-6.577249999999999,-6.↵
6665,-6.765000000000001,-6.874750000000001,-6.990749999999999,-7.106750000000001,-7.219,-7.↵
32425,-7.417999999999999,-7.502000000000001,-7.57,-7.625500000000001,-7.669,-7.7017500000000005,-7.↵
7299999999999995,-7.755249999999999,-7.782749999999999,-7.8155,-7.85825,-7.91175,-7.978000000000001,-8.↵
0605,-8.157250000000001,-8.270249999999999,-8.3985,-8.53875,-8.68375,-8.825750000000001,-8.↵
9545,-9.0645,-9.147,-9.195,-9.203999999999999,-9.16525,-9.07825,-8.93925,-8.74625,-8.50225,-8.↵
204749999999999,-7.858999999999999,-7.46775,-7.02975,-6.557499999999999,-6.05175,-5.523,-4.↵
9735000000000005,-4.407500000000001,-3.83225,-3.25325,-2.67575,-2.10425,-1.54725,-1.0085,-0.↵
49525,-0.004500000000000005,0.45325,0.88425,1.28775,1.6632500000000001,2.00875,2.3307499999999997,2.↵
6237500000000002,2.8937500000000003,3.1425,3.3715,3.58275,3.7735,3.94675,4.0985000000000005,4.↵
22825,4.33725,4.42725,4.50125,4.5625,4.615749999999999,4.662999999999999,4.7035,4.743250000000001,4.↵
7837499999999995,4.828,4.879,4.9355,4.9965,5.0575,5.117,5.172750000000001,5.22375,5.268,5.↵
3062499999999995,5.3405000000000005,5.36875,5.39025,5.40475,5.412999999999999,5.4185,5.↵
42,5.426,5.433000000000001,5.442749999999999,5.45425,5.46725,5.481,5.497,5.51225,5.529,5.↵
545,5.556500000000001,5.5665,5.5755,5.584,5.59225,5.602250000000001,5.609999999999999,5.↵
617500000000001,5.6274999999999995,5.6427499999999995,5.66625,5.7,5.7435,5.7945,5.851,5.↵
91125,5.97525,6.03725,6.1034999999999995,6.164499999999999,6.22175,6.26525,6.296499999999999,6.↵
3095,6.30025,6.273,6.224749999999999,6.157,6.069999999999999,5.964749999999999,5.838,5.↵
6907499999999995,5.524500000000001,5.33975,5.140000000000001,4.9262500000000005,4.702,4.↵
469250000000001,4.2305,3.9855,3.743,3.50125,3.2655000000000003,3.0395,2.82375,2.6222499999999997,2.↵
4307499999999997,2.25075,2.08425,1.92875,1.786,1.6587500000000002,1.54025,1.43425,1.3375,1.↵
2475,1.162,1.085,1.0125,0.9497500000000001,0.8920000000000001,0.8415,0.797999999999999,0.↵
756749999999999,0.72025,0.685999999999999,0.653749999999999,0.62325,0.59975,0.57825,0.↵
56075,0.5485,0.54025,0.5355000000000001,0.530999999999999,0.53025,0.534,0.5377500000000001,0.↵
54775,0.55925,0.5737500000000001,0.59125,0.609499999999999,0.63175,0.65525,0.681999999999999,0.↵
7095,0.7422500000000001,0.77675,0.810999999999999,0.844499999999999,0.8750000000000001,0.↵
898,0.91475,0.924,0.929999999999999,0.9285,0.92175,0.91025,0.8920000000000001,0.869,0.↵
842249999999999,0.81025,0.77675,0.74,0.7005,0.65925,0.621,0.5805,0.5425,0.505,0.4692499999999994,0.↵
4440000000000006,0.42275,0.409,0.402,0.4035,0.415,0.4295,0.4485,0.4692499999999994,0.↵
4882499999999996,0.505,0.518,0.521,0.518,0.5035000000000001,0.4822500000000007,0.4525,0.↵
418,0.3807500000000003,0.3455000000000003,0.3105,0.27775,0.25025,0.2227499999999998,0.↵
1992499999999998,0.1807500000000002,0.1685,0.16025,0.154,0.145,0.1342499999999998,0.↵
11975,0.1015,0.0785,0.052,0.0212499999999998,-0.0085,-0.0375,-0.061,-0.0825,-0.0992499999999999,-0.↵
1115000000000002,-0.11975,-0.1242499999999999,-0.1235,-0.1242499999999999,-0.11975,-0.↵
11525,-0.109,-0.1,-0.08925,-0.07475,-0.0595000000000004,-0.0375,-0.00925,0.02225,0.0587500000000004,0.↵
097,0.13725,0.1755000000000002,0.21125,0.2425,0.27,0.296,0.31825,0.33575,0.34875,0.35625,0.↵
3510000000000003,0.331,0.3205,0.31825,0.3175,0.3165,0.32575,0.3455000000000003,0.3737499999999997,0.↵
4075,0.45175,0.502,0.55925,0.618,0.6805,0.745499999999999,0.808,0.8645,0.916999999999999,0.↵
966,1.0125,1.0545,1.0925,1.1245,1.155,1.184,1.21625,1.248999999999999,1.2862500000000001,1.↵
3215,1.358,1.39225,1.42825,1.46175,1.49525,1.525,1.55575,1.58,1.603,1.62575,1.643249999999999,1.↵
661,1.6740000000000002,1.68,1.68,1.67,1.645,1.60525,1.54275,1.45725,1.349,1.2185,1.065,0.↵
8935000000000001,0.7035,0.5005,0.2885,0.0687499999999999,-0.15025,-0.3715,-0.590499999999999,-0.↵
808,-1.023,-1.236,-1.4435,-1.6495,-1.85625,-2.0629999999999997,-2.27125,-2.47425,-2.6717500000000003,-2.↵
8625000000000003,-3.0380000000000003,-3.2005,-3.34775,-3.4775,-3.59875,-3.707,-3.804,-3.↵
8925,-3.97025,-4.04575,-4.12225,-4.20075,-4.28475,-4.3725,-4.4632499999999995,-4.557,-4.↵
65475,-4.74925,-4.84775,-4.94075,-5.03225,-5.11925,-5.1979999999999995,-5.26725,-5.33075,-5.↵
38625,-5.43825,-5.49175,-5.54275,-5.59925,-5.6572499999999994,-5.7227500000000004,-5.795249999999999,-5.↵
88,-5.973,-6.0760000000000005,-6.1845,-6.29875,-6.41475,-6.529999999999999,-6.638249999999999,-6.↵
7475,-6.84975,-6.9215,-6.99775,-7.08625,-7.148000000000001,-7.1754999999999995,-7.1785000000000005,-7.↵
1677500000000001,-7.16175,-7.170749999999999,-7.1975,-7.242500000000001,-7.3035000000000005,-7.↵
380000000000001,-7.46925,-7.5745000000000005,-7.691249999999999,-7.821,-7.967499999999999,-8.↵
12,-8.28775,-8.464,-8.64475,-8.8265,-9.002,-9.163,-9.30475,-9.41925,-9.501,-9.5435,-9.54125,-9.↵

```
49325,-9.39325,-9.2415,-9.034,-8.7775,-8.4725,-8.126,-7.74075,-7.32575,-6.8839999999999995,-6.↵
4202499999999999,-5.9365000000000006,-5.436,-4.9270000000000005,-4.41125,-3.891,-3.373,-2.↵
8535,-2.336,-1.825,-1.32375,-0.8332499999999999,-0.35925,0.09375,0.525,0.92925,1.307,1.↵
6555,1.9775,2.27125,2.53675,2.7725,2.9884999999999997,3.18,3.35475,3.5132499999999998,3.↵
6575,3.7895,3.91475,4.0305,4.1404999999999999,4.245,4.34725,4.444,4.53725,4.628,4.7135000000000001,4.↵
7905000000000001,4.8607499999999995,4.922499999999999,4.9742500000000005,5.02325,5.0652500000000001,5.↵
104,5.13925,5.1720000000000001,5.20625,5.239,5.27125,5.3039999999999999,5.336749999999999,5.↵
365,5.396249999999999,5.423,5.4465,5.4695,5.481,5.489999999999999,5.4947500000000001,5.4925,5.↵
489249999999999,5.48325,5.4765,5.47175,5.471,5.46725,5.46875,5.47325,5.48325,5.49925,5.↵
5237500000000001,5.55725,5.59775,5.6480000000000001,5.7075000000000005,5.7725,5.8450000000000001,5.↵
9220000000000001,6.005,6.089,6.17525,6.256,6.3325000000000005,6.398,6.4475000000000001,6.↵
482,6.49575,6.4879999999999995,6.4612500000000001,6.4094999999999995,6.337,6.240749999999999,6.↵
1225000000000005,5.9845000000000001,5.827999999999999,5.655749999999999,5.46575,5.2642500000000005,5.↵
05225,4.8355000000000001,4.614999999999999,4.39525,4.1762500000000005,3.95825,3.7445,3.5385,3.↵
3385,3.1454999999999997,2.9579999999999997,2.774,2.59775,2.42925,2.2697499999999997,2.1165,1.↵
97225,1.83725,1.709,1.58925,1.48075,1.377,1.284,1.19925,1.12525,1.05825,0.99875,0.94225,0.↵
88725,0.83625,0.78575,0.73775,0.6905,0.6477499999999999,0.60725,0.5737500000000001,0.5455,0.↵
5225,0.50275,0.492,0.483,0.4815,0.483,0.48975,0.5005,0.51725,0.53475,0.557,0.57975,0.60575,0.↵
63625,0.6697500000000001,0.7072499999999999,0.7437499999999999,0.78125,0.8125,0.842249999999999,0.↵
8630000000000001,0.8782500000000001,0.8865,0.88275,0.8734999999999999,0.8522500000000001,0.↵
8217500000000001,0.78275,0.73625,0.6842499999999999,0.631,0.58125,0.5369999999999999,0.↵
502,0.473,0.45175,0.4425,0.4402500000000003,0.4425,0.44925,0.4547500000000004,0.4592500000000005,0.↵
4607499999999999,0.4599999999999996,0.45625,0.4502500000000004,0.4394999999999995,0.↵
4287499999999996,0.4157499999999995,0.4004999999999997,0.38675,0.3707499999999997,0.↵
35925,0.3455000000000003,0.3334999999999996,0.32275,0.30825,0.29675,0.2830000000000003,0.↵
2677500000000004,0.25025,0.23275,0.21125,0.1885,0.16475,0.14125,0.11825,0.09375,0.0717500000000001,0.↵
0510000000000004,0.0297500000000002,0.00925,-0.01075,-0.03275,-0.05425,-0.0785,-0.10225,-0.↵
12825,-0.158,-0.1855,-0.2167500000000003,-0.2442499999999997,-0.26625,-0.2830000000000003,-0.↵
2945000000000004,-0.2975000000000004,-0.2952499999999996,-0.2845,-0.2654999999999996,-0.↵
2395,-0.2052500000000002,-0.1655,-0.1227500000000001,-0.07625,-0.0335,0.0015,0.0342499999999996,0.↵
0595000000000004,0.0785,0.093,0.1,0.1015,0.1,0.097,0.0915,0.09075,0.09225,0.0985,0.10975,0.↵
126,0.14575,0.1677499999999998,0.19525,0.2249999999999998,0.2525,0.28075,0.3127500000000003,0.↵
3455000000000003,0.38525,0.4287499999999996,0.4760000000000003,0.528749999999999,0.↵
586,0.6455,0.70575,0.7675000000000001,0.82625,0.882,0.9285,0.9690000000000001,1.001,1.02475,1.↵
04225,1.0605,1.08025,1.101,1.13075,1.1672500000000001,1.21525,1.2725,1.3405,1.4160000000000001,1.↵
496,1.57625,1.654,1.7235,1.78675,1.8365,1.87225,1.8967500000000002,1.905,1.906499999999999,1.↵
90575,1.879,1.83175,1.77,1.702,1.6219999999999999,1.52675,1.42125,1.3015,1.1657499999999998,1.↵
014,0.84525,0.6629999999999999,0.4700000000000003,0.2595,0.039,-0.18475,-0.4135,-0.644,-0.↵
8704999999999999,-1.0955,-1.3085,-1.5105,-1.7045000000000001,-1.88525,-2.05525,-2.21475,-2.↵
3635,-2.5025,-2.636,-2.765,-2.8907499999999997,-3.0175,-3.14325,-3.27,-3.39725,-3.52475,-3.↵
65375,-3.77875,-3.8985,-4.01075,-4.1122499999999995,-4.20925,-4.293,-4.36775,-4.4350000000000005,-4.↵
4952499999999995,-4.55175,-4.60425,-4.654,-4.708749999999999,-4.76675,-4.8294999999999995,-4.↵
901,-4.9805,-5.0637500000000001,-5.1560000000000001,-5.25275,-5.3572500000000005,-5.4635000000000001,-5.↵
57625,-5.6907499999999995,-5.806,-5.9205,-6.0302500000000006,-6.1355,-6.2295000000000001,-6.↵
3210000000000001,-6.398,-6.462,-6.5125,-6.5482499999999995,-6.575749999999999,-6.59475,-6.↵
6055,-6.61175,-6.61625,-6.62225,-6.639749999999999,-6.668749999999999,-6.717,-6.789499999999999,-6.↵
88175,-7.00375,-7.14425,-7.309,-7.492749999999999,-7.695,-7.906249999999999,-8.1275,-8.↵
3495,-8.56925,-8.7775,-8.9745,-9.146,-9.298,-9.41625,-9.50475,-9.5605,-9.5810000000000001,-9.↵
5665,-9.50925,-9.414,-9.278749999999999,-9.1035,-8.8927500000000001,-8.644,-8.3595,-8.0405,-7.↵
6875,-7.30125,-6.8839999999999995,-6.43625,-5.9540000000000001,-5.471,-4.96675,-4.42725,-3.↵
89325,-3.3845,-2.8945,-2.427,-1.9737500000000001,-1.52675,-1.081,-0.6485000000000001,-0.↵
2350000000000001,0.1565,0.52425,0.8659999999999999,1.18725,1.48775,1.76925,2.03325,2.279,2.↵
51225,2.7297499999999997,2.93275,3.1205,3.2975,3.45925,3.6065,3.743,3.868,3.984,4.093249999999999,4.↵
19475,4.29,4.38225,4.4677500000000006,4.548,4.62025,4.6905,4.754,4.8125,4.87125,4.9285000000000005,4.↵
98125,5.033,5.0790000000000001,5.1247500000000001,5.16825,5.20775,5.242249999999999,5.272749999999999,5.↵
30025,5.319249999999999,5.33,5.335249999999999,5.33,5.3214999999999995,5.3062499999999995,5.↵
288749999999999,5.2705,5.25125,5.2345000000000001,5.2200000000000001,5.2124999999999995,5.↵
20925,5.2170000000000005,5.2330000000000005,5.2620000000000005,5.304749999999999,5.362,5.↵
```

4305,5.51225,5.60375,5.7052499999999995,5.811999999999999,5.9212500000000001,6.0295000000000005,6.↵
1355,6.2355,6.32175,6.39575,6.4515,6.4902500000000005,6.5087500000000001,6.50775,6.4872499999999995,6.↵
44525,6.3865000000000001,6.3125,6.218,6.1087500000000001,5.986,5.8465000000000001,5.6937500000000005,5.↵
52975,5.35425,5.1705000000000005,4.9775,4.7805,4.57925,4.37325,4.16875,3.96725,3.7657499999999997,3.↵
57125,3.379,3.1945,3.016,2.8457500000000002,2.68475,2.5315,2.38275,2.2407500000000002,2.↵
10425,1.973,1.8485,1.7295,1.61975,1.516,1.4205,1.3335,1.25125,1.17875,1.11075,1.049,0.99025,0.↵
936999999999999,0.882,0.82625,0.7735,0.718,0.66525,0.6142500000000001,0.56675,0.5295,0.↵
50425,0.48975,0.486,0.4935,0.5135,0.54475,0.58375,0.6295000000000001,0.6775,0.72475,0.77125,0.↵
81325,0.846749999999999,0.875749999999999,0.8942500000000001,0.903999999999999,0.9055000000000001,0.↵
901749999999999,0.8902500000000001,0.872,0.8522500000000001,0.82775,0.8057500000000001,0.↵
7835,0.763,0.741499999999999,0.72175,0.70425,0.6867500000000001,0.6697500000000001,0.653,0.↵
63775,0.624749999999999,0.612,0.6005,0.590499999999999,0.5805,0.573,0.5660000000000001,0.↵
5615,0.557,0.553999999999999,0.552249999999999,0.547,0.54325,0.53625,0.5295,0.518,0.5065,0.↵
4927499999999997,0.4760000000000003,0.45775,0.4402500000000003,0.42125,0.4012499999999994,0.↵
3845,0.36775,0.3540000000000004,0.344,0.3365,0.331,0.32575,0.322,0.312,0.296,0.27625,0.↵
2487500000000003,0.2145000000000002,0.17475,0.12975,0.080749999999999,0.0305,-0.0205,-0.↵
0687499999999999,-0.1082499999999999,-0.14275,-0.16475,-0.18,-0.187,-0.1855,-0.18,-0.↵
17175,-0.16025,-0.148,-0.1364999999999998,-0.125,-0.1144999999999999,-0.1015,-0.08625,-0.↵
0710000000000001,-0.0495,-0.0282499999999997,-0.004500000000000005,0.01975,0.0427499999999996,0.↵
064,0.08,0.09075,0.0945,0.09225,0.084,0.07325,0.061,0.0465,0.0357500000000004,0.0282499999999997,0.↵
0297500000000002,0.039,0.0572499999999995,0.085499999999999,0.122,0.1677499999999998,0.↵
21825,0.27225,0.3265,0.3777500000000003,0.42725,0.4715,0.508,0.536999999999999,0.5645,0.↵
58525,0.60125,0.61875,0.63625,0.65625,0.67975,0.7095,0.7422500000000001,0.78275,0.82775,0.↵
875749999999999,0.9285,0.9865,1.043,1.10625,1.17025,1.23525,1.30225,1.37025,1.436499999999999,1.↵
5015,1.56775,1.6327500000000001,1.69675,1.7585,1.818749999999999,1.873,1.91725,1.95,1.↵
9685000000000001,1.97,1.9515,1.9105,1.84625,1.7607500000000003,1.6555,1.5305,1.384,1.2215,1.↵
046,0.859,0.6715,0.4775,0.28775,0.097,-0.09375,-0.2845,-0.47075,-0.6577500000000001,-0.↵
84075,-1.02225,-1.20075,-1.37875,-1.55575,-1.73425,-1.9105,-2.085,-2.2560000000000002,-2.↵
4245,-2.5895,-2.7487500000000002,-2.9059999999999997,-3.05475,-3.19525,-3.32875,-3.456,-3.↵
57425,-3.6835,-3.785,-3.878,-3.95975,-4.0397500000000001,-4.123,-4.186249999999999,-4.2365,-4.↵
29075,-4.348,-4.4067500000000001,-4.47225,-4.5425,-4.6235,-4.71575,-4.81725,-4.9255,-5.043,-5.↵
1712500000000001,-5.3062499999999995,-5.44575,-5.5785,-5.7082500000000005,-5.8305000000000001,-5.↵
94025,-6.03725,-6.1187499999999995,-6.1835,-6.2325,-6.26375,-6.2782500000000001,-6.2835,-6.↵
279,-6.273,-6.2615000000000001,-6.255249999999999,-6.255249999999999,-6.26375,-6.289,-6.↵
32925,-6.392,-6.4750000000000005,-6.58025,-6.7040000000000001,-6.8457500000000001,-7.002999999999999,-7.↵
1762500000000005,-7.3647500000000001,-7.56525,-7.77275,-7.9857499999999995,-8.2000000000000001,-8.↵
413,-8.62725,-8.8302500000000001,-9.0270000000000001,-9.20875,-9.36975,-9.50925,-9.62,-9.↵
7015000000000001,-9.7465,-9.75275,-9.7205,-9.64575,-9.52825,-9.3735,-9.17575,-8.941,-8.67,-8.↵
36325,-8.0285,-7.66675,-7.2747500000000001,-6.8610000000000001,-6.4285,-5.9784999999999995,-5.↵
51825,-5.0475,-4.5714999999999995,-4.0969999999999995,-3.627,-3.15775,-2.697,-2.24525,-1.↵
80675,-1.384,-0.975,-0.58525,-0.21125,0.1465,0.4845000000000004,0.80875,1.11625,1.4075,1.↵
684499999999999,1.9455,2.19125,2.42375,2.6397500000000003,2.84275,3.02975,3.20575,3.3715,3.↵
5255,3.6705,3.8055,3.9322500000000002,4.0497499999999995,4.1602500000000004,4.263999999999999,4.↵
364,4.46025,4.553999999999999,4.64475,4.7325,4.818,4.8987500000000001,4.967499999999999,5.↵
0500000000000001,5.114,5.15125,5.20025,5.2604999999999995,5.323,5.3765,5.4145,5.426,5.41925,5.↵
3955,5.36125,5.3245000000000005,5.285,5.2459999999999996,5.2147499999999996,5.1887500000000001,5.↵
17425,5.1735000000000001,5.185,5.20925,5.2467500000000005,5.2962500000000001,5.36125,5.4345,5.↵
5175,5.6114999999999995,5.71125,5.815,5.92425,6.03325,6.14325,6.24925,6.35,6.4430000000000005,6.↵
52775,6.5972500000000001,6.650499999999999,6.6872500000000001,6.70775,6.70775,6.6850000000000005,6.↵
64525,6.587999999999999,6.514,6.4247499999999995,6.3225000000000001,6.209499999999999,6.↵
0845,5.95025,5.806,5.655,5.4932500000000001,5.3245000000000005,5.1475,4.9597500000000005,4.↵
7692499999999995,4.569249999999999,4.3685,4.165,3.96125,3.7605,3.5675,3.37975,3.2035,3.↵
0364999999999998,2.88325,2.7390000000000003,2.60475,2.4802500000000003,2.366,2.253,2.1445,2.↵
0385,1.931,1.822,1.7105,1.59525,1.48075,1.36575,1.2565,1.1520000000000001,1.05125,0.9575,0.↵
873499999999999,0.802499999999999,0.743749999999999,0.6965,0.6645,0.6425000000000001,0.↵
63025,0.62625,0.63175,0.644,0.66,0.68125,0.701249999999999,0.724,0.747,0.766749999999999,0.↵
78275,0.79725,0.808,0.8177500000000001,0.8217500000000001,0.82475,0.8255,0.820249999999999,0.↵
81475,0.804999999999999,0.7927500000000001,0.7775,0.759,0.737,0.71475,0.692,0.6675,0.64475,0.↵

```
6224999999999999,0.6035,0.59125,0.58125,0.5737500000000001,0.573,0.5715,0.573,0.5715,0.↵
5685,0.55925,0.5455,0.528,0.5065,0.4815,0.45625,0.434,0.41125,0.39375,0.3815,0.3737499999999997,0.↵
37225,0.3752500000000003,0.3807500000000003,0.38975,0.39675,0.39975,0.3989999999999997,0.↵
393,0.38225,0.3655,0.3425,0.3175,0.29075,0.2595,0.2305,0.2022499999999999,0.174,0.1487500000000002,0.↵
1227500000000001,0.09525,0.06475,0.032,-0.00375,-0.0435,-0.0847499999999999,-0.125,-0.↵
1662499999999998,-0.2022499999999999,-0.2350000000000001,-0.25875,-0.2739999999999997,-0.↵
2799999999999997,-0.2755,-0.2625,-0.2425,-0.2145000000000002,-0.18375,-0.15025,-0.11525,-0.↵
08625,-0.06175,-0.0442500000000005,-0.03275,-0.03125,-0.0357500000000004,-0.04575,-0.↵
06025,-0.074,-0.09075,-0.1052500000000001,-0.1175000000000001,-0.122,-0.11825,-0.1082499999999999,-0.↵
093,-0.0694999999999999,-0.04125,-0.01,0.0245,0.0572499999999995,0.08925,0.1227500000000001,0.↵
1525,0.1807500000000002,0.2060000000000002,0.2312499999999998,0.254,0.2785,0.3045,0.↵
33025,0.35925,0.39225,0.4250000000000004,0.4599999999999996,0.4927499999999997,0.5287499999999999,0.↵
55925,0.5875,0.61575,0.64325,0.6697500000000001,0.6965,0.7255,0.76,0.7995,0.8507499999999999,0.↵
911,0.9820000000000001,1.06575,1.15975,1.25875,1.3665,1.47175,1.57925,1.6844999999999999,1.↵
7815,1.86925,1.9485,2.01425,2.0660000000000003,2.102,2.1195,2.12175,2.105,2.0705,2.0187500000000003,1.↵
9522500000000003,1.8685,1.7722499999999999,1.6595,1.53725,1.4045,1.262,1.11225,0.9575,0.↵
7965,0.63325,0.46775,0.29675,0.126,-0.04725,-0.22125,-0.3989999999999997,-0.57975,-0.76225,-0.↵
9490000000000001,-1.139,-1.32975,-1.52275,-1.7157499999999999,-1.90425,-2.09275,-2.2735,-2.↵
4475000000000002,-2.6100000000000003,-2.758,-2.896,-3.019,-3.128,-3.225,-3.3105,-3.38275,-3.↵
4485,-3.50725,-3.5645000000000002,-3.62325,-3.68425,-3.753,-3.8315,-3.92225,-4.02675,-4.↵
1427499999999995,-4.271,-4.40825,-4.551,-4.69525,-4.8415,-4.9827499999999999,-5.12,-5.2497500000000001,-5.↵
3695,-5.47725,-5.56875,-5.6457500000000005,-5.71125,-5.7625,-5.80375,-5.83275,-5.85325,-5.↵
86325,-5.86625,-5.86775,-5.864,-5.864,-5.86625,-5.8770000000000001,-5.8930000000000001,-5.↵
9220000000000001,-5.9655,-6.0280000000000005,-6.1065000000000005,-6.20725,-6.32625,-6.46825,-6.↵
6292500000000001,-6.8092500000000005,-7.009,-7.21125,-7.4179999999999999,-7.6439999999999999,-7.↵
8865000000000001,-8.12825,-8.3625,-8.593,-8.81875,-9.0317499999999999,-9.224,-9.39875,-9.↵
5475000000000001,-9.6695000000000001,-9.7595000000000001,-9.81675,-9.8419999999999999,-9.8267499999999999,-9.↵
77325,-9.6847500000000001,-9.5565,-9.39175,-9.19425,-8.9599999999999999,-8.6945,-8.397,-8.↵
0735,-7.7225,-7.3539999999999999,-6.96575,-6.56125,-6.1455,-5.7189999999999999,-5.2887499999999999,-4.↵
8522500000000001,-4.4152499999999995,-3.9787500000000002,-3.5455,-3.1157500000000002,-2.↵
6932500000000004,-2.27825,-1.87225,-1.4785,-1.0925,-0.7224999999999999,-0.36475,-0.01975,0.↵
30825,0.6195,0.9155,1.191,1.4495,1.6885000000000001,1.9120000000000001,2.1195,2.314,2.497,2.↵
6687499999999997,2.8335,2.99375,3.148,3.29975,3.4515,3.5995,3.7475,3.89325,4.03125,4.1640000000000001,4.↵
29225,4.4105,4.51975,4.621,4.71125,4.79125,4.8585,4.9117500000000005,4.9522499999999999,4.↵
9827499999999999,5.0025,5.01325,5.0147499999999999,5.00575,4.988,4.9645,4.9362499999999999,4.↵
90275,4.8705,4.8425,4.8202500000000001,4.8065,4.8050000000000001,4.8149999999999995,4.8415,4.↵
885,4.947,5.0225,5.1095000000000001,5.2109999999999999,5.3199999999999999,5.43375,5.54875,5.↵
6640000000000001,5.77325,5.87925,5.9784999999999995,6.0714999999999995,6.1585,6.23775,6.↵
3072500000000001,6.3705,6.424,6.46675,6.49725,6.5185,6.53375,6.5407499999999999,6.5095,6.↵
4797499999999999,6.4537499999999999,6.3905,6.28825,6.1530000000000005,5.98975,5.81825,5.6465000000000005,5.↵
4795000000000001,5.3122500000000001,5.1422500000000001,4.9674999999999999,4.79275,4.6164999999999999,4.↵
44325,4.27325,4.1085,3.949,3.794,3.6475,3.50425,3.366,3.22875,3.09225,2.95475,2.8205,2.↵
681,2.5382499999999997,2.39575,2.253,2.1125000000000003,1.976,1.84325,1.7175,1.6,1.4869999999999999,1.↵
3847500000000001,1.28925,1.20175,1.123,1.04975,0.988,0.9322499999999999,0.88575,0.8507499999999999,0.↵
82625,0.81175,0.8064999999999999,0.8109999999999999,0.8210000000000001,0.84075,0.8619999999999999,0.↵
88575,0.9095000000000001,0.9285,0.94225,0.9530000000000001,0.9545,0.9497500000000001,0.↵
93925,0.9247500000000001,0.9095000000000001,0.8942500000000001,0.8789999999999999,0.8675,0.↵
85825,0.8574999999999999,0.8612500000000001,0.8637499999999999,0.869,0.8734999999999999,0.↵
8765000000000001,0.8765000000000001,0.87125,0.8652500000000001,0.85375,0.8370000000000001,0.↵
8162499999999999,0.79425,0.7705,0.75075,0.73175,0.7165,0.70575,0.698,0.6935,0.69125,0.68825,0.↵
6859999999999999,0.6842499999999999,0.68125,0.67675,0.6697500000000001,0.66,0.65225,0.64475,0.↵
63475,0.6287499999999999,0.62325,0.6195,0.62025,0.62175,0.624,0.6247499999999999,0.62175,0.↵
6165,0.60275,0.58375,0.55775,0.5225,0.48075,0.43325,0.3807500000000003,0.32875,0.27225,0.↵
2167500000000003,0.167,0.122,0.084,0.055,0.03275,0.0205,0.0174999999999998,0.0212499999999998,0.↵
029,0.03825,0.0495,0.06025,0.06725,0.07325,0.074,0.0725,0.0687499999999999,0.06325,0.0564999999999995,0.↵
05025,0.04725,0.0442500000000005,0.04125,0.03825,0.0357500000000004,0.0349999999999996,0.↵
0357500000000004,0.0349999999999996,0.0335,0.0275,0.0245,0.02225,0.019,0.0212499999999998,0.↵
0282499999999997,0.03975,0.0572499999999995,0.0755,0.10225,0.132,0.16475,0.2,0.2365,0.↵
```

27399999999999997,0.30525,0.33575,0.35775,0.377,0.38975,0.39975,0.40425,0.4075,0.40825,0.↵
41125,0.4135,0.41875,0.42874999999999996,0.44549999999999995,0.46849999999999997,0.49750000000000005,0.↵
534,0.5805,0.63550000000000001,0.7035,0.779,0.8619999999999999,0.9575,1.05825,1.162,1.27175,1.↵
381,1.493,1.60075000000000001,1.703,1.79825,1.8815,1.95225000000000003,2.01025,2.053,2.08050000000000002,2.↵
092,2.08825,2.06825,2.03550000000000003,1.9875,1.9295,1.8599999999999999,1.78,1.69450000000000001,1.↵
60375,1.503,1.3992499999999999,1.2885,1.17025,1.04375,0.90550000000000001,0.7567499999999999,0.↵
5975,0.4235,0.241,0.0465,-0.1565,-0.36775,-0.583,-0.80175000000000001,-1.01700000000000001,-1.↵
22825,-1.429,-1.62125,-1.8005,-1.96750000000000002,-2.121,-2.25975,-2.38575,-2.501,-2.60475,-2.↵
70075,-2.78925,-2.8725,-2.9517499999999997,-3.02975,-3.1075,-3.18975,-3.279,-3.373,-3.476,-3.↵
58725,-3.707,-3.83675,-3.97175,-4.11450000000000005,-4.26175,-4.40825,-4.55775,-4.705,-4.↵
84624999999999995,-4.9805,-5.1069999999999999,-5.22075,-5.3207499999999999,-5.40775,-5.47950000000000001,-5.↵
53975000000000001,-5.58325,-5.6175000000000001,-5.635,-5.6442499999999999,-5.645,-5.6404999999999999,-5.↵
62975,-5.62125,-5.6129999999999999,-5.60925,-5.61375,-5.6282499999999999,-5.6525,-5.69225000000000005,-5.↵
748,-5.82425,-5.918,-6.02950000000000005,-6.1622499999999999,-6.3147499999999999,-6.485,-6.↵
67500000000000001,-6.88025,-7.1,-7.33025,-7.5707499999999999,-7.8155,-8.0635,-8.30925,-8.55025,-8.↵
78150000000000001,-8.9997499999999999,-9.2024999999999999,-9.3865,-9.54275,-9.6755,-9.7755,-9.↵
84349999999999999,-9.87475,-9.87325,-9.835,-9.76400000000000001,-9.658,-9.5215,-9.352,-9.156,-8.↵
92949999999999999,-8.6785,-8.4029999999999999,-8.10325,-7.7842499999999999,-7.44475,-7.08625,-6.↵
71000000000000001,-6.32175,-5.91575,-5.49925,-5.07425,-4.6425,-4.20925,-3.778,-3.35,-2.929,-2.↵
52075,-2.1239999999999997,-1.73725,-1.377,-1.0345,-0.6995,-0.3715,-0.06475,0.2235,0.50575,0.↵
77525,1.0285,1.27175,1.5015,1.72575,1.9395,2.1462499999999998,2.34525,2.5382499999999997,2.↵
7305,2.9212499999999997,3.11125,3.29525000000000002,3.4805,3.6605,3.83825,4.00475,4.16175000000000005,4.↵
3075,4.43725000000000001,4.55175,4.64925,4.72875000000000001,4.78825000000000001,4.82949999999999995,4.↵
85075000000000001,4.8599999999999999,4.85675,4.84325,4.82625,4.80425000000000001,4.78275,4.76525,4.↵
75075,4.74250000000000001,4.74325000000000001,4.7485,4.763,4.78449999999999995,4.81424999999999995,4.↵
85150000000000001,4.89650000000000005,4.947,5.008,5.0757499999999999,5.14825,5.22924999999999995,5.↵
31625,5.40775,5.50775,5.61224999999999995,5.722,5.83575,5.95325000000000001,6.0699999999999999,6.↵
18675,6.2979999999999999,6.4025,6.49350000000000001,6.5705,6.63075000000000001,6.67200000000000001,6.↵
69175000000000001,6.694,6.67200000000000001,6.63075000000000001,6.572,6.49875,6.40949999999999995,6.↵
30575000000000001,6.192,6.07224999999999995,5.9455,5.815,5.67850000000000005,5.53825,5.3925,5.↵
24374999999999995,5.0917499999999999,4.93999999999999995,4.78525,4.63025,4.4769999999999999,4.↵
322,4.1695,4.02075,3.87125,3.727,3.582,3.4385,3.29275,3.1425,2.9884999999999997,2.83200000000000003,2.↵
67100000000000003,2.5085,2.3467499999999997,2.1835,2.02875,1.87825,1.73875,1.61525,1.49925,1.↵
40000000000000001,1.316,1.24275,1.17875,1.1215,1.08725,1.043,0.98725,0.9575,0.94525,0.94525,0.↵
94975000000000001,0.94975000000000001,0.9369999999999999,0.91775,0.898,0.87975,0.8645,0.85375,0.↵
84925000000000001,0.8467499999999999,0.8484999999999999,0.85000000000000001,0.853,0.85225000000000001,0.↵
8507499999999999,0.84925000000000001,0.84,0.83,0.8162499999999999,0.80350000000000001,0.7872499999999999,0.↵
7735,0.76150000000000001,0.75,0.74075,0.7325,0.72700000000000001,0.72025,0.71575,0.70875,0.↵
69875,0.68975,0.67675,0.65925,0.6385,0.61575,0.59125,0.5685,0.54325,0.5225,0.50725,0.4967499999999997,0.↵
4945,0.49899999999999994,0.509,0.52575,0.541,0.56075,0.5775,0.58975,0.59825,0.599,0.592,0.↵
57975,0.55925,0.53025,0.4982499999999997,0.45849999999999996,0.41875,0.377,0.33349999999999996,0.↵
293,0.25325,0.21450000000000002,0.18075000000000002,0.14725000000000002,0.116,0.08625,0.↵
058,0.02975000000000002,0.00375,-0.019,-0.04125,-0.058,-0.0725,-0.08074999999999999,-0.↵
08625,-0.08625,-0.0825,-0.0785,-0.07025,-0.06325,-0.05575000000000001,-0.052,-0.04874999999999995,-0.↵
05100000000000004,-0.05425,-0.06325,-0.07475,-0.08850000000000001,-0.106,-0.1205,-0.13649999999999998,-0.↵
15175,-0.161,-0.167,-0.1725,-0.1685,-0.161,-0.148,-0.12675,-0.1045,-0.07475,-0.042,-0.00675,0.↵
0275,0.064,0.0985,0.12975,0.158,0.183,0.20375,0.2205,0.232,0.2395,0.24025000000000002,0.↵
24175,0.23800000000000002,0.23124999999999998,0.22499999999999998,0.219,0.216,0.21675000000000003,0.↵
21974999999999997,0.232,0.25175000000000003,0.28375,0.32725,0.3845,0.4555,0.5369999999999999,0.↵
631,0.7355,0.84925000000000001,0.96675000000000001,1.088,1.20925000000000002,1.3245,1.43425,1.↵
53875,1.6305,1.7157499999999999,1.79225000000000001,1.85775,1.915,1.9645,2.008,2.04475,2.↵
07525,2.098,2.11875000000000004,2.12775000000000003,2.12775000000000003,2.12025,2.0995,2.0675,2.↵
02325,1.9645,1.8912499999999999,1.80125,1.69825,1.57550000000000001,1.4364999999999999,1.↵
2855,1.12075000000000001,0.94000000000000001,0.75225,0.5555,0.35325,0.14725000000000002,-0.↵
05875000000000004,-0.2625,-0.46074999999999994,-0.6537499999999999,-0.83700000000000001,-1.↵
00925,-1.171,-1.323,-1.46325,-1.5915,-1.70675,-1.815,-1.91425,-2.008,-2.10125,-2.1905,-2.↵
27975,-2.37200000000000003,-2.46975,-2.5735,-2.6839999999999997,-2.80225,-2.929,-3.06175,-3.↵
20125,-3.34925,-3.5020000000000002,-3.6575,-3.81475,-3.97275,-4.12975000000000005,-4.2807499999999999,-4.↵

425,-4.5607500000000005,-4.6875,-4.802,-4.8995,-4.984249999999999,-5.05375,-5.108,-5.15125,-5.↵
18025,-5.1995,-5.207,-5.20775,-5.2010000000000005,-5.19325,-5.1819999999999995,-5.1705000000000005,-5.↵
162,-5.159,-5.16575,-5.1827499999999995,-5.2139999999999995,-5.2620000000000005,-5.3267500000000005,-5.↵
4115,-5.5175,-5.648000000000001,-5.799,-5.970000000000001,-6.16525,-6.374249999999999,-6.↵
600250000000001,-6.8382499999999995,-7.084,-7.337249999999999,-7.59125,-7.846,-8.0955,-8.↵
33675,-8.5685,-8.78675,-8.98975,-9.17425,-9.33925,-9.48025,-9.59775,-9.6885,-9.751999999999999,-9.↵
78475,-9.788499999999999,-9.76025,-9.701500000000001,-9.610000000000001,-9.48475,-9.33225,-9.↵
1445,-8.934000000000001,-8.6945,-8.432,-8.146749999999999,-7.840750000000001,-7.515,-7.↵
17175,-6.8115,-6.4384999999999994,-6.05175,-5.6525,-5.2467500000000005,-4.835500000000001,-4.↵
425,-4.0184999999999995,-3.61475,-3.22025,-2.83575,-2.4665,-2.114,-1.7785,-1.4565,-1.152000000000001,-0.↵
859,-0.5805,-0.3112499999999997,-0.0487499999999995,0.2082500000000002,0.46375,0.71725,0.↵
9705,1.2237500000000001,1.47925,1.7335,1.98525,2.23475,2.479500000000003,2.717500000000002,2.↵
94425,3.157,3.35475,3.533250000000002,3.695,3.83825,3.9635,4.070250000000001,4.1595,4.↵
234999999999999,4.299250000000001,4.35175,4.39375,4.429749999999999,4.4595,4.483000000000005,4.↵
50275,4.515000000000001,4.5265,4.531000000000001,4.529500000000005,4.5265,4.51725,4.508249999999999,4.↵
5104999999999995,4.519,4.522,4.5325,4.554749999999999,4.59525,4.64925,4.71575,4.79575,4.↵
891249999999999,4.99725,5.111750000000001,5.233750000000001,5.3605,5.491,5.61975,5.748749999999999,5.↵
8715,5.985250000000001,6.0975,6.20425,6.29875,6.38575,6.458250000000005,6.517,6.558999999999999,6.↵
582000000000001,6.587999999999999,6.5765,6.545249999999999,6.49575,6.433,6.351500000000001,6.↵
261500000000001,6.162249999999999,6.0577499999999995,5.951,5.841749999999999,5.732,5.6245,5.↵
51525,5.407,5.29775,5.185,5.06825,4.944500000000001,4.8134999999999994,4.67225,4.523499999999999,4.↵
36475,4.201499999999999,4.0329999999999995,3.8575,3.68125,3.50575,3.3295,3.1555,2.989249999999997,2.↵
82675,2.67325,2.528499999999997,2.39025,2.259,2.134000000000003,2.015,1.90125,1.796,1.↵
6975,1.603,1.51525,1.4375,1.367999999999999,1.30625,1.24975,1.20325,1.165749999999998,1.↵
135249999999999,1.114,1.09775,1.08325,1.0705,1.05975,1.0475,1.03375,1.017750000000002,1.↵
001,0.98575,0.966750000000001,0.949000000000001,0.9315,0.918500000000001,0.909500000000001,0.↵
905500000000001,0.904749999999999,0.909500000000001,0.91475,0.924,0.932249999999999,0.↵
937750000000001,0.9415,0.94075,0.932249999999999,0.918500000000001,0.8995,0.875000000000001,0.↵
848499999999999,0.815500000000001,0.78425,0.75525,0.73025,0.70575,0.686750000000001,0.↵
673,0.673750000000001,0.68125,0.668999999999999,0.6875,0.72175,0.73925,0.742250000000001,0.↵
731,0.71025,0.695,0.69275,0.69875,0.70875,0.71875,0.72475,0.723250000000001,0.718,0.707249999999999,0.↵
689000000000001,0.66525,0.63775,0.60575,0.57075,0.535500000000001,0.4935,0.45325,0.410500000000003,0.↵
367,0.325,0.28525,0.246500000000002,0.21125,0.180750000000002,0.15325,0.1335,0.11525,0.↵
1052500000000001,0.09625,0.09225,0.09225,0.09075,0.09,0.08925,0.085499999999999,0.080749999999999,0.↵
07475,0.06325,0.05025,0.0342499999999996,0.0174999999999998,-0.00075,-0.019,-0.03975,-0.↵
0587500000000004,-0.080749999999999,-0.100749999999999,-0.116749999999999,-0.1335,-0.↵
1472500000000002,-0.158,-0.1625,-0.161,-0.1525,-0.13725,-0.114499999999999,-0.084,-0.↵
0495,-0.00925,0.0305,0.07025,0.11075,0.143500000000002,0.16925,0.1885,0.19675,0.1975,0.↵
18625,0.1685,0.14575,0.116749999999999,0.084749999999999,0.0535,0.026,0.00075,-0.01375,-0.↵
01975,-0.01675,-0.0015,0.02525,0.064,0.10975,0.1685,0.234249999999999,0.3075,0.3845,0.↵
46775,0.5515,0.64,0.727000000000001,0.817750000000001,0.907999999999999,0.997999999999999,1.↵
0865,1.17275,1.25875,1.34425,1.429,1.5105,1.59225,1.67075,1.7495,1.82425,1.89275,1.958500000000001,2.↵
015749999999997,2.065250000000003,2.10425,2.127750000000003,2.13775,2.13175,2.108749999999997,2.↵
0675,2.005,1.9265,1.828,1.7105,1.57925,1.43275,1.27325,1.098750000000001,0.91625,0.729250000000001,0.↵
53875,0.34725,0.1565,-0.032,-0.213500000000002,-0.3875,-0.549999999999999,-0.705,-0.↵
850000000000001,-0.98725,-1.114,-1.2315,-1.34125,-1.444999999999998,-1.545750000000002,-1.↵
64575,-1.747249999999997,-1.850250000000002,-1.957,-2.065250000000003,-2.179,-2.300999999999997,-2.↵
42775,-2.562,-2.7045,-2.85175,-3.005249999999998,-3.164,-3.32325,-3.48575,-3.65,-3.8125,-3.↵
97575,-4.1335,-4.2855,-4.4265,-4.557,-4.6745,-4.779,-4.866000000000005,-4.939249999999995,-4.↵
9935,-5.029999999999999,-5.05375,-5.06275,-5.059749999999999,-5.050000000000001,-5.031499999999999,-5.↵
011,-4.989500000000005,-4.968999999999999,-4.954499999999995,-4.94625,-4.94925,-4.966,-4.↵
995,-5.0385,-5.09875,-5.1765,-5.2705,-5.38475,-5.51825,-5.67025,-5.840999999999999,-6.031750000000001,-6.↵
2385,-6.458250000000005,-6.6925,-6.935,-7.18525,-7.43875,-7.6905,-7.93925,-8.181,-8.4145,-8.↵
63425,-8.83875,-9.027000000000001,-9.195,-9.343,-9.471250000000001,-9.57575,-9.658750000000001,-9.↵
716,-9.748750000000001,-9.754999999999999,-9.732000000000001,-9.6795,-9.59775,-9.48475,-9.↵
3415,-9.16525,-8.958499999999999,-8.72125,-8.45425,-8.161999999999999,-7.845249999999999,-7.↵
50875,-7.15325,-6.784750000000001,-6.40725,-6.027249999999996,-5.647250000000005,-5.2705,-4.↵
898750000000001,-4.53425,-4.18175,-3.8375,-3.5035,-3.1785,-2.859499999999997,-2.54525,-2.↵

2315,-1.9195,-1.60825,-1.29625,-0.9804999999999999,-0.6675,-0.3555,-0.045,0.26175,0.5599999999999999,0.↵
8522500000000001,1.1344999999999998,1.4092500000000001,1.6740000000000002,1.92875,2.172,2.↵
4085,2.633,2.8465000000000003,3.0495,3.241,3.421,3.58575,3.7359999999999998,3.87125,3.9895,4.↵
09,4.1755,4.2465,4.3015,4.3457500000000001,4.3770000000000001,4.4014999999999995,4.4174999999999995,4.↵
4265,4.4319999999999995,4.4365000000000006,4.4387500000000001,4.4380000000000001,4.4387500000000001,4.↵
4387500000000001,4.44325,4.4510000000000005,4.4677500000000006,4.48925,4.522,4.567,4.62575,4.↵
70125,4.7897500000000001,4.8942499999999995,5.011,5.1407500000000001,5.2795000000000005,5.↵
42525,5.57775,5.7275,5.87075,6.005,6.128,6.2385,6.32925,6.404999999999999,6.465999999999999,6.↵
5087500000000001,6.5392500000000001,6.55375,6.5545000000000001,6.5482499999999995,6.53525,6.↵
514,6.4895,6.4582500000000005,6.4232499999999995,6.38425,6.3377500000000001,6.2875,6.227,6.↵
15775,6.0775000000000001,5.98975,5.89,5.783,5.66625,5.52975,5.38625,5.2475000000000005,5.↵
10875,4.95675,4.802,4.647749999999999,4.491499999999999,4.3290000000000001,4.1665,3.99925,3.↵
83225,3.6637500000000003,3.4935,3.31875,3.14875,2.9855,2.82675,2.6717500000000003,2.5229999999999997,2.↵
3872500000000003,2.26125,2.14775,2.04075,1.94475,1.8585,1.7760000000000002,1.70375,1.63575,1.↵
5709999999999997,1.50825,1.4465000000000001,1.3885,1.33275,1.280249999999999,1.23225,1.↵
188,1.149,1.1155,1.088,1.06425,1.0475,1.03375,1.0255,1.0185,1.014,1.0125,1.00925,1.0070000000000001,1.↵
0017500000000001,0.995749999999999,0.98875,0.978749999999999,0.967499999999999,0.952249999999999,0.↵
935249999999999,0.9155,0.8965000000000001,0.87425,0.8500000000000001,0.82775,0.802499999999999,0.↵
7805000000000001,0.759,0.7385,0.718,0.7005,0.685999999999999,0.67225,0.66075,0.653,0.64925,0.↵
64925,0.653749999999999,0.6637500000000001,0.676,0.685999999999999,0.701249999999999,0.↵
713249999999999,0.7270000000000001,0.73775,0.74475,0.7462500000000001,0.74075,0.728499999999999,0.↵
711,0.6890000000000001,0.66,0.6295000000000001,0.59425,0.5555,0.51725,0.47825,0.4394999999999995,0.↵
4035,0.3655,0.3295,0.296,0.264,0.23275,0.2045,0.18,0.15725,0.13875,0.124249999999999,0.↵
11075,0.106,0.10225,0.1015,0.099249999999999,0.1,0.097,0.09225,0.07625,0.061,0.0442500000000005,0.↵
00525,-0.0442500000000005,-0.07775,-0.100749999999999,-0.11525,-0.13125,-0.15025,-0.↵
17775,-0.203,-0.2220000000000003,-0.2380000000000002,-0.24175,-0.234249999999999,-0.↵
219,-0.19525,-0.16325,-0.125,-0.08325,-0.04125,-0.003,0.032,0.0625,0.087,0.10225,0.11075,0.↵
1115000000000002,0.1052500000000001,0.0915,0.07325,0.0495,0.0212499999999998,-0.00375,-0.↵
032,-0.058,-0.07925,-0.0945,-0.10375,-0.1052500000000001,-0.0985,-0.084749999999999,-0.↵
064,-0.0342499999999996,0.0022500000000003,0.0427499999999996,0.09075,0.1420000000000002,0.↵
196,0.2562500000000003,0.319,0.38225,0.4502500000000004,0.51875,0.5935,0.6705,0.753,0.↵
84075,0.92925,1.02625,1.1245,1.229,1.3345,1.44275,1.55475,1.66475,1.7760000000000002,1.↵
8807500000000001,1.98125,2.07225,2.15075,2.2155,2.26525,2.29875,2.30875,2.3032500000000002,2.↵
2765,2.2285,2.1582500000000002,2.0715,1.9645,1.84475,1.71125,1.56775,1.41375,1.25425,1.↵
0902500000000002,0.9247500000000001,0.760749999999999,0.59825,0.4365,0.2815,0.129,-0.0174999999999998,-0.↵
15725,-0.29075,-0.418,-0.53875,-0.653749999999999,-0.7645,-0.86975,-0.972,-1.072749999999999,-1.↵
175,-1.27875,-1.387,-1.502249999999999,-1.6235,-1.7555,-1.8960000000000001,-2.0425,-2.↵
20325,-2.3682499999999997,-2.5405,-2.7175000000000002,-2.89775,-3.0785,-3.25775,-3.4354999999999998,-3.↵
6065,-3.7675,-3.923,-4.0665,-4.19625,-4.316,-4.419,-4.508249999999999,-4.58225,-4.64325,-4.↵
68675,-4.718,-4.7325,-4.73625,-4.73175,-4.7165,-4.699,-4.676749999999999,-4.654,-4.631749999999999,-4.↵
62025,-4.61275,-4.616499999999999,-4.63325,-4.66525,-4.71575,-4.786,-4.8759999999999994,-4.↵
984999999999999,-5.11325,-5.2604999999999995,-5.426,-5.606,-5.80375,-6.0075,-6.223999999999999,-6.↵
44825,-6.679499999999999,-6.914499999999999,-7.15325,-7.3937500000000001,-7.634,-7.875,-8.↵
11225,-8.346499999999999,-8.5725,-8.792,-8.9965000000000001,-9.188749999999999,-9.36275,-9.↵
513,-9.636,-9.7305000000000001,-9.79375,-9.82125,-9.817499999999999,-9.774,-9.69625,-9.587,-9.↵
44675,-9.277249999999999,-9.08425,-8.86525,-8.62875,-8.37475,-8.1070000000000001,-7.8270000000000001,-7.↵
53625,-7.23425,-6.926,-6.61,-6.28975,-5.9632499999999995,-5.635,-5.304749999999999,-4.9742500000000005,-4.↵
646249999999999,-4.319,-3.99475,-3.67125,-3.35075,-3.02975,-2.7115,-2.3924999999999996,-2.↵
07225,-1.754,-1.43275,-1.11225,-0.792,-0.473,-0.15725,0.154,0.4555,0.75225,1.03675,1.3122500000000001,1.↵
57325,1.82025,2.05775,2.28125,2.49325,2.6932500000000004,2.8815,3.05625,3.22275,3.3715,3.↵
51025,3.63475,3.743,3.83675,3.91,3.968,4.0085000000000001,4.0322499999999994,4.042,4.0405000000000001,4.↵
02825,4.0115,3.99025,3.9735,3.959,3.9572499999999997,3.96725,3.99475,4.0375000000000005,4.↵
0969999999999995,4.1732499999999995,4.264749999999999,4.37,4.48525,4.609,4.7355,4.8675000000000001,4.↵
998749999999999,5.1270000000000001,5.25275,5.3725000000000005,5.489249999999999,5.59775,5.↵
70075,5.79825,5.88925,5.973,6.05175,6.1225000000000005,6.186,6.2385,6.282749999999999,6.↵
3165,6.3415,6.358999999999999,6.36525,6.363,6.35225,6.3317499999999995,6.3065,6.276,6.23625,6.↵
192,6.1385000000000005,6.07975,6.0104999999999995,5.9334999999999996,5.8465000000000001,5.↵
748749999999999,5.639,5.51825,5.3870000000000005,5.2475000000000005,5.0995,4.9422500000000005,4.↵

```
779, 4.61275, 4.4457499999999999, 4.2784999999999999, 4.1129999999999995, 3.949, 3.7872500000000002, 3.↵
6285, 3.47525, 3.32575, 3.1815, 3.04175, 2.9082500000000002, 2.77775, 2.65125, 2.52675, 2.40625, 2.↵
291, 2.179, 2.073, 1.9745, 1.8800000000000001, 1.78975, 1.7075, 1.6320000000000001, 1.56325, 1.5015, 1.↵
44575, 1.397, 1.3565, 1.31825, 1.28475, 1.2534999999999998, 1.22675, 1.20325, 1.18025, 1.15975, 1.↵
1520000000000001, 1.1475000000000002, 1.1344999999999998, 1.1215, 1.11325, 1.1085, 1.101, 1.0925, 1.↵
08425, 1.0705, 1.0574999999999999, 1.039, 1.02, 0.9972500000000001, 0.9705, 0.9415, 0.9125, 0.88575, 0.↵
8612500000000001, 0.8347500000000001, 0.81475, 0.7965, 0.7865000000000001, 0.779, 0.77525, 0.77525, 0.↵
7797499999999999, 0.7872499999999999, 0.79575, 0.808, 0.8210000000000001, 0.8347500000000001, 0.↵
8460000000000001, 0.85825, 0.869, 0.87975, 0.8880000000000001, 0.8942500000000001, 0.8965000000000001, 0.↵
8942500000000001, 0.885, 0.87125, 0.8507499999999999, 0.82325, 0.78975, 0.7515, 0.711, 0.666, 0.↵
62175, 0.57825, 0.53475, 0.4967499999999997, 0.4629999999999997, 0.4365, 0.4135, 0.396, 0.3845, 0.↵
37625, 0.3737499999999997, 0.373, 0.3737499999999997, 0.3745, 0.3715, 0.367, 0.35775, 0.34325, 0.↵
32575, 0.30125, 0.27075, 0.2372500000000002, 0.20075, 0.16025, 0.11975, 0.0755, 0.03275, -0.006, -0.↵
04125, -0.068, -0.08925, -0.1030000000000001, -0.109, -0.106, -0.09775, -0.08, -0.0587500000000004, -0.↵
03275, -0.002250000000000003, 0.029750000000000002, 0.06025, 0.09, 0.11525, 0.13575, 0.155, 0.↵
16325, 0.167, 0.1662499999999998, 0.15875, 0.1465, 0.129, 0.10975, 0.08625, 0.06325, 0.039, 0.0174999999999998, -0.↵
00375, -0.01975, -0.0342499999999996, -0.045, -0.0495, -0.0487499999999995, -0.045, -0.035750000000000004, -0.↵
023, 0.0, 0.02375, 0.0349999999999996, 0.0725, 0.11975, 0.151, 0.174, 0.18775, 0.2045, 0.2305, 0.↵
2755, 0.33725, 0.4135, 0.4989999999999994, 0.59425, 0.69575, 0.80725, 0.9247500000000001, 1.0475, 1.↵
1764999999999999, 1.30625, 1.4375, 1.5685, 1.6945000000000001, 1.81425, 1.9265, 2.028, 2.1187500000000004, 2.↵
1950000000000003, 2.2567500000000003, 2.3009999999999997, 2.3285, 2.3369999999999997, 2.327, 2.↵
2965, 2.2485, 2.182, 2.0995, 1.9997500000000001, 1.8875, 1.76, 1.62425, 1.48475, 1.33825, 1.191, 1.↵
04375, 0.89875, 0.7567499999999999, 0.621, 0.48975, 0.3655, 0.24725, 0.1342499999999998, 0.02525, -0.↵
0825, -0.183, -0.28525, -0.3889999999999996, -0.49125, -0.599, -0.70875, -0.82325, -0.943, -1.072, -1.↵
20625, -1.34725, -1.496, -1.65175, -1.818, -1.9875, -2.16675, -2.3497500000000002, -2.53675, -2.↵
72675, -2.9175, -3.106, -3.292, -3.47125, -3.64, -3.801, -3.94525, -4.07725, -4.1884999999999994, -4.↵
2832500000000001, -4.3610000000000001, -4.4167499999999995, -4.4555, -4.4792499999999995, -4.48825, -4.↵
4845000000000001, -4.473, -4.45475, -4.4334999999999996, -4.4152499999999995, -4.3975, -4.3885, -4.↵
3862499999999995, -4.3885, -4.4060000000000001, -4.4319999999999995, -4.4670000000000005, -4.↵
5135000000000005, -4.5692499999999999, -4.63875, -4.71725, -4.8095, -4.91875, -5.04075, -5.18025, -5.↵
3359999999999999, -5.51075, -5.7022499999999999, -5.9105, -6.13625, -6.3735, -6.62525, -6.8839999999999995, -7.↵
14875, -7.41575, -7.67975, -7.9415, -8.1939999999999999, -8.4349999999999999, -8.66325, -8.87225, -9.↵
0622499999999999, -9.22925, -9.375, -9.494, -9.5885, -9.6542500000000001, -9.69, -9.697, -9.6740000000000001, -9.↵
6245, -9.54425, -9.439, -9.30475, -9.147, -8.96825, -8.7715, -8.5595, -8.3312499999999999, -8.0925, -7.↵
8407500000000001, -7.5812500000000001, -7.315, -7.0435, -6.76575, -6.482, -6.1905, -5.8907500000000001, -5.↵
5855000000000001, -5.2742499999999999, -4.95675, -4.63475, -4.30675, -3.9765, -3.6445, -3.3105, -2.↵
9785, -2.64675, -2.32, -1.9975, -1.6785, -1.36325, -1.052, -0.743, -0.4365, -0.1305, 0.174, 0.4760000000000003, 0.↵
77525, 1.072, 1.36325, 1.645, 1.91725, 2.176, 2.4154999999999998, 2.6374999999999997, 2.83425, 3.↵
0082500000000003, 3.15775, 3.2845, 3.3875, 3.4705, 3.5332500000000002, 3.579, 3.61325, 3.63625, 3.↵
6545, 3.66825, 3.67975, 3.691, 3.7055, 3.72325, 3.746, 3.7735, 3.80925, 3.85425, 3.90625, 3.968, 4.↵
036, 4.1175000000000001, 4.2082500000000005, 4.3075, 4.4135, 4.5265, 4.64475, 4.766, 4.88975, 5.01175, 5.↵
13375, 5.25125, 5.3635, 5.4695, 5.5702500000000001, 5.6640000000000001, 5.7517499999999995, 5.8312500000000001, 5.↵
9045, 5.9715, 6.0325, 6.0912500000000005, 6.14325, 6.1905, 6.23625, 6.276, 6.3095, 6.3377500000000001, 6.↵
3589999999999999, 6.369, 6.3727500000000001, 6.3645, 6.3445, 6.3132499999999999, 6.26825, 6.2102499999999999, 6.↵
14025, 6.0585, 5.9647499999999996, 5.8625, 5.7502499999999999, 5.6305000000000005, 5.5044999999999999, 5.↵
3735, 5.2375, 5.09725, 4.9544999999999995, 4.8065, 4.6585, 4.506, 4.3525, 4.19775, 4.04275, 3.8895, 3.↵
73375, 3.582, 3.43325, 3.28675, 3.144, 3.0052499999999998, 2.87025, 2.7375, 2.6100000000000003, 2.↵
48875, 2.3720000000000003, 2.2567500000000003, 2.1492500000000003, 2.0425, 1.94475, 1.85325, 1.↵
7670000000000001, 1.68675, 1.61275, 1.5495, 1.49225, 1.44275, 1.4007500000000002, 1.36575, 1.3345, 1.↵
30925, 1.2862500000000001, 1.265, 1.24425, 1.22525, 1.20475, 1.184, 1.1635, 1.1405, 1.1192499999999999, 1.↵
0987500000000001, 1.07575, 1.0545, 1.0314999999999999, 1.0047499999999998, 0.9795, 0.9537500000000001, 0.↵
9247500000000001, 0.89875, 0.872, 0.84375, 0.8162499999999999, 0.78975, 0.76525, 0.7454999999999999, 0.↵
731, 0.7224999999999999, 0.71725, 0.71875, 0.7255, 0.73775, 0.75375, 0.7735, 0.79875, 0.8225, 0.8460000000000001, 0.↵
8659999999999999, 0.8805000000000001, 0.8909999999999999, 0.8965000000000001, 0.895, 0.8835000000000001, 0.↵
86975, 0.84525, 0.81475, 0.7805000000000001, 0.7437499999999999, 0.69425, 0.647, 0.605, 0.5715, 0.↵
53475, 0.50425, 0.47925, 0.4584999999999996, 0.4454999999999995, 0.43875, 0.43325, 0.4325, 0.↵
43325, 0.4349999999999994, 0.438, 0.4365, 0.43175, 0.42125, 0.406, 0.3837500000000004, 0.35625, 0.↵
322, 0.2815, 0.2402500000000002, 0.19225, 0.1472500000000002, 0.0992499999999999, 0.0535, 0.↵
```

00925,-0.029,-0.06325,-0.08925,-0.11225,-0.12424999999999999,-0.132,-0.1335,-0.125,-0.11299999999999999,-0.↵
0985,-0.077,-0.055,-0.029,-0.00675,0.01375,0.03275,0.0495,0.058,0.0655,0.06725,0.064,0.↵
05724999999999999,0.0495,0.03825,0.0275,0.01675,0.00525,-0.003,-0.01,-0.01825,-0.02525,-0.↵
032,-0.035750000000000004,-0.042,-0.04874999999999999,-0.05425,-0.064,-0.07175000000000001,-0.↵
0825,-0.0915,-0.097,-0.10300000000000001,-0.10300000000000001,-0.10375,-0.097,-0.08474999999999999,-0.↵
0665,-0.0465,-0.01675,0.01975,0.0665,0.125,0.19375,0.27075,0.35925,0.45775,0.5660000000000001,0.↵
6875,0.8170000000000001,0.95525,1.097,1.2449999999999999,1.39325,1.5395,1.6815,1.8135,1.↵
9355,2.0455,2.1385,2.214,2.27125,2.311,2.32925,2.33,2.311,2.2735,2.2225,2.1552499999999997,2.↵
076,1.9875,1.88975,1.78825,1.67925,1.56775,1.4465000000000001,1.332,1.22,1.0895,0.96425,0.↵
856,0.76225,0.67675,0.59125,0.502,0.406,0.3105,0.21125,0.11449999999999999,0.016,-0.08175,-0.↵
183,-0.28675,-0.39975,-0.52025,-0.6537499999999999,-0.79725,-0.9530000000000001,-1.12225,-1.↵
30375,-1.49525,-1.6945000000000001,-1.899,-2.105,-2.311,-2.513,-2.7085000000000004,-2.89525,-3.↵
07225,-3.2365,-3.38675,-3.521,-3.64375,-3.7504999999999997,-3.84375,-3.92225,-3.987,-4.↵
04275,-4.0840000000000005,-4.1137500000000005,-4.1327500000000001,-4.1457500000000005,-4.↵
152,-4.14875,-4.141249999999999,-4.1290000000000004,-4.1167500000000001,-4.1045,-4.093999999999999,-4.↵
08775,-4.091,-4.0992500000000005,-4.12225,-4.152,-4.19775,-4.2565,-4.332,-4.4227500000000001,-4.↵
53025000000000006,-4.65625,-4.80275,-4.9645,-5.14375,-5.3420000000000005,-5.558,-5.7885,-6.↵
0325,-6.28975,-6.555999999999999,-6.8275000000000001,-7.1022500000000001,-7.376,-7.643999999999999,-7.↵
9047500000000001,-8.1557500000000001,-8.3885,-8.603,-8.7967500000000001,-8.96375,-9.106499999999999,-9.↵
22475,-9.317,-9.38275,-9.4245,-9.443,-9.439,-9.414,-9.3705,-9.3085,-9.227,-9.127,-9.0087500000000001,-8.↵
87225,-8.718,-8.5435,-8.3495,-8.13675,-7.909499999999999,-7.6645,-7.4075000000000001,-7.↵
1387500000000001,-6.862749999999999,-6.5795,-6.2935000000000001,-6.00425,-5.716,-5.427499999999999,-5.↵
1385,-4.8485,-4.55775,-4.265499999999999,-3.96725,-3.6605,-3.35075,-3.03125,-2.7045,-2.↵
3705,-2.03025,-1.68525,-1.3345,-0.98575,-0.63775,-0.2952499999999999,0.04125,0.367,0.6819999999999999,0.↵
9804999999999999,1.2642499999999999,1.52975,1.7760000000000002,2.0012499999999998,2.20575,2.↵
391,2.55575,2.7015000000000002,2.826,2.93425,3.0235000000000003,3.0945,3.157,3.20575,3.↵
24775,3.2815,3.3105,3.3385,3.3645,3.39275,3.427,3.4675,3.51725,3.57275,3.63775,3.7125,3.↵
7925,3.8795,3.97275,4.0695000000000001,4.16875,4.2725,4.3747500000000001,4.476999999999999,4.↵
583,4.686,4.7882500000000001,4.88975,4.99125,5.0935,5.19475,5.2940000000000005,5.39475,5.↵
4925,5.58925,5.684,5.77325,5.8602500000000001,5.9395000000000001,6.0127500000000005,6.08225,6.↵
14325,6.194249999999999,6.239999999999999,6.2752500000000001,6.300999999999999,6.32175,6.↵
33075,6.331749999999999,6.3210000000000001,6.299499999999999,6.26825,6.22625,6.16925,6.↵
1027499999999999,6.023499999999999,5.93425,5.83725,5.72975,5.6145,5.4932500000000001,5.367999999999999,5.↵
24075,5.1110000000000001,4.976,4.8385,4.69975,4.5592500000000005,4.419,4.27175,4.125249999999999,3.↵
978,3.8285,3.6805,3.5325,3.38275,3.23725,3.09225,2.9525,2.819,2.68925,2.56725,2.44975,2.↵
3385,2.2384999999999997,2.14075,2.0500000000000003,1.9692500000000002,1.88825,1.81725,1.↵
75175,1.69075,1.6365,1.5885,1.54275,1.50525,1.4695,1.43575,1.4085,1.3832499999999999,1.↵
36025,1.3390000000000002,1.3145,1.291,1.2665,1.2382499999999999,1.21,1.1764999999999999,1.↵
14125,1.10325,1.0635000000000001,1.02475,0.985,0.94675,0.9087500000000001,0.8757499999999999,0.↵
8467499999999999,0.8225,0.8042499999999999,0.795,0.78975,0.79425,0.8057500000000001,0.8225,0.↵
8422499999999999,0.8659999999999999,0.8927499999999999,0.9169999999999999,0.9415,0.9634999999999999,0.↵
9787499999999999,0.98875,0.9917499999999999,0.98725,0.9787499999999999,0.9605,0.9344999999999999,0.↵
9047499999999999,0.87125,0.8354999999999999,0.8009999999999999,0.7645,0.73025,0.69875,0.↵
673,0.65075,0.63325,0.62025,0.6102500000000001,0.605,0.6035,0.60725,0.612,0.615,0.61725,0.↵
6134999999999999,0.608,0.59425,0.5745,0.547,0.515,0.47600000000000003,0.4325,0.3845,0.331,0.↵
27775,0.2265,0.18,0.13575,0.097,0.06325,0.03499999999999999,0.01525,0.0,-0.0085,-0.013,-0.↵
01525,-0.01525,-0.01,-0.004500000000000005,0.0,0.006,0.01,0.0145,0.019,0.0245,0.0365,0.↵
04725,0.04874999999999999,0.04874999999999999,0.051000000000000004,0.05425,0.055,0.055,0.↵
05425,0.0535,0.05425,0.051000000000000004,0.0435,0.0365,0.0335,0.02675,0.01825,0.0085,-0.↵
004500000000000005,-0.017499999999999998,-0.032,-0.05025,-0.06949999999999999,-0.08850000000000001,-0.↵
10975,-0.132,-0.154,-0.17625,-0.1975,-0.21525,-0.22725,-0.23424999999999999,-0.23500000000000001,-0.↵
22499999999999998,-0.20525000000000002,-0.17175,-0.122,-0.061,0.016,0.1075,0.21275,0.331,0.↵
4615,0.59825,0.7437499999999999,0.8909999999999999,1.0415,1.191,1.336,1.474,1.60525,1.72725,1.↵
8365,1.934,2.0187500000000003,2.086,2.1392499999999997,2.176,2.19875,2.20325,2.1927499999999998,2.↵
166,2.12325,2.0652500000000003,1.9975,1.91875,1.8325,1.73725,1.641,1.54275,1.448,1.352,1.↵
2605,1.17275,1.0895,1.01175,0.9384999999999999,0.8704999999999999,0.8049999999999999,0.↵
73625,0.66525,0.58975,0.509,0.42050000000000004,0.32275,0.21675000000000003,0.10074999999999999,-0.↵
029750000000000002,-0.16774999999999998,-0.3165,-0.47450000000000003,-0.644,-0.8187500000000001,-1.↵

```
0017500000000001,-1.1925,-1.38625,-1.58225,-1.78,-1.97675,-2.1767499999999997,-2.36975,-2.↵
5597499999999997,-2.74125,-2.9152500000000003,-3.07775,-3.228,-3.3674999999999997,-3.49425,-3.↵
60575,-3.7039999999999997,-3.7872500000000002,-3.8552500000000003,-3.9114999999999998,-3.↵
94525,-3.97725,-4.0070000000000001,-4.00075,-3.98625,-3.98325,-3.98875,-4.0015,-4.01375,-4.↵
016,-4.00625,-3.9895,-3.9735,-3.96425,-3.96425,-3.97725,-4.0077500000000001,-4.055000000000001,-4.↵
1244999999999999,-4.21375,-4.3282500000000001,-4.4647499999999996,-4.6235,-4.80275,-5.00575,-5.↵
223,-5.45425,-5.69925,-5.9517500000000005,-6.208749999999999,-6.47125,-6.73225,-6.9915,-7.↵
2440000000000001,-7.489,-7.7255,-7.9545,-8.16875,-8.373999999999999,-8.56325,-8.741,-8.89975,-9.↵
0407500000000001,-9.16375,-9.26525,-9.346,-9.4055,-9.43975,-9.44975,-9.43525,-9.39625,-9.↵
33375,-9.248249999999999,-9.141499999999999,-9.016499999999999,-8.87525,-8.7165,-8.54575,-8.↵
3625,-8.1695000000000001,-7.967499999999999,-7.7545,-7.5370000000000001,-7.312,-7.079249999999999,-6.↵
8412500000000005,-6.5955,-6.343,-6.083,-5.812749999999999,-5.536,-5.2475000000000005,-4.↵
952999999999999,-4.646999999999999,-4.3282500000000001,-4.0032499999999995,-3.66825,-3.325,-2.↵
9732499999999997,-2.616,-2.2544999999999997,-1.88975,-1.52425,-1.1605,-0.800999999999999,-0.↵
45025000000000004,-0.10975,0.216,0.52425,0.8162499999999999,1.08575,1.3335,1.55875,1.7607500000000003,1.↵
944,2.1035000000000004,2.2445,2.3689999999999998,2.4787500000000002,2.5757499999999998,2.↵
662,2.7405,2.8129999999999997,2.8815,2.948,3.01125,3.074,3.13725,3.19975,3.263,3.32875,3.↵
39725,3.4645,3.5385,3.614,3.6942500000000003,3.77725,3.86425,3.9565,4.052,4.1505,4.251,4.↵
354,4.45775,4.56325,4.6685,4.7730000000000001,4.8805000000000005,4.9857499999999995,5.092499999999999,5.↵
20025,5.3062499999999995,5.412249999999999,5.51825,5.62125,5.7227500000000004,5.82275,5.↵
91975,6.012,6.0975,6.17375,6.24225,6.29725,6.343749999999999,6.3805,6.406499999999999,6.↵
4232499999999995,6.4292500000000001,6.4255,6.411,6.389499999999999,6.3560000000000001,6.3155,6.↵
2630000000000001,6.20275,6.1339999999999995,6.055499999999999,5.9677500000000006,5.872999999999999,5.↵
7700000000000005,5.6610000000000005,5.54575,5.42375,5.2970000000000001,5.16575,5.029249999999999,4.↵
8919999999999995,4.75225,4.61125,4.4662500000000005,4.322,4.1762500000000005,4.0329999999999995,3.↵
88725,3.74225,3.601,3.457,3.31875,3.17925,3.045,2.9145,2.787,2.6687499999999997,2.555,2.↵
44975,2.3529999999999998,2.262,2.179,2.1065,2.0385,1.9805,1.92875,1.88375,1.841,1.80425,1.↵
77,1.7365,1.703,1.667,1.62975,1.58925,1.54425,1.497,1.4465000000000001,1.3915,1.3375,1.↵
28325,1.22825,1.17725,1.1275,1.081,1.0415,1.0070000000000001,0.9804999999999999,0.9575,0.↵
9445,0.9377500000000001,0.9369999999999999,0.9415,0.9530000000000001,0.9667500000000001,0.↵
985,1.00325,1.0215,1.039,1.05275,1.065,1.07125,1.0705,1.06275,1.04975,1.03,1.00325,0.97425,0.↵
94225,0.9087500000000001,0.8750000000000001,0.84525,0.8177500000000001,0.79425,0.77525,0.↵
76225,0.75375,0.74925,0.75,0.75075,0.75525,0.7575000000000001,0.75825,0.75525,0.74925,0.↵
74,0.72475,0.70275,0.67825,0.647,0.6102500000000001,0.5737500000000001,0.53025,0.4890000000000005,0.↵
44325,0.39675,0.35175,0.30674999999999997,0.264,0.22425,0.187,0.1525,0.1242499999999999,0.↵
09525,0.07625,0.058,0.04125,0.0305,0.02225,0.01375,0.0085,0.00525,0.006,0.004500000000000005,0.↵
004500000000000005,0.00525,0.00675,0.00675,0.00675,0.00675,0.00675,0.006,0.00775,0.00525,0.↵
0085,0.0115,0.01749999999999998,0.02375,0.02975000000000002,0.03575000000000004,0.04425000000000005,0.↵
05025,0.055,0.05875000000000004,0.058,0.048749999999999995,0.03424999999999996,0.0145,-0.↵
01375,-0.0465,-0.08625,-0.12825,-0.17325000000000002,-0.21974999999999997,-0.26475,-0.3075,-0.↵
344,-0.37625,-0.39825,-0.41125,-0.41650000000000004,-0.40675,-0.38825,-0.357,-0.30975,-0.↵
251750000000000003,-0.1815,-0.09525,0.0,0.106,0.22799999999999998,0.3585,0.48675,0.624,0.↵
779,0.9377500000000001,1.0887499999999999,1.2375,1.3832499999999999,1.5182499999999999,1.↵
64425,1.7585,1.854,1.934,1.99575,2.037,2.0622499999999997,2.069,2.06075,2.0425,2.00875,1.↵
9685000000000001,1.91875,1.863,1.805,1.74475,1.6844999999999999,1.6272499999999999,1.56625,1.↵
50675,1.451,1.397,1.34425,1.29325,1.24125,1.1895,1.1352499999999999,1.07725,1.0162499999999999,0.↵
9482499999999999,0.8757499999999999,0.79425,0.7035,0.6042500000000001,0.4935,0.37,0.2350000000000001,0.↵
0885000000000001,-0.06874999999999999,-0.2365,-0.41275,-0.59825,-0.7872499999999999,-0.↵
98125,-1.17575,-1.37025,-1.56475,-1.75775,-1.947,-2.13175,-2.31175,-2.48575,-2.6565,-2.↵
819,-2.9732499999999997,-3.1189999999999998,-3.25475,-3.37825,-3.4905,-3.58725,-3.67125,-3.↵
7359999999999998,-3.7865,-3.82,-3.8375,-3.8415,-3.83075,-3.81,-3.785,-3.75375,-3.7215,-3.↵
691,-3.66125,-3.643,-3.63475,-3.64,-3.659,-3.69575,-3.74825,-3.8185,-3.90475,-4.01,-4.1320000000000001,-4.↵
27025,-4.4265,-4.596,-4.78125,-4.9805,-5.1925,-5.41675,-5.651,-5.8930000000000001,-6.14175,-6.↵
3935,-6.648999999999999,-6.903,-7.1572499999999994,-7.4035,-7.643999999999999,-7.8727500000000001,-8.↵
091,-8.2925,-8.47625,-8.64475,-8.79375,-8.9310000000000001,-9.0392500000000001,-9.124749999999999,-9.↵
21025,-9.2675,-9.28875,-9.278749999999999,-9.24225,-9.190249999999999,-9.127,-9.05375,-8.↵
9722499999999999,-8.87675,-8.7685,-8.6457500000000001,-8.509,-8.36325,-8.2030000000000001,-8.↵
03525,-7.85975,-7.67675,-7.4852500000000001,-7.2852500000000004,-7.0762500000000001,-6.855,-6.↵
```

6229999999999999,-6.37675,-6.1164999999999999,-5.8395,-5.5465,-5.23675,-4.9102500000000004,-4.↵
5707499999999999,-4.2205,-3.859,-3.48975,-3.1165000000000003,-2.7405,-2.36675,-1.9975,-1.↵
63575,-1.2825,-0.94075,-0.6142500000000001,-0.303,-0.006,0.27,0.53025,0.77275,0.9957499999999999,1.↵
20175,1.39075,1.5610000000000002,1.7175,1.8585,1.98525,2.10425,2.2102500000000003,2.31175,2.↵
407,2.494,2.58175,2.665,2.74575,2.8245,2.9045,2.98075,3.06025,3.13725,3.2135,3.28975,3.↵
36675,3.44625,3.524,3.605,3.68725,3.7765,3.86425,3.9605,4.058,4.1610000000000005,4.267,4.↵
37775,4.49,4.60425,4.7195,4.8355000000000001,4.94925,5.06275,5.175,5.285,5.3925,5.49775,5.↵
5962499999999995,5.6922500000000005,5.783,5.86925,5.9495,6.02275,6.0882499999999995,6.1477499999999999,6.↵
1965,6.23775,6.2689999999999999,6.289,6.30025,6.3009999999999999,6.2905000000000001,6.2704999999999999,6.↵
2385,6.1965,6.14325,6.08075,6.012,5.9334999999999996,5.84875,5.75725,5.6594999999999995,5.↵
558,5.45275,5.3405000000000005,5.2245,5.104,4.97975,4.8500000000000005,4.71575,4.57925,4.↵
4365000000000006,4.2915,4.1427499999999995,3.99025,3.83825,3.685,3.53625,3.3875,3.24325,3.↵
10375,2.9732499999999997,2.848,2.73125,2.6252500000000003,2.53,2.4445,2.3682499999999997,2.↵
2995,2.2355,2.179,2.1277500000000003,2.0774999999999997,2.031,1.986,1.9377499999999999,1.↵
88975,1.8365,1.7822500000000001,1.72575,1.6632500000000001,1.6007500000000001,1.532,1.46175,1.↵
394,1.323,1.25725,1.194,1.13825,1.0902500000000002,1.052,1.02075,0.99875,0.98725,0.9804999999999999,0.↵
98125,0.9894999999999999,0.99875,1.01175,1.027,1.04075,1.05275,1.05975,1.065,1.068,1.06275,1.↵
0545,1.0415,1.0255,1.01025,0.99025,0.9712500000000001,0.9514999999999999,0.9337500000000001,0.↵
9169999999999999,0.9032500000000001,0.8902500000000001,0.882,0.8765000000000001,0.87125,0.↵
86975,0.869,0.87125,0.872,0.8734999999999999,0.8750000000000001,0.8750000000000001,0.87425,0.↵
8704999999999999,0.8659999999999999,0.856,0.84375,0.827,0.80725,0.782,0.753,0.72025,0.6842499999999999,0.↵
6455,0.6042500000000001,0.5615,0.518,0.4752499999999995,0.434,0.396,0.364,0.331,0.303,0.↵
27925,0.2602500000000004,0.2435000000000002,0.232,0.2205,0.2105000000000002,0.2,0.18775,0.↵
17625,0.16325,0.1465,0.1327499999999998,0.1167499999999999,0.10375,0.0915,0.08175,0.077,0.↵
07625,0.0785,0.0885000000000001,0.1015,0.1175000000000001,0.1364999999999998,0.1565,0.↵
1785,0.1982499999999998,0.21525,0.2279999999999998,0.2380000000000002,0.2425,0.24175,0.↵
23575,0.2220000000000003,0.2045,0.17925,0.148,0.1129999999999999,0.07325,0.0282499999999997,-0.↵
01975,-0.0710000000000001,-0.1227500000000001,-0.17625,-0.2279999999999998,-0.2769999999999997,-0.↵
32275,-0.36175,-0.396,-0.4195,-0.43175,-0.4295,-0.4135,-0.3807500000000003,-0.3334999999999996,-0.↵
2715,-0.19225,-0.0985,0.01075,0.1342499999999998,0.267,0.40825,0.55325,0.70275,0.85375,1.↵
001,1.142,1.27875,1.4045,1.52125,1.6265,1.7197500000000001,1.8005,1.86775,1.915,1.95,1.↵
97075,1.9775,1.97225,1.95925,1.9377499999999999,1.908,1.87675,1.84175,1.806,1.77075,1.7395,1.↵
71125,1.683,1.657,1.6327500000000001,1.60975,1.58375,1.5565,1.52275,1.48325,1.43575,1.381,1.↵
31525,1.24125,1.1575,1.06275,0.95975,0.8467499999999999,0.7284999999999999,0.612,0.4845000000000004,0.↵
33875,0.1885,0.03499999999999996,-0.129,-0.3005,-0.48075,-0.666,-0.8574999999999999,-1.↵
0545,-1.2565,-1.464,-1.6715,-1.8775,-2.0805000000000002,-2.27825,-2.4665,-2.64675,-2.81225,-2.↵
964,-3.099,-3.2165,-3.3165,-3.39725,-3.463,-3.511,-3.5437499999999997,-3.5645000000000002,-3.↵
57425,-3.57125,-3.563,-3.5492500000000002,-3.5325,-3.51175,-3.49425,-3.47675,-3.46375,-3.↵
45525,-3.456,-3.46525,-3.48575,-3.5195,-3.56525,-3.62925,-3.7095000000000002,-3.8055,-3.↵
9215,-4.0535000000000005,-4.2044999999999995,-4.36925,-4.5532499999999999,-4.75,-4.9597500000000005,-5.↵
18425,-5.416,-5.6564999999999999,-5.90525,-6.15475,-6.4094999999999995,-6.6635,-6.916,-7.↵
16175,-7.402,-7.6332499999999999,-7.85,-8.0560000000000001,-8.2457500000000001,-8.4197499999999999,-8.↵
57325,-8.7052500000000001,-8.81725,-8.91025,-8.9835,-9.0392500000000001,-9.07675,-9.0972499999999999,-9.↵
0995,-9.089,-9.069,-9.034,-8.99125,-8.93475,-8.8700000000000001,-8.7975,-8.712,-8.615,-8.↵
5075,-8.3877499999999999,-8.25575,-8.1085000000000001,-7.94675,-7.76975,-7.5767500000000005,-7.↵
3662500000000001,-7.141,-6.8977499999999999,-6.6345,-6.3575000000000001,-6.0615000000000006,-5.↵
7494999999999999,-5.42375,-5.08425,-4.73475,-4.3747500000000001,-4.0085000000000001,-3.63775,-3.↵
2685,-2.89775,-2.53525,-2.179,-1.83175,-1.4985,-1.168,-0.8652500000000001,-0.5805,-0.2860000000000003,-0.↵
0115,0.22975,0.4462500000000004,0.63775,0.8162499999999999,0.99025,1.16425,1.33525,1.49775,1.↵
6495,1.78975,1.92025,2.04,2.153,2.2575000000000003,2.3560000000000003,2.4467499999999998,2.↵
5345,2.62,2.7015000000000002,2.78325,2.864,2.945,3.02575,3.10825,3.1905,3.2775,3.36675,3.↵
45775,3.55225,3.65075,3.7504999999999997,3.85425,3.96125,4.0695000000000001,4.18175,4.2944999999999999,4.↵
40825,4.5234999999999999,4.6395,4.7585000000000001,4.873,4.99125,5.1069999999999999,5.22225,5.↵
33375,5.4465,5.55275,5.6572499999999994,5.7547500000000005,5.8457500000000001,5.9295,6.005,6.↵
0714999999999995,6.12875,6.17525,6.2125,6.2399999999999999,6.2585,6.26825,6.26825,6.2600000000000001,6.↵
2440000000000001,6.2195,6.18825,6.1492499999999999,6.1019999999999999,6.0447500000000005,5.↵
9830000000000005,5.91275,5.8365,5.75325,5.66025,5.5595,5.4565,5.34375,5.22225,5.0935,4.↵
9575000000000005,4.8157499999999995,4.66925,4.519,4.3625,4.20675,4.052,3.89775,3.746,3.↵

```
6025, 3.466, 3.3369999999999997, 3.2195, 3.1105, 3.01275, 2.9242500000000002, 2.8457500000000002, 2.↵
7725, 2.707, 2.64425, 2.5825, 2.5229999999999997, 2.4604999999999997, 2.39575, 2.32775, 2.2560000000000002, 2.↵
1774999999999998, 2.0972500000000003, 2.011, 1.925, 1.8339999999999999, 1.74875, 1.6625, 1.58475, 1.↵
5075, 1.43975, 1.38025, 1.3259999999999998, 1.2817500000000002, 1.2449999999999999, 1.217, 1.194, 1.↵
17875, 1.1672500000000001, 1.15825, 1.1535, 1.14825, 1.143, 1.139, 1.13225, 1.126, 1.117, 1.1055000000000001, 1.↵
09325, 1.07875, 1.065, 1.052, 1.0375, 1.0215, 1.0055, 0.985, 0.96825, 0.9497500000000001, 0.9337500000000001, 0.↵
9192499999999999, 0.9087500000000001, 0.89725, 0.8920000000000001, 0.8865, 0.88725, 0.8880000000000001, 0.↵
8927499999999999, 0.89725, 0.9017499999999999, 0.9087500000000001, 0.91025, 0.9095000000000001, 0.↵
9047499999999999, 0.8995, 0.885, 0.869, 0.8484999999999999, 0.82325, 0.792, 0.7575000000000001, 0.↵
72025, 0.6835, 0.64625, 0.6094999999999999, 0.5745, 0.53875, 0.505, 0.4745000000000003, 0.4462500000000004, 0.↵
4165000000000004, 0.3905, 0.36324999999999996, 0.33425, 0.3067499999999997, 0.2769999999999997, 0.↵
25025, 0.2197499999999997, 0.19, 0.161, 0.12975, 0.10225, 0.07475, 0.0487499999999995, 0.0275, 0.↵
0085, -0.00925, -0.0212499999999998, -0.029, -0.0349999999999996, -0.0349999999999996, -0.↵
0342499999999996, -0.02675, -0.01825, -0.004500000000000005, 0.0085, 0.026, 0.045, 0.0655, 0.↵
08775000000000001, 0.10824999999999999, 0.129, 0.14575, 0.15875, 0.1655, 0.167, 0.15875, 0.14275, 0.↵
116, 0.0825, 0.039, -0.013, -0.0725, -0.13725, -0.20600000000000002, -0.27625, -0.348, -0.415, -0.↵
47825, -0.53625, -0.58375, -0.62325, -0.6515, -0.666, -0.6697500000000001, -0.6577500000000001, -0.↵
63025, -0.58975, -0.53025, -0.45399999999999996, -0.36475, -0.26175, -0.14425, -0.016, 0.122, 0.↵
2654999999999996, 0.41275, 0.56075, 0.7065, 0.8484999999999999, 0.9795, 1.10175, 1.21525, 1.316, 1.↵
40225, 1.4755, 1.538, 1.587, 1.6265, 1.657, 1.68, 1.69825, 1.71275, 1.722, 1.73175, 1.73875, 1.744, 1.↵
74875, 1.75325, 1.75475, 1.7525, 1.7495, 1.74325, 1.73275, 1.71825, 1.70125, 1.68, 1.65475, 1.62575, 1.↵
59375, 1.55475, 1.513, 1.464, 1.41075, 1.3535, 1.28475, 1.2085000000000001, 1.12225, 1.0255, 0.9155, 0.↵
7965, 0.66225, 0.515, 0.35400000000000004, 0.18475, 0.00375, -0.187, -0.3845, -0.5904999999999999, -0.↵
7979999999999999, -1.0085, -1.22, -1.4297499999999999, -1.6335, -1.82875, -2.0172499999999998, -2.↵
19875, -2.365, -2.5185, -2.65875, -2.78325, -2.8930000000000002, -2.98625, -3.06025, -3.1205, -3.↵
1655, -3.19675, -3.2165, -3.2235, -3.222, -3.21425, -3.20125, -3.18375, -3.1685, -3.15025, -3.1350000000000002, -3.↵
12125, -3.1135, -3.1165000000000003, -3.1220000000000003, -3.1350000000000002, -3.164, -3.215, -3.↵
273, -3.3425000000000002, -3.4255, -3.5292499999999998, -3.6515, -3.79475, -3.9565, -4.136, -4.↵
33125, -4.5425, -4.766, -5.00325, -5.2497500000000001, -5.49925, -5.7555000000000005, -6.0104999999999995, -6.↵
2615000000000001, -6.50625, -6.7434999999999999, -6.97325, -7.19225, -7.399, -7.5905000000000005, -7.↵
7689999999999999, -7.9307500000000001, -8.0795000000000001, -8.2167500000000001, -8.33675, -8.442, -8.↵
53275, -8.61125, -8.6762500000000001, -8.72875, -8.7692500000000001, -8.8005, -8.8195, -8.831, -8.↵
83175, -8.82175, -8.80275, -8.77225, -8.731, -8.677, -8.60975, -8.52575, -8.426, -8.3099999999999999, -8.↵
1742499999999999, -8.0185000000000001, -7.843, -7.6454999999999999, -7.4287500000000001, -7.1915000000000004, -6.↵
935, -6.662, -6.3705, -6.0645, -5.74575, -5.4177500000000001, -5.0805000000000001, -4.73625, -4.3892500000000005, -4.↵
03750000000000005, -3.68725, -3.3385, -2.9975, -2.65875, -2.3307499999999997, -2.00875, -1.69675, -1.↵
3955, -1.1039999999999999, -0.82475, -0.55775, -0.3037499999999996, -0.06325, 0.16325, 0.3785, 0.↵
57825, 0.76825, 0.9475000000000001, 1.11225, 1.268, 1.413, 1.548, 1.67925, 1.80125, 1.915, 2.02725, 2.↵
1325, 2.23225, 2.3314999999999997, 2.4245, 2.51625, 2.6054999999999997, 2.69175, 2.77775, 2.861, 2.↵
9457500000000003, 3.0305, 3.1157500000000002, 3.205, 3.2960000000000003, 3.38975, 3.4875000000000003, 3.↵
58975, 3.6965, 3.80625, 3.92, 4.036, 4.1557499999999999, 4.27325, 4.3900000000000001, 4.51825, 4.6425, 4.↵
74475, 4.8515000000000001, 4.97525, 5.108, 5.23975, 5.365, 5.4787500000000001, 5.57475, 5.6594999999999995, 5.↵
7342499999999999, 5.803, 5.86625, 5.92425, 5.97225, 6.01725, 6.0539999999999999, 6.0859999999999999, 6.↵
11275, 6.1317499999999999, 6.144, 6.15, 6.1484999999999999, 6.1425, 6.125, 6.1004999999999999, 6.0685, 6.↵
0257499999999995, 5.97225, 5.9082500000000001, 5.8295, 5.7410000000000005, 5.63975, 5.526, 5.40225, 5.↵
2695, 5.1270000000000001, 4.9805, 4.8287499999999999, 4.6774999999999999, 4.5234999999999999, 4.3747500000000001, 4.↵
22975, 4.09, 3.95825, 3.8337499999999998, 3.71775, 3.61175, 3.5125, 3.4195, 3.3325, 3.2485, 3.16775, 3.↵
0885, 3.0082500000000003, 2.9267499999999997, 2.8434999999999997, 2.75875, 2.6710000000000003, 2.↵
5795, 2.48425, 2.3895, 2.2950000000000004, 2.198, 2.10275, 2.01175, 1.921, 1.83575, 1.75475, 1.6785, 1.↵
60975, 1.5457500000000002, 1.4855, 1.4335, 1.38625, 1.348, 1.3145, 1.2855, 1.2642499999999999, 1.↵
2475, 1.2315, 1.2185, 1.2055, 1.194, 1.18025, 1.1657499999999998, 1.15275, 1.1352499999999999, 1.↵
114, 1.0947500000000001, 1.06975, 1.0467499999999998, 1.02375, 1.0025, 0.9835, 0.966, 0.9514999999999999, 0.↵
9400000000000001, 0.93075, 0.92775, 0.9285, 0.9322499999999999, 0.9377500000000001, 0.94525, 0.↵
9522499999999999, 0.9612499999999999, 0.96825, 0.972, 0.97425, 0.9735, 0.9690000000000001, 0.962, 0.↵
9530000000000001, 0.9415, 0.92925, 0.914, 0.898, 0.87975, 0.8630000000000001, 0.84375, 0.8240000000000001, 0.↵
8024999999999999, 0.7805000000000001, 0.7567499999999999, 0.73175, 0.70575, 0.6775, 0.6477499999999999, 0.↵
6142500000000001, 0.5845, 0.5507500000000001, 0.521, 0.48900000000000005, 0.45775, 0.4287499999999996, 0.↵
3989999999999997, 0.36625, 0.335, 0.302, 0.27225, 0.241, 0.20975, 0.18, 0.14725000000000002, 0.↵
```

1167499999999999,0.08925,0.068,0.04874999999999995,0.03424999999999996,0.026,0.02375,0.↵
0275,0.039,0.05575000000000001,0.07925,0.106,0.13274999999999998,0.1625,0.193,0.22425,0.↵
24949999999999997,0.27475,0.29524999999999996,0.309,0.31575000000000003,0.31275000000000003,0.↵
30525,0.28775,0.2595,0.2205,0.174,0.11674999999999999,0.05275,-0.019,-0.097,-0.17700000000000002,-0.↵
260250000000000004,-0.34325,-0.42050000000000004,-0.49525,-0.55925,-0.611,-0.6515,-0.67525,-0.↵
68275,-0.67225,-0.64250000000000001,-0.599,-0.53775000000000001,-0.45999999999999996,-0.37074999999999997,-0.↵
26849999999999996,-0.15875,-0.04274999999999996,0.07775,0.20075,0.322,0.44400000000000006,0.↵
5599999999999999,0.6745,0.78425,0.88800000000000001,0.9865,1.07875,1.1635,1.2435,1.31675,1.↵
38324999999999999,1.44425,1.497,1.545,1.58625000000000002,1.62125,1.65025,1.67400000000000002,1.↵
6915,1.7075,1.7175,1.728,1.735,1.7395,1.74325,1.748,1.75175,1.7555,1.75775,1.75925,1.75925,1.↵
7555,1.748,1.73575000000000001,1.719,1.6915,1.6595,1.61825,1.56475,1.503,1.429,1.3419999999999999,1.↵
24275,1.13,1.004,0.8619999999999999,0.70800000000000001,0.5395,0.36075,0.17099999999999999,-0.↵
0275,-0.23575,-0.44475,-0.66075,-0.87425,-1.08725,-1.29475,-1.497,-1.69075,-1.87675,-2.↵
0515,-2.214,-2.3635,-2.50025,-2.62375000000000002,-2.73350000000000003,-2.8289999999999997,-2.↵
91149999999999998,-2.98075,-3.0335,-3.07475,-3.1045,-3.12275,-3.13175,-3.1295,-3.12125,-3.↵
10675000000000003,-3.08675,-3.0655,-3.0465,-3.02725,-3.016,-3.0105,-3.01425,-3.03125,-3.↵
05775,-3.10050000000000003,-3.16074999999999997,-3.23575,-3.331,-3.44475,-3.57525,-3.72550000000000003,-3.↵
89175,-4.074,-4.27175,-4.48150000000000005,-4.705,-4.9369999999999999,-5.175,-5.4185,-5.66250000000000005,-5.↵
90274999999999999,-6.13925000000000005,-6.3705,-6.5939999999999999,-6.8115,-7.0167499999999999,-7.↵
21200000000000001,-7.3945,-7.56375,-7.7232499999999999,-7.87275000000000001,-8.01075,-8.1390000000000001,-8.↵
25875,-8.371,-8.47475,-8.56925,-8.6555,-8.7365,-8.80725,-8.87075,-8.924,-8.9690000000000001,-9.↵
002,-9.0209999999999999,-9.02700000000000001,-9.0164999999999999,-8.9782500000000001,-8.9294999999999999,-8.↵
8669999999999999,-8.78375,-8.68,-8.554,-8.40825,-8.2442500000000001,-8.0582500000000001,-7.↵
85375,-7.631,-7.39125,-7.1335,-6.862,-6.5742499999999999,-6.2745,-5.96225,-5.6434999999999995,-5.↵
317,-4.9834999999999999,-4.6455,-4.30525,-3.9665,-3.63,-3.29600000000000003,-2.9685,-2.6475,-2.↵
33450000000000002,-2.03174999999999997,-1.73875,-1.45425,-1.1825,-0.92075000000000001,-0.6715,-0.↵
43325,-0.20525000000000002,0.01225,0.21974999999999997,0.41650000000000004,0.60425000000000001,0.↵
78050000000000001,0.94750000000000001,1.10550000000000001,1.25575,1.3992499999999999,1.5334999999999999,1.↵
6595,1.78075,1.8935,2.00350000000000003,2.1065,2.2025,2.29875,2.39175,2.481,2.571,2.66125,2.↵
752,2.84499999999999998,2.94274999999999998,3.041,3.14475,3.251,3.36375,3.4835,3.60575,3.↵
73225,3.86275,3.994,4.1282499999999995,4.2625,4.396,4.52800000000000005,4.65475,4.7775,4.↵
89575000000000005,5.00875,5.1185,5.2215,5.3199999999999999,5.4115,5.50075,5.584,5.6655,5.↵
7435,5.82275,5.899,5.9700000000000001,6.0387499999999999,6.10425,6.16525,6.2195,6.2659999999999999,6.↵
30200000000000005,6.32625,6.3377500000000001,6.337,6.3210000000000001,6.2865,6.2415,6.176,6.↵
0975,6.0104999999999995,5.9082500000000001,5.7974999999999999,5.68325,5.56025,5.43525,5.30775,5.↵
18025,5.05225,4.92700000000000005,4.8050000000000001,4.69425,4.57925,4.46700000000000005,4.↵
38,4.2825000000000001,4.17700000000000005,4.0665,3.9505,3.83675,3.73375,3.6415,3.556,3.46975,3.↵
38275,3.292,3.19675,3.09674999999999997,2.9945,2.88999999999999997,2.784,2.67575,2.5695,2.↵
465,2.365,2.2675,2.176,2.08975,2.011,1.94025,1.8752499999999999,1.818,1.76925,1.7235,1.↵
68525,1.651,1.6175,1.5855000000000001,1.55575,1.525,1.49225,1.45875,1.42375,1.3885,1.3535,1.↵
32,1.284,1.252,1.2215,1.19475,1.171,1.14975,1.13375,1.12075000000000001,1.11,1.10175,1.09625,1.↵
091,1.08725,1.0825,1.08175,1.07875,1.078,1.07575,1.07725,1.07575,1.0765,1.07125,1.068,1.↵
06350000000000001,1.05749999999999999,1.0475,1.036,1.02375,1.00775,0.9924999999999999,0.97425,0.↵
9545,0.93524999999999999,0.91325,0.8942500000000001,0.87425,0.8545,0.83225,0.814,0.792,0.↵
769,0.74225000000000001,0.714,0.6835,0.64925,0.6142500000000001,0.57375000000000001,0.534,0.↵
490499999999999994,0.44775,0.40275,0.35775,0.31425,0.27075,0.22975,0.19075,0.15175,0.11825,0.↵
08925,0.061,0.042,0.026,0.01375,0.01075,0.0115,0.0205,0.0365,0.05724999999999995,0.084,0.↵
11750000000000001,0.151,0.1915,0.23124999999999998,0.27075,0.306,0.33575,0.357,0.37,0.37074999999999997,0.↵
3585,0.33349999999999996,0.29524999999999996,0.24175,0.17775,0.10225,0.01975,-0.07025,-0.↵
1655,-0.25775,-0.35025,-0.4365,-0.51425,-0.5805,-0.63475,-0.6775,-0.71025,-0.72775,-0.73625,-0.↵
72775,-0.70724999999999999,-0.673,-0.6295000000000001,-0.573,-0.508,-0.43175,-0.348,-0.257,-0.↵
15875,-0.058,0.04725,0.154,0.26175,0.36925,0.4775,0.582,0.6842499999999999,0.78275,0.8750000000000001,0.↵
962,1.043,1.11625,1.184,1.24425,1.30075,1.34975,1.39325,1.4312500000000001,1.46625,1.497,1.↵
52275,1.5465,1.5685,1.59225,1.6135,1.6365,1.66175,1.68675,1.7149999999999999,1.744,1.77075,1.↵
7975,1.82275,1.84475,1.8615,1.87075,1.87225,1.867,1.8455,1.81575,1.7722499999999999,1.71275,1.↵
6395,1.55325,1.45025,1.336,1.20625,1.06350000000000001,0.90950000000000001,0.7437499999999999,0.↵
57000000000000001,0.38525,0.19225,-0.004500000000000005,-0.20600000000000002,-0.41125,-0.↵
61425000000000001,-0.81949999999999999,-1.0215,-1.21925,-1.41225,-1.59675,-1.77075,-1.93625,-2.↵

089,-2.23,-2.3575,-2.46975,-2.5687499999999996,-2.6505,-2.71825,-2.77175,-2.8115,-2.83725,-2.↵
85025,-2.85275,-2.8465000000000003,-2.8320000000000003,-2.81,-2.787,-2.76325,-2.7397500000000004,-2.↵
719,-2.703,-2.69775,-2.7022500000000003,-2.7199999999999998,-2.74975,-2.7977499999999997,-2.↵
8632500000000003,-2.9457500000000003,-3.0465,-3.1655,-3.302,-3.457,-3.627,-3.81,-4.00475,-4.↵
2115,-4.425,-4.6425,-4.8629999999999995,-5.085,-5.303999999999999,-5.5215,-5.734999999999999,-5.↵
944,-6.146999999999999,-6.3445,-6.5345,-6.7185,-6.897749999999999,-7.06875,-7.23725,-7.↵
402749999999999,-7.5592500000000005,-7.710999999999999,-7.85675,-7.99875,-8.13225,-8.26025,-8.↵
381,-8.49375,-8.5975,-8.691500000000001,-8.7745,-8.84475,-8.902000000000001,-8.94625,-8.↵
97675,-8.98825,-8.98525,-8.964500000000001,-8.923250000000001,-8.863,-8.78375,-8.68,-8.↵
561,-8.419749999999999,-8.258000000000001,-8.07725,-7.877249999999999,-7.6585,-7.425,-7.↵
1762500000000005,-6.91075,-6.632250000000001,-6.343749999999999,-6.0455000000000005,-5.↵
7395000000000005,-5.42675,-5.10875,-4.789750000000001,-4.4677500000000006,-4.1457500000000005,-3.↵
82625,-3.51025,-3.20125,-2.89925,-2.6025,-2.314,-2.031,-1.75925,-1.493,-1.238249999999999,-0.↵
993999999999999,-0.756749999999999,-0.528,-0.3112499999999997,-0.0992499999999999,0.↵
1,0.29225,0.4760000000000003,0.647749999999999,0.800249999999999,0.9475000000000001,1.↵
094,1.22975,1.35275,1.46625,1.57475,1.67475,1.772249999999999,1.8655000000000002,1.95475,2.↵
04625,2.13775,2.23075,2.32925,2.4345,2.546,2.6635,2.7847500000000003,2.9105,3.041,3.1745,3.↵
308,3.44,3.57125,3.69725,3.8185,3.936,4.052,4.1602500000000004,4.2695,4.37325,4.4784999999999995,4.↵
58375,4.68675,4.79275,4.901,5.0125,5.12325,5.236,5.34825,5.46025,5.570250000000001,5.6747499999999995,5.↵
77325,5.86325,5.9455,6.015750000000001,6.07375,6.1195,6.1530000000000005,6.17,6.17525,6.↵
164499999999999,6.1392500000000005,6.101999999999999,6.053999999999999,5.992999999999999,5.↵
9235000000000001,5.844,5.7540000000000004,5.66025,5.5595,5.45725,5.351249999999999,5.2444999999999995,5.↵
136,5.029999999999999,4.9255,4.8255,4.72575,4.63025,4.53575,4.444,4.354,4.26175,4.16725,4.↵
07175,3.978,3.87875,3.77575,3.6734999999999998,3.5645000000000002,3.45375,3.341,3.22725,3.↵
1135,3.0044999999999997,2.8945,2.79,2.68925,2.5909999999999997,2.50025,2.411,2.33,2.253,2.↵
17825,2.1065,2.04,1.976,1.918,1.8615,1.8082500000000001,1.7607500000000003,1.712,1.667,1.↵
625,1.58475,1.54425,1.50525,1.46625,1.429,1.387,1.34425,1.3145,1.27475,1.22525,1.1895,1.↵
172,1.165,1.1657499999999998,1.1672500000000001,1.1635,1.15425,1.14675,1.14125,1.13675,1.↵
1352499999999999,1.135249999999999,1.1360000000000001,1.1375,1.1360000000000001,1.1360000000000001,1.↵
13225,1.1275,1.12375,1.11625,1.1085,1.10025,1.09175,1.08425,1.078,1.07125,1.0635000000000001,1.↵
05825,1.052,1.04525,1.03675,1.0285,1.0177500000000002,1.0055,0.98875,0.9735,0.952249999999999,0.↵
9315,0.907999999999999,0.8805000000000001,0.85375,0.82325,0.788999999999999,0.756,0.7195,0.↵
6775,0.63475,0.5890000000000001,0.5425,0.492,0.4394999999999995,0.3889999999999996,0.↵
338,0.2885,0.24175,0.2022499999999999,0.1677499999999998,0.1420000000000002,0.125,0.↵
1190000000000001,0.1242499999999999,0.13575,0.161,0.19375,0.2334999999999999,0.2799999999999997,0.↵
328,0.37925,0.4287499999999996,0.47225,0.51125,0.54475,0.56525,0.575249999999999,0.5760000000000001,0.↵
5645,0.53875,0.502,0.45325,0.396,0.32725,0.25325,0.1732500000000002,0.08925,0.004500000000000005,-0.↵
08325,-0.167,-0.24725,-0.3212500000000004,-0.3905,-0.4502500000000004,-0.50125,-0.54175,-0.↵
5700000000000001,-0.588249999999999,-0.5935,-0.5875,-0.5715,-0.54325,-0.505,-0.45775,-0.↵
4004999999999997,-0.3334999999999996,-0.26175,-0.1815,-0.09625,-0.004500000000000005,0.↵
09225,0.19075,0.29,0.3915,0.4904999999999994,0.5890000000000001,0.6835,0.77275,0.857499999999999,0.↵
935249999999999,1.00625,1.072,1.13,1.1795,1.2245000000000001,1.265,1.30075,1.3390000000000002,1.↵
37175,1.406,1.44125,1.477,1.5174999999999998,1.5565,1.59825,1.6424999999999998,1.6875,1.↵
73725,1.78525,1.834749999999999,1.88225,1.9295,1.97,2.00275,2.0255,2.04325,2.0500000000000003,2.↵
0455,2.0295,1.999,1.95625,1.899,1.83025,1.7472499999999997,1.653249999999999,1.54025,1.↵
41675,1.2795,1.12925,0.96825,0.79575,0.612,0.422,0.2227499999999998,0.019,-0.18375,-0.↵
39225,-0.602,-0.806499999999999,-1.01175,-1.207749999999999,-1.397,-1.5785,-1.7495,-1.↵
90975,-2.0545,-2.18425,-2.3002499999999997,-2.4017500000000003,-2.48575,-2.55275,-2.60325,-2.↵
6382499999999998,-2.66025,-2.6687499999999997,-2.66575,-2.6527499999999997,-2.63225,-2.↵
60775,-2.5825,-2.5604999999999998,-2.53975,-2.5245,-2.5185,-2.5245,-2.5412500000000002,-2.↵
5735,-2.6214999999999997,-2.6832499999999997,-2.76425,-2.8587499999999997,-2.96925,-3.0945,-3.↵
23175,-3.38375,-3.54625,-3.7155,-3.8962499999999998,-4.0817499999999995,-4.2725,-4.469250000000001,-4.↵
66775,-4.8660000000000005,-5.065250000000001,-5.2627500000000005,-5.459499999999999,-5.↵
655749999999999,-5.85025,-6.038749999999999,-6.22625,-6.4102500000000004,-6.59175,-6.771000000000001,-6.↵
94725,-7.120500000000001,-7.292999999999999,-7.462249999999999,-7.627000000000001,-7.788,-7.↵
94525,-8.0985,-8.2405,-8.37625,-8.5045,-8.62275,-8.7295,-8.8235,-8.90425,-8.973,-9.0225,-9.↵
057500000000001,-9.07375,-9.06975,-9.05075,-9.010250000000001,-8.9515,-8.871500000000001,-8.↵
773,-8.65175,-8.51375,-8.35425,-8.175,-7.97875,-7.76525,-7.535500000000001,-7.289,-7.031999999999999,-6.↵

76125,-6.479749999999999,-6.192,-5.89825,-5.59925,-5.297000000000001,-4.995,-4.6905,-4.↵
3885,-4.087,-3.788,-3.489,-3.1915,-2.89775,-2.608500000000003,-2.3225,-2.04175,-1.76625,-1.↵
4985,-1.2405,-0.992499999999999,-0.756749999999999,-0.5325,-0.3235,-0.126,0.0587500000000004,0.↵
2279999999999998,0.38525,0.534,0.6705,0.800249999999999,0.924,1.04225,1.159,1.274,1.38925,1.↵
506,1.625,1.746249999999999,1.87,1.994999999999999,2.12175,2.25,2.377249999999996,2.↵
50625,2.63225,2.75875,2.87925,3.0,3.118249999999997,3.231,3.3455,3.456,3.56825,3.678250000000002,3.↵
7895,3.9025,4.016999999999995,4.132000000000001,4.253500000000001,4.376250000000001,4.↵
50125,4.630999999999999,4.7615,4.891249999999999,5.02325,5.1545,5.28325,5.40775,5.53125,5.↵
657999999999995,5.7685,5.86475,5.951,6.026499999999995,6.087499999999995,6.133999999999995,6.↵
164499999999999,6.178999999999999,6.1805,6.1675,6.14175,6.101999999999999,6.054749999999999,6.↵
001250000000001,5.94175,5.876250000000001,5.806,5.734999999999999,5.662500000000005,5.↵
58925,5.51525,5.43975,5.362,5.28325,5.20025,5.1185,5.029249999999999,4.94075,4.84775,4.↵
75475,4.657,4.557,4.454000000000001,4.34875,4.24275,4.1335,4.02525,3.9155,3.80475,3.694250000000003,3.↵
582,3.472250000000003,3.36225,3.2555,3.148,3.045,2.945,2.848,2.7535,2.66425,2.578000000000003,2.↵
495500000000003,2.41925,2.343,2.27425,2.205,2.139999999999997,2.07525,2.015,1.953999999999997,1.↵
89275,1.833999999999999,1.777000000000001,1.720500000000001,1.664,1.611249999999998,1.↵
559499999999999,1.5115,1.46875,1.42825,1.39075,1.358750000000001,1.326749999999999,1.↵
29925,1.27725,1.2565,1.238249999999999,1.222999999999999,1.208500000000001,1.196249999999998,1.↵
18325,1.172,1.162,1.152000000000001,1.14375,1.135249999999999,1.1285,1.1215,1.11625,1.↵
11,1.10625,1.098750000000001,1.09175,1.085,1.075,1.06575,1.059,1.049,1.03675,1.0255,1.↵
0155,1.00775,1.001,0.993250000000001,0.98875,0.982749999999999,0.97725,0.971250000000001,0.↵
962750000000001,0.953000000000001,0.937750000000001,0.922500000000001,0.904749999999999,0.↵
869,0.834750000000001,0.8095,0.766749999999999,0.705,0.63175,0.5455,0.4615,0.3845,0.31975,0.↵
26175,0.21125,0.16325,0.1235,0.09,0.06325,0.04725,0.0404999999999994,0.04125,0.055750000000001,0.↵
07625,0.105250000000001,0.14275,0.18475,0.2305,0.276999999999997,0.325,0.370749999999997,0.↵
4097499999999995,0.44325,0.468499999999997,0.483,0.489000000000005,0.48525,0.470000000000003,0.↵
441,0.402,0.354000000000004,0.297500000000004,0.231249999999998,0.16025,0.084,0.00375,-0.↵
07775,-0.15875,-0.2395,-0.3175,-0.3915,-0.46225,-0.525,-0.582,-0.631,-0.6705,-0.702000000000001,-0.↵
72175,-0.73325,-0.73325,-0.72175,-0.695,-0.65925,-0.614250000000001,-0.55775,-0.492749999999997,-0.↵
41875,-0.33725,-0.25325,-0.1625,-0.07025,0.02225,0.114499999999999,0.202249999999999,0.↵
29,0.37225,0.451,0.52575,0.5935,0.6585,0.71725,0.7735,0.8255,0.876500000000001,0.922500000000001,0.↵
96975,1.017750000000002,1.063500000000001,1.11475,1.16425,1.21625,1.2725,1.33125,1.3915,1.↵
45275,1.51975,1.587,1.65475,1.72275,1.786,1.84775,1.908,1.9615,2.008,2.0455,2.07525,2.09275,2.↵
10125,2.095750000000002,2.078249999999997,2.048499999999998,2.00575,1.95,1.879,1.796,1.↵
6975,1.58775,1.46325,1.325999999999998,1.17425,1.01325,0.837000000000001,0.6515,0.45775,0.↵
25775,0.05025,-0.161,-0.3715,-0.58125,-0.786500000000001,-0.9865,-1.17875,-1.358,-1.52425,-1.↵
677,-1.815,-1.9355,-2.03925,-2.1255,-2.192749999999998,-2.243,-2.27975,-2.3025,-2.3125,-2.↵
314,-2.308,-2.29575,-2.27825,-2.259,-2.241500000000003,-2.227000000000003,-2.21875,-2.↵
2155,-2.223249999999997,-2.238499999999997,-2.2675,-2.30725,-2.36275,-2.429999999999997,-2.↵
510000000000002,-2.60075,-2.7055,-2.819,-2.945750000000003,-3.083,-3.22875,-3.38125,-3.↵
5415,-3.70625,-3.87575,-4.048249999999995,-4.22525,-4.405250000000006,-4.585249999999999,-4.↵
7675,-4.952999999999999,-5.13675,-5.319999999999999,-5.50775,-5.693750000000005,-5.88075,-6.↵
06775,-6.253,-6.43925,-6.622999999999999,-6.806249999999995,-6.987,-7.164750000000001,-7.↵
340249999999999,-7.512000000000005,-7.67825,-7.842250000000001,-7.99725,-8.148249999999999,-8.↵
28925,-8.419749999999999,-8.538,-8.645750000000001,-8.7365,-8.812000000000001,-8.871500000000001,-8.↵
91025,-8.932500000000001,-8.931000000000001,-8.90875,-8.867750000000001,-8.805,-8.720500000000001,-8.↵
61975,-8.497499999999999,-8.360249999999999,-8.207,-8.03675,-7.854500000000001,-7.656000000000001,-7.↵
44775,-7.228,-6.99775,-6.754250000000001,-6.502499999999995,-6.239999999999999,-5.967750000000006,-5.↵
688499999999994,-5.4,-5.10475,-4.804250000000001,-4.499750000000001,-4.192250000000005,-3.↵
8825,-3.57675,-3.27375,-2.977000000000003,-2.687000000000003,-2.4085,-2.139249999999997,-1.↵
883,-1.63725,-1.403,-1.18475,-0.97575,-0.776,-0.58975,-0.409749999999995,-0.238000000000002,-0.↵
074,0.085499999999999,0.23875,0.38825,0.535500000000001,0.679,0.820249999999999,0.95825,1.↵
09625,1.23225,1.36575,1.497,1.627249999999999,1.7555,1.88375,2.00875,2.12925,2.25,2.368999999999998,2.↵
48175,2.593250000000003,2.70075,2.8075,2.911499999999998,3.01425,3.118249999999997,3.↵
22275,3.331,3.4415,3.56,3.6835,3.81475,3.949,4.091,4.23725,4.386249999999995,4.539499999999999,4.↵
692749999999999,4.84325,4.9905,5.1315,5.262000000000005,5.387000000000005,5.49925,5.59925,5.↵
688499999999994,5.763999999999999,5.82425,5.877750000000001,5.9165,5.9425,5.95775,5.964749999999996,5.↵
9615,5.951750000000005,5.933499999999996,5.908250000000001,5.87925,5.845000000000001,5.↵

806, 5.761, 5.713, 5.6617500000000005, 5.606, 5.548, 5.48475, 5.41925, 5.351249999999999, 5.2787500000000005, 5.↵
2025000000000001, 5.124, 5.03925, 4.952249999999999, 4.8607499999999995, 4.7645, 4.6645, 4.5615000000000006, 4.↵
45625, 4.3495, 4.239, 4.1290000000000004, 4.0184999999999995, 3.90925, 3.8025, 3.69875, 3.58875, 3.↵
4805, 3.38125, 3.286, 3.19225, 3.0997500000000002, 3.0075000000000003, 2.919, 2.8320000000000003, 2.↵
74425, 2.658, 2.57425, 2.49325, 2.40925, 2.32925, 2.2537499999999997, 2.1805000000000003, 2.1125000000000003, 2.↵
044, 1.9805, 1.9217499999999998, 1.863, 1.80675, 1.75475, 1.702, 1.6525, 1.60675, 1.5610000000000002, 1.↵
51675, 1.474, 1.43575, 1.4007500000000002, 1.36875, 1.3405, 1.31675, 1.29775, 1.2825, 1.2710000000000001, 1.↵
26125, 1.25425, 1.24825, 1.242, 1.23675, 1.22975, 1.21925, 1.207, 1.194, 1.1795, 1.1635, 1.14825, 1.↵
133, 1.1185, 1.1055000000000001, 1.09625, 1.0895, 1.085, 1.0865, 1.088749999999999, 1.0955, 1.1055000000000001, 1.↵
11625, 1.126, 1.1375, 1.146, 1.1520000000000001, 1.1535, 1.14975, 1.13975, 1.12375, 1.0995, 1.06725, 1.↵
0285, 0.9780000000000001, 0.923249999999999, 0.861999999999999, 0.795, 0.724, 0.64925, 0.5737500000000001, 0.↵
50275, 0.43325, 0.3685, 0.312, 0.2625, 0.2227499999999998, 0.19075, 0.16925, 0.16025, 0.161, 0.170999999999999, 0.↵
18925, 0.2145000000000002, 0.24725, 0.28225, 0.3205, 0.36075, 0.4004999999999997, 0.4394999999999995, 0.↵
4752499999999995, 0.50425, 0.53025, 0.54925, 0.56075, 0.56775, 0.5645, 0.549999999999999, 0.53025, 0.↵
49975, 0.4599999999999996, 0.41275, 0.35475, 0.2915, 0.21825, 0.1342499999999998, 0.05025, -0.↵
0275, -0.12125, -0.2249999999999998, -0.31425, -0.3875, -0.4462500000000004, -0.496, -0.54475, -0.↵
5875, -0.62625, -0.65625, -0.679, -0.6890000000000001, -0.6875, -0.67225, -0.647, -0.612, -0.56775, -0.↵
51275, -0.4547500000000004, -0.38825, -0.31975, -0.2465000000000002, -0.1732500000000002, -0.↵
097, -0.02375, 0.05025, 0.12125, 0.1915, 0.2562500000000003, 0.3205, 0.3785, 0.4349999999999994, 0.↵
4875, 0.536999999999999, 0.58675, 0.634, 0.68275, 0.73025, 0.77825, 0.83, 0.88425, 0.9415, 1.0047499999999998, 1.↵
06975, 1.1375, 1.2085000000000001, 1.281, 1.35575, 1.4335, 1.509, 1.58375, 1.66025, 1.7302499999999998, 1.↵
79975, 1.86375, 1.92325, 1.97975, 2.028, 2.06825, 2.10025, 2.12175, 2.13325, 2.1325, 2.11725, 2.08975, 2.↵
0484999999999998, 1.98975, 1.91725, 1.8265, 1.719, 1.5975, 1.458, 1.30225, 1.1344999999999998, 0.↵
951499999999999, 0.760749999999999, 0.557, 0.34875, 0.1364999999999998, -0.07625, -0.29075, -0.↵
49975, -0.7035, -0.89875, -1.081, -1.2534999999999998, -1.4115, -1.5525, -1.67775, -1.78825, -1.↵
8807500000000001, -1.9585000000000001, -2.018, -2.0645000000000002, -2.0972500000000003, -2.↵
11725, -2.1302499999999998, -2.1340000000000003, -2.1325, -2.1285, -2.12325, -2.11725, -2.11575, -2.↵
11575, -2.12175, -2.1340000000000003, -2.15375, -2.18125, -2.21875, -2.26375, -2.32075, -2.3872500000000003, -2.↵
4612499999999997, -2.5475000000000003, -2.642, -2.748, -2.861, -2.983, -3.1165000000000003, -3.↵
25325, -3.39725, -3.55075, -3.70875, -3.868, -4.03525, -4.202999999999999, -4.3725, -4.5440000000000005, -4.↵
71725, -4.891249999999999, -5.06675, -5.2437499999999995, -5.42525, -5.60675, -5.7915, -5.98075, -6.↵
16925, -6.3622499999999995, -6.55675, -6.7512500000000001, -6.9465, -7.140249999999999, -7.3342500000000001, -7.↵
523249999999999, -7.707999999999999, -7.8850000000000001, -8.05125, -8.207, -8.35125, -8.483, -8.↵
5975, -8.696, -8.77675, -8.8410000000000001, -8.88525, -8.9165, -8.927999999999999, -8.92575, -8.↵
9080000000000001, -8.87375, -8.825, -8.76, -8.6785, -8.58, -8.465499999999999, -8.336, -8.18875, -8.↵
0215, -7.8415, -7.6415, -7.4257500000000001, -7.1952500000000001, -6.9487499999999995, -6.6902500000000001, -6.↵
421, -6.14025, -5.851, -5.56025, -5.2627500000000005, -4.96525, -4.6662500000000001, -4.36925, -4.↵
0765, -3.7872500000000002, -3.5027500000000003, -3.2265, -2.95725, -2.6992499999999997, -2.4490000000000003, -2.↵
2087499999999998, -1.97975, -1.75625, -1.54425, -1.33825, -1.1405, -0.9497500000000001, -0.76525, -0.↵
583, -0.40425, -0.22975, -0.0557500000000001, 0.11375, 0.28225, 0.44925, 0.609499999999999, 0.↵
769, 0.92175, 1.06975, 1.21, 1.34575, 1.4755, 1.59825, 1.71425, 1.825, 1.93175, 2.03325, 2.13625, 2.↵
23775, 2.3415, 2.4475000000000002, 2.5565, 2.6702500000000002, 2.7885, 2.9145, 3.04575, 3.18375, 3.↵
32325, 3.47125, 3.6225, 3.77725, 3.9345, 4.09, 4.247999999999999, 4.4045000000000005, 4.5592500000000005, 4.↵
7105, 4.856, 4.99425, 5.1247500000000001, 5.2467500000000005, 5.359, 5.462, 5.55275, 5.6320000000000001, 5.↵
7015, 5.75875, 5.80525, 5.840999999999999, 5.8685, 5.888249999999999, 5.89975, 5.9060000000000001, 5.↵
909, 5.9060000000000001, 5.899, 5.887499999999999, 5.8715, 5.854, 5.82875, 5.79825, 5.76325, 5.718999999999999, 5.↵
6695, 5.61375, 5.552, 5.48325, 5.40775, 5.3267500000000005, 5.242249999999999, 5.1552500000000006, 5.↵
0637500000000001, 4.97125, 4.8774999999999995, 4.782, 4.6875, 4.589, 4.49, 4.3907500000000001, 4.↵
2885, 4.1892499999999995, 4.08925, 3.98475, 3.88325, 3.7795, 3.6765, 3.57675, 3.476, 3.37675, 3.2800000000000002, 3.↵
18375, 3.093, 3.006, 2.91675, 2.8335, 2.7512499999999998, 2.6710000000000003, 2.5917499999999998, 2.↵
51225, 2.4322500000000002, 2.35525, 2.27725, 2.2025, 2.131, 2.05925, 1.99125, 1.9265, 1.8644999999999998, 1.↵
809, 1.7555, 1.7075, 1.66475, 1.625, 1.58925, 1.55875, 1.528249999999999, 1.50675, 1.484, 1.4625000000000001, 1.↵
44575, 1.42675, 1.4075, 1.387, 1.36425, 1.3405, 1.3145, 1.29175, 1.27475, 1.25125, 1.2215, 1.19775, 1.↵
17725, 1.15825, 1.145249999999999, 1.133, 1.12925, 1.1315, 1.1375, 1.14825, 1.1627500000000002, 1.↵
184, 1.207, 1.22525, 1.24425, 1.262, 1.2765, 1.28325, 1.2855, 1.27725, 1.25875, 1.2315, 1.1962499999999998, 1.↵
14975, 1.097, 1.039, 0.972, 0.901, 0.827, 0.753, 0.67825, 0.6065, 0.5377500000000001, 0.4752499999999995, 0.↵
41725, 0.36475, 0.3212500000000004, 0.28675, 0.2562500000000003, 0.2350000000000001, 0.22425, 0.↵
2197499999999997, 0.2227499999999998, 0.2350000000000001, 0.25025, 0.27325, 0.303, 0.33575, 0.↵

```
370749999999999997,0.40825,0.44625000000000004,0.48225000000000007,0.5135,0.5425,0.56775,0.↵
58525,0.59500000000000001,0.5975,0.58900000000000001,0.57225,0.54325,0.50575,0.457,0.40124999999999994,0.↵
3365,0.26475,0.18625,0.1045,0.02225,-0.064,-0.14875000000000002,-0.23424999999999999,-0.↵
315750000000000003,-0.3915,-0.46074999999999994,-0.521,-0.573,-0.6165,-0.647,-0.6689999999999999,-0.↵
68125,-0.67975,-0.6715,-0.65225,-0.62175,-0.5875,-0.5455,-0.49824999999999997,-0.44775,-0.↵
393,-0.33575,-0.2785,-0.22125,-0.1625,-0.10375,-0.04575,0.01,0.06325,0.116,0.167,0.21825,0.↵
27,0.319,0.3685,0.41650000000000004,0.46699999999999997,0.5217499999999999,}
```

Definition at line 1 of file [processed-data.h](#).

4.4 processed-data.h

[Go to the documentation of this file.](#)

```
00001 float test_buffer[] = {5.656499999999999,5.667,5.673999999999995,5.672499999999999,5.6655,5.6525,5.638,5.625999999999999}
```

4.5 lib/audio/i2s_sampler.c File Reference

I2S Sampler Driver Source for ESP32.

```
#include "i2s_sampler.h"
```

Macros

- `#define RESAMPLE_DENOMINATOR 2`
- `#define READER_TIMEOUT_MS 10`
- `#define READER_TIMEOUT_TICKS (READER_TIMEOUT_MS / portTICK_PERIOD_MS)`

Functions

- `i2s_sampler_t * i2s_sampler_start (adc_channel_t adc1_channel, QueueHandle_t recv_queue, const size_t buffer_size, const size_t f_sample)`
Starts a sampler Task that samples from the given ADC1 Channel and sends the samples to the given Queue.
- `esp_err_t i2s_sampler_stop (i2s_sampler_t *sampler)`
Deletes sampler Task, frees memory and uninstalls adc as well as i2s driver.

4.5.1 Detailed Description

I2S Sampler Driver Source for ESP32.

Author

@s-grundner

Version

0.1

Date

2023-03-26

Copyright

Copyright (c) 2023

Definition in file [i2s_sampler.c](#).

4.5.2 Macro Definition Documentation

4.5.2.1 READER_TIMEOUT_MS

```
#define READER_TIMEOUT_MS 10
```

Definition at line 16 of file [i2s_sampler.c](#).

4.5.2.2 READER_TIMEOUT_TICKS

```
#define READER_TIMEOUT_TICKS (READER_TIMEOUT_MS / portTICK_PERIOD_MS)
```

Definition at line 17 of file [i2s_sampler.c](#).

4.5.2.3 RESAMPLE_DENOMINATOR

```
#define RESAMPLE_DENOMINATOR 2
```

Definition at line 15 of file [i2s_sampler.c](#).

4.5.3 Function Documentation

4.5.3.1 i2s_sampler_start()

```
i2s_sampler_t * i2s_sampler_start (  
    adc_channel_t adc1_channel,  
    QueueHandle_t recv_queue,  
    size_t buffer_size,  
    size_t f_sample )
```

Starts a sampler Task that samples from the given ADC1 Channel and sends the samples to the given Queue.

Parameters

<i>adc1_channel</i>	ADC1 Channel to use (Only ADC1 Channels are supported)
<i>recv_queue</i>	Queue to send samples to
<i>buffer_size</i>	Size of the buffer in samples
<i>f_sample</i>	Sample rate

Returns

i2s_sampler_t* Sampler context or NULL if failed

Definition at line 53 of file [i2s_sampler.c](#).

4.5.3.2 i2s_sampler_stop()

```
esp_err_t i2s_sampler_stop (
    i2s_sampler_t * sampler )
```

Deletes sampler Task, frees memory and uninstalls adc as well as i2s driver.

Parameters

<i>sampler</i>	Sampler to stop
----------------	-----------------

Returns

ESP_OK if successful

Definition at line 92 of file [i2s_sampler.c](#).

4.6 i2s_sampler.c

[Go to the documentation of this file.](#)

```
00001
00012 #include "i2s_sampler.h"
00013 static const char *TAG = "I2S_SAMPLER";
00014
00015 #define RESAMPLE_DENOMINATOR 2
00016 #define READER_TIMEOUT_MS 10
00017 #define READER_TIMEOUT_TICKS (READER_TIMEOUT_MS / portTICK_PERIOD_MS)
00018
00019 static TaskHandle_t sampler_task_handle;
00020
00021 static void IRAM_ATTR sampler_task(void *arg)
00022 {
00023     i2s_sampler_t *sampler = (i2s_sampler_t *)arg;
00024     for (;;)
00025     {
00026         i2s_event_t evt;
00027         if (xQueueReceive(sampler->dma_queue, &evt, portMAX_DELAY) == pdTRUE)
00028         {
00029             if (evt.type == I2S_EVENT_RX_DONE)
00030             {
00031                 size_t bytes_read = 0;
00032                 do
00033                 {
00034                     // fill audio buffer
00035                     size_t bytes_to_read = RESAMPLE_DENOMINATOR * (sampler->buffer_size -
sampler->buffer_pos);
00036                     void *buffer_position = (void *) (sampler->buffer + sampler->buffer_pos);
00037
00038                     // read data from i2s
00039                     i2s_read(I2S_NUM_0, buffer_position, bytes_to_read, &bytes_read,
READER_TIMEOUT_TICKS);
00040                     sampler->buffer_pos += bytes_read / RESAMPLE_DENOMINATOR;
00041
00042                     if (sampler->buffer_pos == sampler->buffer_size)
00043                     {
```

```

00044         sampler->buffer_pos = 0;
00045         xQueueSend(sampler->dsp_queue, &sampler->buffer, portMAX_DELAY);
00046     }
00047     } while (bytes_read > 0);
00048 }
00049 }
00050 }
00051 }
00052
00053 i2s_sampler_t *i2s_sampler_start(adc_channel_t adcl_channel, QueueHandle_t recv_queue, const size_t
    buffer_size, const size_t f_sample)
00054 {
00055     ESP_LOGI(TAG, "Initializing I2S Sampler...");
00056
00057     QueueHandle_t dma_queue;
00058
00059     i2s_config_t i2s_cfg = {
00060         .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX | I2S_MODE_ADC_BUILT_IN),
00061         .sample_rate = f_sample,
00062         .bits_per_sample = I2S_BITS_PER_SAMPLE_16BIT,
00063         .channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
00064         .communication_format = I2S_COMM_FORMAT_STAND_I2S,
00065         .intr_alloc_flags = ESP_INTR_FLAG_LEVEL1,
00066         .dma_buf_count = 4,
00067         .dma_buf_len = 1024,
00068         .use_apll = false,
00069         .fixed_mclk = 0};
00070
00071     // Initialize ADC
00072     ESP_ERROR_CHECK(i2s_driver_install(I2S_NUM_0, &i2s_cfg, 4, &dma_queue));
00073     ESP_ERROR_CHECK(i2s_set_adc_mode(ADC_UNIT_1, adcl_channel));
00074     ESP_ERROR_CHECK(i2s_adc_enable(I2S_NUM_0));
00075
00076     i2s_sampler_t *sampler = (i2s_sampler_t *)malloc(sizeof(i2s_sampler_t));
00077
00078     *sampler = (i2s_sampler_t){
00079         .buffer = (size_t *)malloc(buffer_size * sizeof(size_t)),
00080         .buffer_pos = 0,
00081         .buffer_size = buffer_size,
00082         .dma_queue = dma_queue,
00083         .dsp_queue = recv_queue};
00084
00085     // DMA task: receives audio data from ADC and sends it to DSP task
00086     if (xTaskCreatePinnedToCore(sampler_task, "sampler_task", 1 << 14, sampler, 5,
        &sampler_task_handle, 0) == pdFALSE)
00087         return NULL;
00088
00089     return sampler;
00090 }
00091
00092 esp_err_t i2s_sampler_stop(i2s_sampler_t *sampler)
00093 {
00094     ESP_LOGI(TAG, "Stopping I2S Sampler...");
00095     esp_err_t err;
00096     // stop i2s
00097     ESP_ERROR_CHECK(i2s_adc_disable(I2S_NUM_0));
00098     ESP_ERROR_CHECK(i2s_driver_uninstall(I2S_NUM_0));
00099
00100     // stop task
00101     vTaskDelete(sampler_task_handle);
00102
00103     // free memory
00104     free(sampler->buffer);
00105     free(sampler);
00106
00107     return ESP_OK;
00108 }

```

4.7 lib/audio/i2s_sampler.h File Reference

I2S Sampler Driver for ESP32.

```

#include <stdint.h>
#include "driver/i2s.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/queue.h"
#include "freertos/timers.h"

```

```
#include "esp_log.h"
```

Data Structures

- struct [i2s_sampler_t](#)
Sampler Configuration.

Functions

- [i2s_sampler_t](#) * [i2s_sampler_start](#) ([adc_channel_t](#) adc1_channel, [QueueHandle_t](#) recv_queue, [size_t](#) buffer_size, [size_t](#) f_sample)
Starts a sampler Task that samples from the given ADC1 Channel and sends the samples to the given Queue.
- [esp_err_t](#) [i2s_sampler_stop](#) ([i2s_sampler_t](#) *sampler)
Deletes sampler Task, frees memory and uninstalls adc as well as i2s driver.

4.7.1 Detailed Description

I2S Sampler Driver for ESP32.

Author

@s-grundner

Version

0.1

Date

2023-03-26

Copyright

Copyright (c) 2023

Definition in file [i2s_sampler.h](#).

4.7.2 Function Documentation

4.7.2.1 i2s_sampler_start()

```
i2s\_sampler\_t * i2s_sampler_start (  
    adc\_channel\_t adc1_channel,  
    QueueHandle\_t recv_queue,  
    size\_t buffer_size,  
    size\_t f_sample )
```

Starts a sampler Task that samples from the given ADC1 Channel and sends the samples to the given Queue.

Parameters

<i>adc1_channel</i>	ADC1 Channel to use (Only ADC1 Channels are supported)
<i>recv_queue</i>	Queue to send samples to
<i>buffer_size</i>	Size of the buffer in samples
<i>f_sample</i>	Sample rate

Returns

i2s_sampler_t* Sampler context or NULL if failed

Definition at line 53 of file [i2s_sampler.c](#).

4.7.2.2 i2s_sampler_stop()

```
esp_err_t i2s_sampler_stop (
    i2s_sampler_t * sampler )
```

Deletes sampler Task, frees memory and uninstalls adc as well as i2s driver.

Parameters

<i>sampler</i>	Sampler to stop
----------------	-----------------

Returns

ESP_OK if successful

Definition at line 92 of file [i2s_sampler.c](#).

4.8 i2s_sampler.h

[Go to the documentation of this file.](#)

```
00001
00011 #ifndef SAMPLING_H
00012 #define SAMPLING_H
00013
00014 #include <stdint.h>
00015 #include "driver/i2s.h"
00016 #include "freertos/FreeRTOS.h"
00017 #include "freertos/task.h"
00018 #include "freertos/queue.h"
00019 #include "freertos/timers.h"
00020 #include "esp_log.h"
00021
00030 typedef struct
00031 {
00032     QueueHandle_t dma_queue;
00033     QueueHandle_t dsp_queue;
00034     size_t *buffer;
00035     size_t buffer_pos;
00036     size_t buffer_size;
00037 } i2s_sampler_t;
00038
```

```

00048 i2s_sampler_t *i2s_sampler_start(adc_channel_t adc1_channel, QueueHandle_t recv_queue, size_t
      buffer_size, size_t f_sample);
00049
00055 esp_err_t i2s_sampler_stop(i2s_sampler_t *sampler);
00056
00057 #endif // SAMPLING_H

```

4.9 lib/fft/fft.c File Reference

```

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <complex.h>
#include "fft.h"

```

Macros

- #define [TWO_PI](#) 6.28318530
- #define [USE_SPLIT_RADIX](#) 1
- #define [LARGE_BASE_CASE](#) 1

Functions

- [fft_config_t](#) * [fft_init](#) (int size, [fft_type_t](#) type, [fft_direction_t](#) direction, float *input, float *output)
- void [fft_destroy](#) ([fft_config_t](#) *config)
- void [fft_execute](#) ([fft_config_t](#) *config)
- void [fft](#) (float *input, float *output, float *twiddle_factors, int n)
- void [ifft](#) (float *input, float *output, float *twiddle_factors, int n)
- void [rfft](#) (float *x, float *y, float *twiddle_factors, int n)
- void [irfft](#) (float *x, float *y, float *twiddle_factors, int n)
- void [fft_primitive](#) (float *x, float *y, int n, int stride, float *twiddle_factors, int tw_stride)
- void [split_radix_fft](#) (float *x, float *y, int n, int stride, float *twiddle_factors, int tw_stride)
- void [ifft_primitive](#) (float *input, float *output, int n, int stride, float *twiddle_factors, int tw_stride)
- void [fft8](#) (float *input, int stride_in, float *output, int stride_out)
- void [fft4](#) (float *input, int stride_in, float *output, int stride_out)

4.9.1 Macro Definition Documentation

4.9.1.1 LARGE_BASE_CASE

```
#define LARGE_BASE_CASE 1
```

Definition at line 11 of file [fft.c](#).

4.9.1.2 TWO_PI

```
#define TWO_PI 6.28318530
```

Definition at line 9 of file [fft.c](#).

4.9.1.3 USE_SPLIT_RADIX

```
#define USE_SPLIT_RADIX 1
```

Definition at line 10 of file [fft.c](#).

4.9.2 Function Documentation

4.9.2.1 fft()

```
void fft (  
    float * input,  
    float * output,  
    float * twiddle_factors,  
    int n )
```

Definition at line 104 of file [fft.c](#).

4.9.2.2 fft4()

```
void fft4 (  
    float * input,  
    int stride_in,  
    float * output,  
    int stride_out ) [inline]
```

Definition at line 627 of file [fft.c](#).

4.9.2.3 fft8()

```
void fft8 (  
    float * input,  
    int stride_in,  
    float * output,  
    int stride_out ) [inline]
```

Definition at line 500 of file [fft.c](#).

4.9.2.4 `fft_destroy()`

```
void fft_destroy (
    fft_config_t * config )
```

Definition at line 80 of file [fft.c](#).

4.9.2.5 `fft_execute()`

```
void fft_execute (
    fft_config_t * config )
```

Definition at line 92 of file [fft.c](#).

4.9.2.6 `fft_init()`

```
fft_config_t * fft_init (
    int size,
    fft_type_t type,
    fft_direction_t direction,
    float * input,
    float * output )
```

Definition at line 13 of file [fft.c](#).

4.9.2.7 `fft_primitive()`

```
void fft_primitive (
    float * x,
    float * y,
    int n,
    int stride,
    float * twiddle_factors,
    int tw_stride )
```

Definition at line 237 of file [fft.c](#).

4.9.2.8 `ifft()`

```
void ifft (
    float * input,
    float * output,
    float * twiddle_factors,
    int n )
```

Definition at line 129 of file [fft.c](#).

4.9.2.9 ifft_primitive()

```
void ifft_primitive (
    float * input,
    float * output,
    int n,
    int stride,
    float * twiddle_factors,
    int tw_stride )
```

Definition at line [464](#) of file [fft.c](#).

4.9.2.10 irfft()

```
void irfft (
    float * x,
    float * y,
    float * twiddle_factors,
    int n )
```

Definition at line [197](#) of file [fft.c](#).

4.9.2.11 rfft()

```
void rfft (
    float * x,
    float * y,
    float * twiddle_factors,
    int n )
```

Definition at line [149](#) of file [fft.c](#).

4.9.2.12 split_radix_fft()

```
void split_radix_fft (
    float * x,
    float * y,
    int n,
    int stride,
    float * twiddle_factors,
    int tw_stride )
```

Definition at line [325](#) of file [fft.c](#).

4.10 fft.c

[Go to the documentation of this file.](#)

```

00001
00002 #include <stdlib.h>
00003 #include <stdio.h>
00004 #include <math.h>
00005 #include <complex.h>
00006
00007 #include "fft.h"
00008
00009 #define TWO_PI 6.28318530
00010 #define USE_SPLIT_RADIX 1
00011 #define LARGE_BASE_CASE 1
00012
00013 fft_config_t* fft_init(int size, fft_type_t type, fft_direction_t direction, float* input, float*
    output)
00014 {
00015     /*
00016      * Prepare an FFT of correct size and types.
00017      *
00018      * If no input or output buffers are provided, they will be allocated.
00019      */
00020     int k, m;
00021
00022     fft_config_t* config = (fft_config_t*)malloc(sizeof(fft_config_t));
00023
00024     // Check if the size is a power of two
00025     if ((size & (size - 1)) != 0) // tests if size is a power of two
00026         return NULL;
00027
00028     // start configuration
00029     config->flags = 0;
00030     config->type = type;
00031     config->direction = direction;
00032     config->size = size;
00033
00034     // Allocate and precompute twiddle factors
00035     config->twiddle_factors = (float*)malloc(2 * config->size * sizeof(float));
00036
00037     float two_pi_by_n = TWO_PI / config->size;
00038
00039     for (k = 0, m = 0; k < config->size; k++, m += 2)
00040     {
00041         config->twiddle_factors[m] = cosf(two_pi_by_n * k); // real
00042         config->twiddle_factors[m + 1] = sinf(two_pi_by_n * k); // imag
00043     }
00044
00045     // Allocate input buffer
00046     if (input != NULL)
00047         config->input = input;
00048     else
00049     {
00050         if (config->type == FFT_REAL)
00051             config->input = (float*)malloc(config->size * sizeof(float));
00052         else if (config->type == FFT_COMPLEX)
00053             config->input = (float*)malloc(2 * config->size * sizeof(float));
00054
00055         config->flags |= FFT_OWN_INPUT_MEM;
00056     }
00057
00058     if (config->input == NULL)
00059         return NULL;
00060
00061     // Allocate output buffer
00062     if (output != NULL)
00063         config->output = output;
00064     else
00065     {
00066         if (config->type == FFT_REAL)
00067             config->output = (float*)malloc(config->size * sizeof(float));
00068         else if (config->type == FFT_COMPLEX)
00069             config->output = (float*)malloc(2 * config->size * sizeof(float));
00070
00071         config->flags |= FFT_OWN_OUTPUT_MEM;
00072     }
00073
00074     if (config->output == NULL)
00075         return NULL;
00076
00077     return config;
00078 }
00079
00080 void fft_destroy(fft_config_t* config)
00081 {

```

```

00082     if (config->flags & FFT_OWN_INPUT_MEM)
00083         free(config->input);
00084
00085     if (config->flags & FFT_OWN_OUTPUT_MEM)
00086         free(config->output);
00087
00088     free(config->twiddle_factors);
00089     free(config);
00090 }
00091
00092 void fft_execute(fft_config_t* config)
00093 {
00094     if (config->type == FFT_REAL && config->direction == FFT_FORWARD)
00095         rfft(config->input, config->output, config->twiddle_factors, config->size);
00096     else if (config->type == FFT_REAL && config->direction == FFT_BACKWARD)
00097         irfft(config->input, config->output, config->twiddle_factors, config->size);
00098     else if (config->type == FFT_COMPLEX && config->direction == FFT_FORWARD)
00099         fft(config->input, config->output, config->twiddle_factors, config->size);
00100     else if (config->type == FFT_COMPLEX && config->direction == FFT_BACKWARD)
00101         ifft(config->input, config->output, config->twiddle_factors, config->size);
00102 }
00103
00104 void fft(float* input, float* output, float* twiddle_factors, int n)
00105 {
00106     /*
00107      * Forward fast Fourier transform
00108      * DIT, radix-2, out-of-place implementation
00109      *
00110      * Parameters
00111      * -----
00112      * input (float *)
00113      *     The input array containing the complex samples with
00114      *     real/imaginary parts interleaved [Re(x0), Im(x0), ..., Re(xn-1), Im(xn-1)]
00115      * output (float *)
00116      *     The output array containing the complex samples with
00117      *     real/imaginary parts interleaved [Re(x0), Im(x0), ..., Re(xn-1), Im(xn-1)]
00118      * n (int)
00119      *     The FFT size, should be a power of 2
00120      */
00121
00122     #if USE_SPLIT_RADIX
00123         split_radix_fft(input, output, n, 2, twiddle_factors, 2);
00124     #else
00125         fft_primitive(input, output, n, 2, twiddle_factors, 2);
00126     #endif
00127 }
00128
00129 void ifft(float* input, float* output, float* twiddle_factors, int n)
00130 {
00131     /*
00132      * Inverse fast Fourier transform
00133      * DIT, radix-2, out-of-place implementation
00134      *
00135      * Parameters
00136      * -----
00137      * input (float *)
00138      *     The input array containing the complex samples with
00139      *     real/imaginary parts interleaved [Re(x0), Im(x0), ..., Re(xn-1), Im(xn-1)]
00140      * output (float *)
00141      *     The output array containing the complex samples with
00142      *     real/imaginary parts interleaved [Re(x0), Im(x0), ..., Re(xn-1), Im(xn-1)]
00143      * n (int)
00144      *     The FFT size, should be a power of 2
00145      */
00146     ifft_primitive(input, output, n, 2, twiddle_factors, 2);
00147 }
00148
00149 void rfft(float* x, float* y, float* twiddle_factors, int n)
00150 {
00151     // This code uses the two-for-the-price-of-one strategy
00152     #if USE_SPLIT_RADIX
00153         split_radix_fft(x, y, n / 2, 2, twiddle_factors, 4);
00154     #else
00155         fft_primitive(x, y, n / 2, 2, twiddle_factors, 4);
00156     #endif
00157
00158     // Now apply post processing to recover positive
00159     // frequencies of the real FFT
00160     float t = y[0];
00161     y[0] = t + y[1]; // DC coefficient
00162     y[1] = t - y[1]; // Center coefficient
00163
00164     // Apply post processing to quarter element
00165     // this boils down to taking complex conjugate
00166     y[n / 2 + 1] = -y[n / 2 + 1];
00167 }
00168

```

```

00169 // Now process all the other frequencies
00170 int k;
00171 for (k = 2; k < n / 2; k += 2)
00172 {
00173     float xer, xei, xor_t, xoi, c, s, tr, ti;
00174
00175     c = twiddle_factors[k];
00176     s = twiddle_factors[k + 1];
00177
00178     // even half coefficient
00179     xer = 0.5 * (y[k] + y[n - k]);
00180     xei = 0.5 * (y[k + 1] - y[n - k + 1]);
00181
00182     // odd half coefficient
00183     xor_t = 0.5 * (y[k + 1] + y[n - k + 1]);
00184     xoi = -0.5 * (y[k] - y[n - k]);
00185
00186     tr = c * xor_t + s * xoi;
00187     ti = -s * xor_t + c * xoi;
00188
00189     y[k] = xer + tr;
00190     y[k + 1] = xei + ti;
00191
00192     y[n - k] = xer - tr;
00193     y[n - k + 1] = -(xei - ti);
00194 }
00195 }
00196
00197 void irfft(float* x, float* y, float* twiddle_factors, int n)
00198 {
00199     /*
00200      * Destroys content of input vector
00201      */
00202     int k;
00203
00204     // Here we need to apply a pre-processing first
00205     float t = x[0];
00206     x[0] = 0.5 * (t + x[1]);
00207     x[1] = 0.5 * (t - x[1]);
00208
00209     x[n / 2 + 1] = -x[n / 2 + 1];
00210
00211     for (k = 2; k < n / 2; k += 2)
00212     {
00213         float xer, xei, xor_t, xoi, c, s, tr, ti;
00214
00215         c = twiddle_factors[k];
00216         s = twiddle_factors[k + 1];
00217
00218         xer = 0.5 * (x[k] + x[n - k]);
00219         tr = 0.5 * (x[k] - x[n - k]);
00220
00221         xei = 0.5 * (x[k + 1] - x[n - k + 1]);
00222         ti = 0.5 * (x[k + 1] + x[n - k + 1]);
00223
00224         xor_t = c * tr - s * ti;
00225         xoi = s * tr + c * ti;
00226
00227         x[k] = xer - xoi;
00228         x[k + 1] = xor_t + xei;
00229
00230         x[n - k] = xer + xoi;
00231         x[n - k + 1] = xor_t - xei;
00232     }
00233
00234     ifft_primitive(x, y, n / 2, 2, twiddle_factors, 4);
00235 }
00236
00237 void fft_primitive(float* x, float* y, int n, int stride, float* twiddle_factors, int tw_stride)
00238 {
00239     /*
00240      * This code will compute the FFT of the input vector x
00241      *
00242      * The input data is assumed to be real/imag interleaved
00243      *
00244      * The size n should be a power of two
00245      *
00246      * y is an output buffer of size 2n to accomodate for complex numbers
00247      *
00248      * Forward fast Fourier transform
00249      * DIT, radix-2, out-of-place implementation
00250      *
00251      * For a complex FFT, call first stage as:
00252      * fft(x, y, n, 2, 2);
00253      *
00254      * Parameters
00255      * -----

```



```

00256     * x (float *)
00257     *   The input array containing the complex samples with
00258     *   real/imaginary parts interleaved [Re(x0), Im(x0), ..., Re(x_n-1), Im(x_n-1)]
00259     * y (float *)
00260     *   The output array containing the complex samples with
00261     *   real/imaginary parts interleaved [Re(x0), Im(x0), ..., Re(x_n-1), Im(x_n-1)]
00262     * n (int)
00263     *   The FFT size, should be a power of 2
00264     * stride (int)
00265     *   The number of elements to skip between two successive samples
00266     * tw_stride (int)
00267     *   The number of elements to skip between two successive twiddle factors
00268     */
00269     int k;
00270     float t;
00271
00272     #if LARGE_BASE_CASE
00273     // End condition, stop at n=8 to avoid one trivial recursion
00274     if (n == 8)
00275     {
00276         fft8(x, stride, y, 2);
00277         return;
00278     }
00279     #else
00280     // End condition, stop at n=2 to avoid one trivial recursion
00281     if (n == 2)
00282     {
00283         y[0] = x[0] + x[stride];
00284         y[1] = x[1] + x[stride + 1];
00285         y[2] = x[0] - x[stride];
00286         y[3] = x[1] - x[stride + 1];
00287         return;
00288     }
00289     #endif
00290
00291     // Recursion -- Decimation In Time algorithm
00292     fft_primitive(x, y, n / 2, 2 * stride, twiddle_factors, 2 * tw_stride); // even half
00293     fft_primitive(x + stride, y + n, n / 2, 2 * stride, twiddle_factors, 2 * tw_stride); // odd half
00294
00295     // Stitch back together
00296
00297     // We can a few multiplications in the first step
00298     t = y[0];
00299     y[0] = t + y[n];
00300     y[n] = t - y[n];
00301
00302     t = y[1];
00303     y[1] = t + y[n + 1];
00304     y[n + 1] = t - y[n + 1];
00305
00306     for (k = 1; k < n / 2; k++)
00307     {
00308         float x1r, x1i, x2r, x2i, c, s;
00309         c = twiddle_factors[k * tw_stride];
00310         s = twiddle_factors[k * tw_stride + 1];
00311
00312         x1r = y[2 * k];
00313         x1i = y[2 * k + 1];
00314         x2r = c * y[n + 2 * k] + s * y[n + 2 * k + 1];
00315         x2i = -s * y[n + 2 * k] + c * y[n + 2 * k + 1];
00316
00317         y[2 * k] = x1r + x2r;
00318         y[2 * k + 1] = x1i + x2i;
00319
00320         y[n + 2 * k] = x1r - x2r;
00321         y[n + 2 * k + 1] = x1i - x2i;
00322     }
00323 }
00324
00325 void split_radix_fft(float* x, float* y, int n, int stride, float* twiddle_factors, int tw_stride)
00326 {
00327     /*
00328     * This code will compute the FFT of the input vector x
00329     *
00330     * The input data is assumed to be real/imag interleaved
00331     *
00332     * The size n should be a power of two
00333     *
00334     * y is an output buffer of size 2n to accomodate for complex numbers
00335     *
00336     * Forward fast Fourier transform
00337     * Split-Radix
00338     * DIT, radix-2, out-of-place implementation
00339     *
00340     * For a complex FFT, call first stage as:
00341     * fft(x, y, n, 2, 2);
00342     */

```

```

00343     * Parameters
00344     * -----
00345     * x (float *)
00346     *   The input array containing the complex samples with
00347     *   real/imaginary parts interleaved [Re(x0), Im(x0), ..., Re(x_n-1), Im(x_n-1)]
00348     * y (float *)
00349     *   The output array containing the complex samples with
00350     *   real/imaginary parts interleaved [Re(x0), Im(x0), ..., Re(x_n-1), Im(x_n-1)]
00351     * n (int)
00352     *   The FFT size, should be a power of 2
00353     * stride (int)
00354     *   The number of elements to skip between two successive samples
00355     * twiddle_factors (float *)
00356     *   The array of twiddle factors
00357     * tw_stride (int)
00358     *   The number of elements to skip between two successive twiddle factors
00359     */
00360     int k;
00361
00362     #if LARGE_BASE_CASE
00363     // End condition, stop at n=2 to avoid one trivial recursion
00364     if (n == 8)
00365     {
00366         fft8(x, stride, y, 2);
00367         return;
00368     }
00369     else if (n == 4)
00370     {
00371         fft4(x, stride, y, 2);
00372         return;
00373     }
00374     #else
00375     // End condition, stop at n=2 to avoid one trivial recursion
00376     if (n == 2)
00377     {
00378         y[0] = x[0] + x[stride];
00379         y[1] = x[1] + x[stride + 1];
00380         y[2] = x[0] - x[stride];
00381         y[3] = x[1] - x[stride + 1];
00382         return;
00383     }
00384     else if (n == 1)
00385     {
00386         y[0] = x[0];
00387         y[1] = x[1];
00388         return;
00389     }
00390     #endif
00391
00392     // Recursion -- Decimation In Time algorithm
00393     split_radix_fft(x, y, n / 2, 2 * stride, twiddle_factors, 2 * tw_stride);
00394     split_radix_fft(x + stride, y + n, n / 4, 4 * stride, twiddle_factors, 4 * tw_stride);
00395     split_radix_fft(x + 3 * stride, y + n + n / 2, n / 4, 4 * stride, twiddle_factors, 4 * tw_stride);
00396
00397     // Stitch together the output
00398     float ulr, uli, u2r, u2i, x1r, x1i, x2r, x2i;
00399     float t;
00400
00401     // We can save a few multiplications in the first step
00402     ulr = y[0];
00403     uli = y[1];
00404     u2r = y[n / 2];
00405     u2i = y[n / 2 + 1];
00406
00407     x1r = y[n];
00408     x1i = y[n + 1];
00409     x2r = y[n / 2 + n];
00410     x2i = y[n / 2 + n + 1];
00411
00412     t = x1r + x2r;
00413     y[0] = ulr + t;
00414     y[n] = ulr - t;
00415
00416     t = x1i + x2i;
00417     y[1] = uli + t;
00418     y[n + 1] = uli - t;
00419
00420     t = x2i - x1i;
00421     y[n / 2] = u2r - t;
00422     y[n + n / 2] = u2r + t;
00423
00424     t = x1r - x2r;
00425     y[n / 2 + 1] = u2i - t;
00426     y[n + n / 2 + 1] = u2i + t;
00427
00428     for (k = 1; k < n / 4; k++)
00429     {

```

```

00430     float u1r, u1i, u2r, u2i, x1r, x1i, x2r, x2i, c1, s1, c2, s2;
00431     c1 = twiddle_factors[k * tw_stride];
00432     s1 = twiddle_factors[k * tw_stride + 1];
00433     c2 = twiddle_factors[3 * k * tw_stride];
00434     s2 = twiddle_factors[3 * k * tw_stride + 1];
00435
00436     u1r = y[2 * k];
00437     u1i = y[2 * k + 1];
00438     u2r = y[2 * k + n / 2];
00439     u2i = y[2 * k + n / 2 + 1];
00440
00441     x1r = c1 * y[n + 2 * k] + s1 * y[n + 2 * k + 1];
00442     x1i = -s1 * y[n + 2 * k] + c1 * y[n + 2 * k + 1];
00443     x2r = c2 * y[n / 2 + n + 2 * k] + s2 * y[n / 2 + n + 2 * k + 1];
00444     x2i = -s2 * y[n / 2 + n + 2 * k] + c2 * y[n / 2 + n + 2 * k + 1];
00445
00446     t = x1r + x2r;
00447     y[2 * k] = u1r + t;
00448     y[2 * k + n] = u1r - t;
00449
00450     t = x1i + x2i;
00451     y[2 * k + 1] = u1i + t;
00452     y[2 * k + n + 1] = u1i - t;
00453
00454     t = x2i - x1i;
00455     y[2 * k + n / 2] = u2r - t;
00456     y[2 * k + n + n / 2] = u2r + t;
00457
00458     t = x1r - x2r;
00459     y[2 * k + n / 2 + 1] = u2i - t;
00460     y[2 * k + n + n / 2 + 1] = u2i + t;
00461 }
00462 }
00463
00464 void ifft_primitive(float* input, float* output, int n, int stride, float* twiddle_factors, int
tw_stride)
00465 {
00466
00467     #if USE_SPLIT_RADIX
00468         split_radix_fft(input, output, n, stride, twiddle_factors, tw_stride);
00469     #else
00470         fft_primitive(input, output, n, stride, twiddle_factors, tw_stride);
00471     #endif
00472
00473     int ks;
00474
00475     int ns = n * stride;
00476
00477     // reverse all coefficients from 1 to n / 2 - 1
00478     for (ks = stride; ks < ns / 2; ks += stride)
00479     {
00480         float t;
00481
00482         t = output[ks];
00483         output[ks] = output[ns - ks];
00484         output[ns - ks] = t;
00485
00486         t = output[ks + 1];
00487         output[ks + 1] = output[ns - ks + 1];
00488         output[ns - ks + 1] = t;
00489     }
00490
00491     // Apply normalization
00492     float norm = 1. / n;
00493     for (ks = 0; ks < ns; ks += stride)
00494     {
00495         output[ks] *= norm;
00496         output[ks + 1] *= norm;
00497     }
00498 }
00499
00500 inline void fft8(float* input, int stride_in, float* output, int stride_out)
00501 {
00502     /*
00503      * Unrolled implementation of FFT8 for a little more performance
00504      */
00505     float a0r, a1r, a2r, a3r, a4r, a5r, a6r, a7r;
00506     float a0i, a1i, a2i, a3i, a4i, a5i, a6i, a7i;
00507     float b0r, b1r, b2r, b3r, b4r, b5r, b6r, b7r;
00508     float b0i, b1i, b2i, b3i, b4i, b5i, b6i, b7i;
00509     float t;
00510     float sin_pi_4 = 0.7071067812;
00511
00512     a0r = input[0];
00513     a0i = input[1];
00514     a1r = input[stride_in];
00515     a1i = input[stride_in + 1];

```

```

00516     a2r = input[2 * stride_in];
00517     a2i = input[2 * stride_in + 1];
00518     a3r = input[3 * stride_in];
00519     a3i = input[3 * stride_in + 1];
00520     a4r = input[4 * stride_in];
00521     a4i = input[4 * stride_in + 1];
00522     a5r = input[5 * stride_in];
00523     a5i = input[5 * stride_in + 1];
00524     a6r = input[6 * stride_in];
00525     a6i = input[6 * stride_in + 1];
00526     a7r = input[7 * stride_in];
00527     a7i = input[7 * stride_in + 1];
00528
00529     // Stage 1
00530
00531     b0r = a0r + a4r;
00532     b0i = a0i + a4i;
00533
00534     b1r = a1r + a5r;
00535     b1i = a1i + a5i;
00536
00537     b2r = a2r + a6r;
00538     b2i = a2i + a6i;
00539
00540     b3r = a3r + a7r;
00541     b3i = a3i + a7i;
00542
00543     b4r = a0r - a4r;
00544     b4i = a0i - a4i;
00545
00546     b5r = a1r - a5r;
00547     b5i = a1i - a5i;
00548     // W_8^1 = 1/sqrt(2) - j / sqrt(2)
00549     t = b5r + b5i;
00550     b5i = (b5i - b5r) * sin_pi_4;
00551     b5r = t * sin_pi_4;
00552
00553     // W_8^2 = -j
00554     b6r = a2i - a6i;
00555     b6i = a6r - a2r;
00556
00557     b7r = a3r - a7r;
00558     b7i = a3i - a7i;
00559     // W_8^3 = -1 / sqrt(2) + j / sqrt(2)
00560     t = sin_pi_4 * (b7i - b7r);
00561     b7i = -(b7r + b7i) * sin_pi_4;
00562     b7r = t;
00563
00564     // Stage 2
00565
00566     a0r = b0r + b2r;
00567     a0i = b0i + b2i;
00568
00569     a1r = b1r + b3r;
00570     a1i = b1i + b3i;
00571
00572     a2r = b0r - b2r;
00573     a2i = b0i - b2i;
00574
00575     // * j
00576     a3r = b1i - b3i;
00577     a3i = b3r - b1r;
00578
00579     a4r = b4r + b6r;
00580     a4i = b4i + b6i;
00581
00582     a5r = b5r + b7r;
00583     a5i = b5i + b7i;
00584
00585     a6r = b4r - b6r;
00586     a6i = b4i - b6i;
00587
00588     // * j
00589     a7r = b5i - b7i;
00590     a7i = b7r - b5r;
00591
00592     // Stage 3
00593
00594     // X[0]
00595     output[0] = a0r + a1r;
00596     output[1] = a0i + a1i;
00597
00598     // X[4]
00599     output[4 * stride_out] = a0r - a1r;
00600     output[4 * stride_out + 1] = a0i - a1i;
00601
00602     // X[2]

```

```

00603     output[2 * stride_out] = a2r + a3r;
00604     output[2 * stride_out + 1] = a2i + a3i;
00605
00606     // X[6]
00607     output[6 * stride_out] = a2r - a3r;
00608     output[6 * stride_out + 1] = a2i - a3i;
00609
00610     // X[1]
00611     output[stride_out] = a4r + a5r;
00612     output[stride_out + 1] = a4i + a5i;
00613
00614     // X[5]
00615     output[5 * stride_out] = a4r - a5r;
00616     output[5 * stride_out + 1] = a4i - a5i;
00617
00618     // X[3]
00619     output[3 * stride_out] = a6r + a7r;
00620     output[3 * stride_out + 1] = a6i + a7i;
00621
00622     // X[7]
00623     output[7 * stride_out] = a6r - a7r;
00624     output[7 * stride_out + 1] = a6i - a7i;
00625 }
00626
00627 inline void fft4(float* input, int stride_in, float* output, int stride_out)
00628 {
00629     /*
00630      * Unrolled implementation of FFT4 for a little more performance
00631      */
00632     float t1, t2;
00633
00634     t1 = input[0] + input[2 * stride_in];
00635     t2 = input[stride_in] + input[3 * stride_in];
00636     output[0] = t1 + t2;
00637     output[2 * stride_out] = t1 - t2;
00638
00639     t1 = input[1] + input[2 * stride_in + 1];
00640     t2 = input[stride_in + 1] + input[3 * stride_in + 1];
00641     output[1] = t1 + t2;
00642     output[2 * stride_out + 1] = t1 - t2;
00643
00644     t1 = input[0] - input[2 * stride_in];
00645     t2 = input[stride_in + 1] - input[3 * stride_in + 1];
00646     output[stride_out] = t1 + t2;
00647     output[3 * stride_out] = t1 - t2;
00648
00649     t1 = input[1] - input[2 * stride_in + 1];
00650     t2 = input[3 * stride_in] - input[stride_in];
00651     output[stride_out + 1] = t1 + t2;
00652     output[3 * stride_out + 1] = t1 - t2;
00653 }

```

4.11 lib/fft/fft.h File Reference

Data Structures

- struct [fft_config_t](#)

Macros

- #define [FFT_OWN_INPUT_MEM](#) 1
- #define [FFT_OWN_OUTPUT_MEM](#) 2

Enumerations

- enum [fft_type_t](#) { [FFT_REAL](#) , [FFT_COMPLEX](#) }
ESP32 FFT. This provides a vanilla radix-2 FFT implementation and a test example.
- enum [fft_direction_t](#) { [FFT_FORWARD](#) , [FFT_BACKWARD](#) }

Functions

- `fft_config_t * fft_init` (int size, `fft_type_t` type, `fft_direction_t` direction, float *input, float *output)
- void `fft_destroy` (`fft_config_t` *config)
- void `fft_execute` (`fft_config_t` *config)
- void `fft` (float *input, float *output, float *twiddle_factors, int n)
- void `ifft` (float *input, float *output, float *twiddle_factors, int n)
- void `rfft` (float *x, float *y, float *twiddle_factors, int n)
- void `irfft` (float *x, float *y, float *twiddle_factors, int n)
- void `fft_primitive` (float *x, float *y, int n, int stride, float *twiddle_factors, int tw_stride)
- void `split_radix_fft` (float *x, float *y, int n, int stride, float *twiddle_factors, int tw_stride)
- void `ifft_primitive` (float *input, float *output, int n, int stride, float *twiddle_factors, int tw_stride)
- void `fft8` (float *input, int stride_in, float *output, int stride_out)
- void `fft4` (float *input, int stride_in, float *output, int stride_out)

4.11.1 Macro Definition Documentation

4.11.1.1 FFT_OWN_INPUT_MEM

```
#define FFT_OWN_INPUT_MEM 1
```

Definition at line 40 of file `fft.h`.

4.11.1.2 FFT_OWN_OUTPUT_MEM

```
#define FFT_OWN_OUTPUT_MEM 2
```

Definition at line 41 of file `fft.h`.

4.11.2 Enumeration Type Documentation

4.11.2.1 fft_direction_t

```
enum fft_direction_t
```

Enumerator

FFT_FORWARD	
FFT_BACKWARD	

Definition at line 34 of file [fft.h](#).

4.11.2.2 `fft_type_t`

enum `fft_type_t`

ESP32 FFT. This provides a vanilla radix-2 FFT implementation and a test example.

Author

This code was written by [Robin Scheibler](#)

Date

during rainy days in October 2017.

Copyright

(c) 2017 Robin Scheibler Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Enumerator

FFT_REAL	
FFT_COMPLEX	

Definition at line 28 of file [fft.h](#).

4.11.3 Function Documentation

4.11.3.1 `fft()`

```
void fft (
    float * input,
    float * output,
    float * twiddle_factors,
    int n )
```

Definition at line 104 of file [fft.c](#).

4.11.3.2 `fft4()`

```
void fft4 (
    float * input,
    int stride_in,
    float * output,
    int stride_out ) [inline]
```

Definition at line 627 of file [fft.c](#).

4.11.3.3 `fft8()`

```
void fft8 (
    float * input,
    int stride_in,
    float * output,
    int stride_out ) [inline]
```

Definition at line 500 of file [fft.c](#).

4.11.3.4 `fft_destroy()`

```
void fft_destroy (
    fft\_config\_t * config )
```

Definition at line 80 of file [fft.c](#).

4.11.3.5 `fft_execute()`

```
void fft_execute (
    fft\_config\_t * config )
```

Definition at line 92 of file [fft.c](#).

4.11.3.6 `fft_init()`

```
fft_config_t * fft_init (
    int size,
    fft_type_t type,
    fft_direction_t direction,
    float * input,
    float * output )
```

Definition at line 13 of file [fft.c](#).

4.11.3.7 `fft_primitive()`

```
void fft_primitive (
    float * x,
    float * y,
    int n,
    int stride,
    float * twiddle_factors,
    int tw_stride )
```

Definition at line 237 of file [fft.c](#).

4.11.3.8 `ifft()`

```
void ifft (
    float * input,
    float * output,
    float * twiddle_factors,
    int n )
```

Definition at line 129 of file [fft.c](#).

4.11.3.9 `ifft_primitive()`

```
void ifft_primitive (
    float * input,
    float * output,
    int n,
    int stride,
    float * twiddle_factors,
    int tw_stride )
```

Definition at line 464 of file [fft.c](#).

4.11.3.10 irfft()

```
void irfft (
    float * x,
    float * y,
    float * twiddle_factors,
    int n )
```

Definition at line 197 of file [fft.c](#).

4.11.3.11 rfft()

```
void rfft (
    float * x,
    float * y,
    float * twiddle_factors,
    int n )
```

Definition at line 149 of file [fft.c](#).

4.11.3.12 split_radix_fft()

```
void split_radix_fft (
    float * x,
    float * y,
    int n,
    int stride,
    float * twiddle_factors,
    int tw_stride )
```

Definition at line 325 of file [fft.c](#).

4.12 fft.h

[Go to the documentation of this file.](#)

```
00001
00025 #ifndef __FFT_H__
00026 #define __FFT_H__
00027
00028 typedef enum
00029 {
00030     FFT_REAL,
00031     FFT_COMPLEX
00032 } fft_type_t;
00033
00034 typedef enum
00035 {
00036     FFT_FORWARD,
00037     FFT_BACKWARD
00038 } fft_direction_t;
00039
00040 #define FFT_OWN_INPUT_MEM 1
00041 #define FFT_OWN_OUTPUT_MEM 2
00042
```

```

00043 typedef struct
00044 {
00045     int size;                // FFT size
00046     float *input;            // pointer to input buffer
00047     float *output;           // pointer to output buffer
00048     float *twiddle_factors;  // pointer to buffer holding twiddle factors
00049     fft_type_t type;         // real or complex
00050     fft_direction_t direction; // forward or backward
00051     unsigned int flags;      // FFT flags
00052 } fft_config_t;
00053
00054 fft_config_t *fft_init(int size, fft_type_t type, fft_direction_t direction, float *input, float
    *output);
00055 void fft_destroy(fft_config_t *config);
00056 void fft_execute(fft_config_t *config);
00057 void fft(float *input, float *output, float *twiddle_factors, int n);
00058 void ifft(float *input, float *output, float *twiddle_factors, int n);
00059 void rfft(float *x, float *y, float *twiddle_factors, int n);
00060 void irfft(float *x, float *y, float *twiddle_factors, int n);
00061 void fft_primitive(float *x, float *y, int n, int stride, float *twiddle_factors, int tw_stride);
00062 void split_radix_fft(float *x, float *y, int n, int stride, float *twiddle_factors, int tw_stride);
00063 void ifft_primitive(float *input, float *output, int n, int stride, float *twiddle_factors, int
    tw_stride);
00064 void fft8(float *input, int stride_in, float *output, int stride_out);
00065 void fft4(float *input, int stride_in, float *output, int stride_out);
00066
00067 #endif // __FFT_H__

```

4.13 lib/mcp3201/mcp3201.c File Reference

```
#include "mcp3201.h"
```

Data Structures

- struct [mcp3201_context_t](#)
MCP3201 Context struct for internal use.

Typedefs

- typedef struct [mcp3201_context_t](#) [mcp3201_context_t](#)

Functions

- [esp_err_t mcp3201_init](#) ([mcp3201_context_t](#) **out_ctx, const [mcp3201_config_t](#) *cfg)
- [esp_err_t mcp3201_read](#) ([mcp3201_context_t](#) *ctx, [uint16_t](#) *out_value)
- [esp_err_t mcp3201_exit](#) ([mcp3201_handle_t](#) mcp_handle)
Exits the MCP3201 ADC device and frees all resources.

4.13.1 Detailed Description

Author

@s-grundner

Version

0.1

Date

2022-12-24

Copyright

Copyright (c) 2022

Definition in file [mcp3201.c](#).

4.13.2 Typedef Documentation

4.13.2.1 mcp3201_context_t

```
typedef struct mcp3201_context_t mcp3201_context_t
```

Definition at line 28 of file [mcp3201.c](#).

4.13.3 Function Documentation

4.13.3.1 mcp3201_exit()

```
esp_err_t mcp3201_exit (
    mcp3201_handle_t mcp_handle )
```

Exits the MCP3201 ADC device and frees all resources.

Parameters

<i>mcp_handle</i>	MCP3201 Device to exit
-------------------	------------------------

Returns

ESP_OK on success

Definition at line 105 of file [mcp3201.c](#).

4.13.3.2 mcp3201_init()

```
esp_err_t mcp3201_init (
    mcp3201_context_t ** out_ctx,
    const mcp3201_config_t * cfg )
```

Definition at line 42 of file [mcp3201.c](#).

4.13.3.3 mcp3201_read()

```
esp_err_t mcp3201_read (
    mcp3201_context_t * ctx,
    uint16_t * out_value )
```

Definition at line 88 of file [mcp3201.c](#).

4.14 mcp3201.c

[Go to the documentation of this file.](#)

```
00001
00012 #include "mcp3201.h"
00013
00014 static const char *TAG = "mcp3201";
00015
00022 struct mcp3201_context_t
00023 {
00024     mcp3201_config_t cfg;
00025     spi_device_handle_t spi;
00026     spi_transaction_t *ongoing_transaction;
00027 };
00028 typedef struct mcp3201_context_t mcp3201_context_t;
00029
00030 static void cs_low(spi_transaction_t *t)
00031 {
00032     mcp3201_handle_t mcp_handle = (mcp3201_handle_t)t->user;
00033     gpio_set_level(mcp_handle->cfg.cs_io, 0);
00034 }
00035
00036 static void cs_high(spi_transaction_t *t)
00037 {
00038     mcp3201_handle_t mcp_handle = (mcp3201_handle_t)t->user;
00039     gpio_set_level(mcp_handle->cfg.cs_io, 1);
00040 }
00041
00042 esp_err_t mcp3201_init(mcp3201_context_t **out_ctx, const mcp3201_config_t *cfg)
00043 {
00044     mcp3201_context_t *ctx = (mcp3201_context_t *)malloc(sizeof(mcp3201_context_t));
00045     esp_err_t err = ESP_OK;
00046     if (ctx == NULL)
00047     {
00048         ESP_LOGE(TAG, "Failed to allocate memory for mcp3201 context");
00049         return ESP_ERR_NO_MEM;
00050     }
00051
00052     *ctx = (mcp3201_context_t){
00053         .cfg = *cfg;
00054
00055         spi_device_interface_config_t dev_cfg = {
00056             .clock_speed_hz = ADC_CLK,
00057             .mode = 0,
00058             .spics_io_num = ctx->cfg.cs_io,
00059             .queue_size = 1,
00060             .pre_cb = cs_low,
00061             .post_cb = cs_high,
00062             .command_bits = 0,
00063             .address_bits = 0,
00064             .dummy_bits = 0,
00065             .flags = 0,
00066         };
00067
00068         err = spi_bus_add_device(ctx->cfg.host, &dev_cfg, &ctx->spi);
00069         ESP_ERROR_CHECK(err);
00070         if (err != ESP_OK)
00071         {
00072             ESP_LOGE(TAG, "Failed to add device to spi bus");
00073             goto cleanup;
00074         }
00075         *out_ctx = ctx;
```

```

00076     return err;
00077
00078 cleanup:
00079     if (ctx->spi)
00080     {
00081         spi_bus_remove_device(ctx->spi);
00082         ctx->spi = NULL;
00083     }
00084     free(ctx);
00085     return err;
00086 }
00087
00088 esp_err_t mcp3201_read(mcp3201_context_t *ctx, uint16_t *out_value)
00089 {
00090     esp_err_t err = ESP_OK;
00091     spi_transaction_t t = {
00092         .user = (void *)ctx,
00093         .length = 16,
00094         .rx_buffer = out_value,
00095     };
00096     err = spi_device_polling_transmit(ctx->spi, &t);
00097     if (err != ESP_OK)
00098     {
00099         ESP_LOGE(TAG, "Failed to transmit to mcp3201");
00100         return err;
00101     }
00102     return err;
00103 }
00104
00105 esp_err_t mcp3201_exit(mcp3201_handle_t mcp_handle)
00106 {
00107     esp_err_t err = ESP_OK;
00108
00109     if (mcp_handle->spi)
00110     {
00111         err = spi_bus_remove_device(mcp_handle->spi);
00112         if (err != ESP_OK)
00113         {
00114             ESP_LOGE(TAG, "Failed to remove device from spi bus");
00115             return err;
00116         }
00117         mcp_handle->spi = NULL;
00118     }
00119
00120     free(mcp_handle);
00121     return err;
00122 }

```

4.15 lib/mcp3201/mcp3201.h File Reference

```

#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/queue.h"
#include "driver/gpio.h"
#include "driver/spi_master.h"
#include "hal/spi_types.h"
#include "esp_log.h"

```

Data Structures

- struct [mcp3201_config_t](#)
- struct [mcp3201_sampler_t](#)

Macros

- #define [ADC_CLK](#) SPI_MASTER_FREQ_8M

Typedefs

- typedef struct [mcp3201_context_t](#) * [mcp3201_handle_t](#)

Functions

- [esp_err_t](#) [mcp3201_init](#) ([mcp3201_handle_t](#) *out_handle, const [mcp3201_config_t](#) *cfg)
Initializes the MCP3201 ADC device.
- [esp_err_t](#) [mcp3201_exit](#) ([mcp3201_handle_t](#) mcp_handle)
Exits the MCP3201 ADC device and frees all resources.
- [esp_err_t](#) [mcp3201_read](#) ([mcp3201_handle_t](#) handle, [uint16_t](#) *out_value)
Reads a single value from the MCP3201 ADC.
- [mcp3201_sampler_t](#) * [mcp3201_sampler_start](#) ([mcp3201_handle_t](#) mcp_handle, [QueueHandle_t](#) recv_↔ queue, const [size_t](#) buffer_size, const [size_t](#) f_sample)
Starts the MCP3201 Sampler, which samples continuously and puts the samples into a queue.
- void [mcp3201_sampler_stop](#) ([mcp3201_sampler_t](#) *sampler)
Stops the MCP3201 Sampler.

4.15.1 Detailed Description

Author

@s-grundner

Version

0.1

Date

2022-12-24

Copyright

Copyright (c) 2022

Definition in file [mcp3201.h](#).

4.15.2 Macro Definition Documentation

4.15.2.1 ADC_CLK

```
#define ADC_CLK SPI_MASTER_FREQ_8M
```

Definition at line 24 of file [mcp3201.h](#).

4.15.3 Typedef Documentation

4.15.3.1 mcp3201_handle_t

```
typedef struct mcp3201_context_t* mcp3201_handle_t
```

Definition at line 46 of file [mcp3201.h](#).

4.15.4 Function Documentation

4.15.4.1 mcp3201_exit()

```
esp_err_t mcp3201_exit (
    mcp3201_handle_t mcp_handle )
```

Exits the MCP3201 ADC device and frees all resources.

Parameters

<i>mcp_handle</i>	MCP3201 Device to exit
-------------------	------------------------

Returns

ESP_OK on success

Definition at line 105 of file [mcp3201.c](#).

4.15.4.2 mcp3201_init()

```
esp_err_t mcp3201_init (
    mcp3201_handle_t * out_handle,
    const mcp3201_config_t * cfg )
```

Initializes the MCP3201 ADC device.

Parameters

<i>out_handle</i>	MCP3201 Handler to store initialization data
<i>cfg</i>	MCP3201 Configuration

Returns

4.15.4.3 mcp3201_read()

```
esp_err_t mcp3201_read (
    mcp3201_handle_t handle,
    uint16_t * out_value )
```

Reads a single value from the MCP3201 ADC.

Parameters

	<i>handle</i>	MCP3201 Device
out	<i>out_value</i>	Value

Returns

ESP_OK on success

4.15.4.4 mcp3201_sampler_start()

```
mcp3201_sampler_t * mcp3201_sampler_start (
    mcp3201_handle_t mcp_handle,
    QueueHandle_t recv_queue,
    const size_t buffer_size,
    const size_t f_sample )
```

Starts the MCP3201 Sampler, which samples continuously and puts the samples into a queue.

Parameters

<i>mcp_handle</i>	MCP3201 Device Handler
<i>recv_queue</i>	queue to send the samples into
<i>buffer_size</i>	size of the audio buffer
<i>f_sample</i>	Sample rate

Returns

mcp3201_sampler_t* Sampler Handler

Definition at line 84 of file [mcp3201_sampler.c](#).

4.15.4.5 mcp3201_sampler_stop()

```
void mcp3201_sampler_stop (
    mcp3201_sampler_t * sampler )
```

Stops the MCP3201 Sampler.

Parameters

<i>sampler</i>	Sampler Handler to stop
----------------	-------------------------

Definition at line 110 of file [mcp3201_sampler.c](#).

4.16 mcp3201.h

[Go to the documentation of this file.](#)

```
00001
00012 #ifndef MCP3201_DRIVER_H
00013 #define MCP3201_DRIVER_H
00014
00015 #include <stdio.h>
00016 #include "freertos/FreeRTOS.h"
00017 #include "freertos/task.h"
00018 #include "freertos/queue.h"
00019 #include "driver/gpio.h"
00020 #include "driver/spi_master.h"
00021 #include "hal/spi_types.h"
00022 #include "esp_log.h"
00023
00024 #define ADC_CLK SPI_MASTER_FREQ_8M
00025
00036 typedef struct
00037 {
00038     spi_host_device_t host;
00039     gpio_num_t cs_io;
00040     gpio_num_t miso_io;
00041     gpio_num_t mosi_io;
00042     int dma_chan;
00043 } mcp3201_config_t;
00044
00046 typedef struct mcp3201_context_t *mcp3201_handle_t;
00047
00058 typedef struct
00059 {
00060     mcp3201_handle_t mcp_handle;
00061     QueueHandle_t dma_queue;
00062     QueueHandle_t dsp_queue;
00063     size_t *buffer;
00064     size_t buffer_pos;
00065     size_t buffer_size;
00066 } mcp3201_sampler_t;
00067
00075 esp_err_t mcp3201_init(mcp3201_handle_t *out_handle, const mcp3201_config_t *cfg);
00076
00083 esp_err_t mcp3201_exit(mcp3201_handle_t mcp_handle);
00084
00092 esp_err_t mcp3201_read(mcp3201_handle_t handle, uint16_t *out_value);
00093
00103 mcp3201_sampler_t *mcp3201_sampler_start(mcp3201_handle_t mcp_handle, QueueHandle_t recv_queue, const
    size_t buffer_size, const size_t f_sample);
00104
00110 void mcp3201_sampler_stop(mcp3201_sampler_t *sampler);
00111
00112 #endif // MCP3201_H
```

4.17 lib/mcp3201/mcp3201_sampler.c File Reference

MCP3201 Sampler.

```
#include "mcp3201.h"
```

Data Structures

- struct [mcp3201_context_t](#)
MCP3201 Context struct for internal use.

Macros

- #define [RESAMPLE_DENOMINATOR](#) 2
- #define [READER_TIMEOUT_MS](#) 10
- #define [READER_TIMEOUT_TICKS](#) ([READER_TIMEOUT_MS](#) / portTICK_PERIOD_MS)

Typedefs

- typedef struct [mcp3201_context_t](#) [mcp3201_context_t](#)

Functions

- [mcp3201_sampler_t](#) * [mcp3201_sampler_start](#) ([mcp3201_handle_t](#) mcp_handle, QueueHandle_t recv_↔ queue, const size_t buffer_size, const size_t f_sample)
Starts the MCP3201 Sampler, which samples continuously and puts the samples into a queue.
- void [mcp3201_sampler_stop](#) ([mcp3201_sampler_t](#) *sampler)
Stops the MCP3201 Sampler.

4.17.1 Detailed Description

MCP3201 Sampler.

Author

@s-grundner

Version

0.1

Date

2023-03-26

Copyright

Copyright (c) 2023

Definition in file [mcp3201_sampler.c](#).

4.17.2 Macro Definition Documentation

4.17.2.1 `READER_TIMEOUT_MS`

```
#define READER_TIMEOUT_MS 10
```

Definition at line 14 of file [mcp3201_sampler.c](#).

4.17.2.2 `READER_TIMEOUT_TICKS`

```
#define READER_TIMEOUT_TICKS (READER_TIMEOUT_MS / portTICK_PERIOD_MS)
```

Definition at line 15 of file [mcp3201_sampler.c](#).

4.17.2.3 `RESAMPLE_DENOMINATOR`

```
#define RESAMPLE_DENOMINATOR 2
```

Definition at line 13 of file [mcp3201_sampler.c](#).

4.17.3 Typedef Documentation

4.17.3.1 `mcp3201_context_t`

```
typedef struct mcp3201_context_t mcp3201_context_t
```

Definition at line 24 of file [mcp3201_sampler.c](#).

4.17.4 Function Documentation

4.17.4.1 `mcp3201_sampler_start()`

```
mcp3201_sampler_t * mcp3201_sampler_start (
    mcp3201_handle_t mcp_handle,
    QueueHandle_t recv_queue,
    const size_t buffer_size,
    const size_t f_sample )
```

Starts the MCP3201 Sampler, which samples continuously and puts the samples into a queue.

Parameters

<i>mcp_handle</i>	MCP3201 Device Handler
<i>recv_queue</i>	queue to send the samples into
<i>buffer_size</i>	size of the audio buffer
<i>f_sample</i>	Sample rate

Returns

mcp3201_sampler_t* Sampler Handler

Definition at line 84 of file [mcp3201_sampler.c](#).

4.17.4.2 mcp3201_sampler_stop()

```
void mcp3201_sampler_stop (
    mcp3201_sampler_t * sampler )
```

Stops the MCP3201 Sampler.

Parameters

<i>sampler</i>	Sampler Handler to stop
----------------	-------------------------

Definition at line 110 of file [mcp3201_sampler.c](#).

4.18 mcp3201_sampler.c

[Go to the documentation of this file.](#)

```
00001
00011 #include "mcp3201.h"
00012
00013 #define RESAMPLE_DENOMINATOR 2
00014 #define READER_TIMEOUT_MS 10
00015 #define READER_TIMEOUT_TICKS (READER_TIMEOUT_MS / portTICK_PERIOD_MS)
00016
00017 static const char *TAG = "MCP3201_SAMPLER";
00018 struct mcp3201_context_t
00019 {
00020     mcp3201_config_t cfg;
00021     spi_device_handle_t spi;
00022     spi_transaction_t *ongoing_transaction;
00023 };
00024 typedef struct mcp3201_context_t mcp3201_context_t;
00025
00026 static TaskHandle_t sampler_task_handle;
00027 static TaskHandle_t reader_task_handle;
00028
00029 static void IRAM_ATTR sampler_task(void *arg)
00030 {
00031     mcp3201_sampler_t *sampler = (mcp3201_sampler_t *)arg;
00032     for (;;)
00033     {
00034         spi_event_t evt;
00035         if (xQueueReceive(sampler->dma_queue, &evt, portMAX_DELAY) == pdTRUE)
00036         {
00037             if (evt == SPI_EV_SEND_DMA_READY)
```

```

00038         {
00039             size_t bytes_read = 0;
00040             do
00041             {
00042                 // fill audio buffer
00043                 size_t bytes_to_read = RESAMPLE_DENOMINATOR * (sampler->buffer_size -
sampler->buffer_pos);
00044                 void *buffer_position = (void *) (sampler->buffer + sampler->buffer_pos);
00045
00046                 // read data from spi
00047                 spi_transaction_t t = {
00048                     .user = (void *)sampler->mcp_handle,
00049                     .length = bytes_to_read,
00050                     .rx_buffer = buffer_position,
00051                 };
00052                 sampler->mcp_handle->ongoing_transaction = &t;
00053                 spi_device_get_trans_result(
00054                     sampler->mcp_handle->spi,
00055                     (spi_transaction_t *) (sampler->mcp_handle->ongoing_transaction),
00056                     READER_TIMEOUT_TICKS);
00057                 bytes_read = t.rxlenght;
00058
00059                 sampler->buffer_pos += bytes_read / RESAMPLE_DENOMINATOR;
00060
00061                 if (sampler->buffer_pos == sampler->buffer_size)
00062                 {
00063                     // send data to DSP queue
00064                     sampler->buffer_pos = 0;
00065                     xQueueSend(sampler->dsp_queue, &sampler->buffer, portMAX_DELAY);
00066                 }
00067                 } while (bytes_read > 0);
00068             }
00069         }
00070     }
00071 }
00072
00073 static void IRAM_ATTR reader_task(void *arg)
00074 {
00075     mcp3201_sampler_t *sampler = (mcp3201_sampler_t *)arg;
00076     for (;;)
00077     {
00078         spi_device_queue_trans(sampler->mcp_handle->spi, sampler->mcp_handle->ongoing_transaction,
portMAX_DELAY);
00079         spi_event_t evt = SPI_EV_SEND_DMA_READY;
00080         xQueueSendFromISR(sampler->dma_queue, &evt, NULL);
00081     }
00082 }
00083
00084 mcp3201_sampler_t *mcp3201_sampler_start(mcp3201_handle_t mcp_handle, QueueHandle_t recv_queue, const
size_t buffer_size, const size_t f_sample)
00085 {
00086     if (mcp_handle == NULL || recv_queue == NULL || buffer_size == 0 || f_sample == 0)
00087     {
00088         ESP_LOGE(TAG, "Invalid arguments");
00089         return NULL;
00090     }
00091
00092     mcp3201_sampler_t *sampler = (mcp3201_sampler_t *)malloc(sizeof(mcp3201_sampler_t));
00093     *sampler = (mcp3201_sampler_t){
00094         .mcp_handle = mcp_handle,
00095         .buffer_size = buffer_size,
00096         .buffer = (size_t *)malloc(sizeof(size_t) * buffer_size),
00097         .buffer_pos = 0,
00098         .dma_queue = xQueueCreate(4, sizeof(spi_event_t)),
00099         .dsp_queue = recv_queue;
00100
00101         while (spi_device_acquire_bus(mcp_handle->spi, portMAX_DELAY) != ESP_OK)
00102             ESP_LOGE(TAG, "Failed to acquire bus\n retrying...");
00103
00104         xTaskCreatePinnedToCore(sampler_task, "sampler_task", RESAMPLE_DENOMINATOR * buffer_size, sampler,
5, &sampler_task_handle, 0);
00105         xTaskCreatePinnedToCore(reader_task, "reader_task", 2048, sampler, 5, &reader_task_handle, 0);
00106
00107         return sampler;
00108     }
00109
00110 void mcp3201_sampler_stop(mcp3201_sampler_t *sampler)
00111 {
00112     vQueueDelete(sampler->dma_queue);
00113     vQueueDelete(sampler->dsp_queue);
00114
00115     spi_device_release_bus(sampler->mcp_handle->spi);
00116
00117     vTaskDelete(sampler_task_handle);
00118     vTaskDelete(reader_task_handle);
00119
00120     free(sampler->buffer);

```

```
00121     free(sampler);  
00122 }
```

4.19 lib/midi/midi.c File Reference

MIDI UART Driver Source for ESP32.

```
#include "midi.h"
```

Data Structures

- struct [midi_context_t](#)
MIDI Context (internal!! not to be accessed externally, use midi_handle_t instead)

Macros

- #define [MIDI_BYTE_SIZE_DEFAULT](#) 3
- #define [MIDI_BYTE_SIZE_SHORT](#) 2

Typedefs

- typedef struct [midi_context_t](#) [midi_context_t](#)

Functions

- esp_err_t [midi_init](#) ([midi_context_t](#) **out_ctx, [midi_config_t](#) *out_cfg)
- esp_err_t [midi_exit](#) ([midi_handle_t](#) midi_handle)
Exits MIDI and frees all resources.
- esp_err_t [midi_write](#) ([midi_handle_t](#) handle, [midi_message_t](#) *msg)
Writes MIDI Message to UART.
- esp_err_t [midi_read](#) ([midi_handle_t](#) midi_handle, [midi_message_t](#) *msg, TickType_t timeout)
Reads MIDI Message from UART.

4.19.1 Detailed Description

MIDI UART Driver Source for ESP32.

Author

@s-grundner

Version

0.1

Date

2022-12-23

Copyright

Copyright (c) 2022

Definition in file [midi.c](#).

4.19.2 Macro Definition Documentation

4.19.2.1 MIDI_BYTE_SIZE_DEFAULT

```
#define MIDI_BYTE_SIZE_DEFAULT 3
```

Definition at line 13 of file [midi.c](#).

4.19.2.2 MIDI_BYTE_SIZE_SHORT

```
#define MIDI_BYTE_SIZE_SHORT 2
```

Definition at line 14 of file [midi.c](#).

4.19.3 Typedef Documentation

4.19.3.1 midi_context_t

```
typedef struct midi\_context\_t midi\_context\_t
```

4.19.4 Function Documentation

4.19.4.1 midi_exit()

```
esp_err_t midi_exit (  
    midi\_handle\_t midi_handle )
```

Exits MIDI and frees all resources.

Parameters

<i>midi_handle</i>	MIDI Handle to be freed
--------------------	-------------------------

Returns

[esp_err_t](#)

Definition at line 80 of file [midi.c](#).

4.19.4.2 midi_init()

```
esp_err_t midi_init (
    midi_context_t ** out_ctx,
    midi_config_t * out_cfg )
```

Definition at line 33 of file [midi.c](#).

4.19.4.3 midi_read()

```
esp_err_t midi_read (
    midi_handle_t midi_handle,
    midi_message_t * msg,
    TickType_t timeout )
```

Reads MIDI Message from UART.

Parameters

<i>midi_handle</i>	MIDI Handle to pass parameters
<i>msg</i>	MIDI Message to be read

Returns

`esp_err_t`

Definition at line 133 of file [midi.c](#).

4.19.4.4 midi_write()

```
esp_err_t midi_write (
    midi_handle_t midi_handle,
    midi_message_t * msg )
```

Writes MIDI Message to UART.

Parameters

<i>midi_handle</i>	MIDI Handle to pass parameters
<i>msg</i>	MIDI Message to be sent

Returns

esp_err_t

Definition at line 92 of file [midi.c](#).

4.20 midi.c

[Go to the documentation of this file.](#)

```

00001
00011 #include "midi.h"
00012
00013 #define MIDI_BYTE_SIZE_DEFAULT 3
00014 #define MIDI_BYTE_SIZE_SHORT 2
00015
00016 static const char *TAG = "MIDI";
00017 static const char *MIDI_MON = "MIDI MONITOR";
00018
00024 typedef struct midi_context_t
00025 {
00026     midi_config_t cfg;
00027 } midi_context_t;
00028
00029 // -----
00030 // MIDI CONFIG
00031 // -----
00032
00033 esp_err_t midi_init(midi_context_t **out_ctx, midi_config_t *out_cfg)
00034 {
00035     // Allocate memory for context
00036     midi_context_t *ctx = (midi_context_t *)malloc(sizeof(midi_context_t));
00037     if (!ctx)
00038         return ESP_ERR_NO_MEM;
00039
00040     *ctx = (midi_context_t){
00041         .cfg = *out_cfg;
00042
00043     ESP_LOGI(TAG, "Initializing MIDI on %d (rx:%d, tx:%d) with %d baud...", ctx->cfg.uart_num,
00044             ctx->cfg.rx_io, ctx->cfg.tx_io, ctx->cfg.baudrate);
00045
00046     // Configure UART
00047     gpio_config_t rx_pin_config = {
00048         .pin_bit_mask = (1ULL < ctx->cfg.rx_io),
00049         .mode = GPIO_MODE_INPUT,
00050         .pull_up_en = GPIO_PULLUP_DISABLE,
00051         .pull_down_en = GPIO_PULLDOWN_DISABLE,
00052         .intr_type = GPIO_INTR_DISABLE};
00053
00054     gpio_config_t tx_pin_config = {
00055         .pin_bit_mask = (1ULL < ctx->cfg.tx_io),
00056         .mode = GPIO_MODE_OUTPUT,
00057         .pull_up_en = GPIO_PULLUP_DISABLE,
00058         .pull_down_en = GPIO_PULLDOWN_DISABLE,
00059         .intr_type = GPIO_INTR_DISABLE};
00060
00061     ESP_ERROR_CHECK(gpio_config(&rx_pin_config));
00062     ESP_ERROR_CHECK(gpio_config(&tx_pin_config));
00063
00064     uart_config_t uart_config = {
00065         .baud_rate = ctx->cfg.baudrate,
00066         .data_bits = UART_DATA_8_BITS,
00067         .parity = UART_PARITY_DISABLE,
00068         .stop_bits = UART_STOP_BITS_1,
00069         .flow_ctrl = UART_HW_FLOWCTRL_DISABLE,
00070         .source_clk = UART_SCLK_APB};
00071
00072     ESP_ERROR_CHECK(uart_param_config(ctx->cfg.uart_num, &uart_config));
00073     ESP_ERROR_CHECK(uart_set_pin(ctx->cfg.uart_num, ctx->cfg.tx_io, ctx->cfg.rx_io,
00074             UART_PIN_NO_CHANGE, UART_PIN_NO_CHANGE));
00075     ESP_ERROR_CHECK(uart_driver_install(ctx->cfg.uart_num, 1024 * 2, 1024 * 2, 0, NULL, 0));
00076
00077     // Pass configured context to outer parameters
00078     *out_ctx = ctx;
00079     return ESP_OK;
00080 }
00081
00082 esp_err_t midi_exit(midi_handle_t midi_handle)
00083 {
00084     esp_err_t err = ESP_OK;

```

```

00083     err = uart_driver_delete(midi_handle->cfg.uart_num);
00084     free(midi_handle);
00085     return err;
00086 }
00087
00088 // -----
00089 // MIDI TRANSMISSIONS
00090 // -----
00091
00092 esp_err_t midi_write(midi_handle_t handle, midi_message_t *msg)
00093 {
00094     int len = 0;
00095     const char data[] = {msg->status | msg->channel, msg->param1, msg->param2};
00096
00097     // switch status to determine message length
00098     switch (msg->status)
00099     {
00100     case MIDI_STATUS_NOTE_OFF:
00101     case MIDI_STATUS_NOTE_ON:
00102     case MIDI_STATUS_CONTROL_CHANGE:
00103     case MIDI_STATUS_PITCH_BEND:
00104     case MIDI_STATUS_POLYPHONIC_KEY_PRESSURE:
00105         len = uart_write_bytes(handle->cfg.uart_num, data, MIDI_BYTE_SIZE_DEFAULT);
00106         break;
00107     case MIDI_STATUS_PROGRAM_CHANGE:
00108     case MIDI_STATUS_CHANNEL_PRESSURE:
00109         len = uart_write_bytes(handle->cfg.uart_num, data, MIDI_BYTE_SIZE_SHORT);
00110         break;
00111     default:
00112         ESP_LOGE(TAG, "midi_send: invalid status: %02X", msg->status);
00113         return ESP_ERR_INVALID_ARG;
00114         break;
00115     }
00116     switch (len)
00117     {
00118     case -1:
00119         ESP_LOGE(TAG, "uart_write_bytes failed");
00120         return ESP_FAIL;
00121         break;
00122     case 0:
00123         ESP_LOGE(TAG, "uart_write_bytes timed out");
00124         return ESP_ERR_TIMEOUT;
00125         break;
00126     default:
00127         // ESP_LOGI(MIDI_MON, "Status: %02X\tChannel: %02X\tData: %02X %02X\tLength:%d",
msg->status, msg->channel, msg->param1, msg->param2, len);
00128         break;
00129     }
00130     return ESP_OK;
00131 }
00132
00133 esp_err_t midi_read(midi_handle_t midi_handle, midi_message_t *msg, TickType_t timeout)
00134 {
00135     char data[3];
00136     int len = uart_read_bytes(midi_handle->cfg.uart_num, (uint8_t *)data, 3, timeout);
00137     switch (len)
00138     {
00139     case -1:
00140         ESP_LOGE(TAG, "uart_read_bytes failed");
00141         return ESP_FAIL;
00142         break;
00143     case 0:
00144         ESP_LOGE(TAG, "uart_read_bytes timeout");
00145         return ESP_ERR_TIMEOUT;
00146         break;
00147     case 2:
00148         msg->status = data[0] & 0xF0;
00149         msg->channel = data[0] & 0x0F;
00150         msg->param1 = data[1];
00151         msg->param2 = 0;
00152         break;
00153     case 3:
00154         msg->status = data[0] & 0xF0;
00155         msg->channel = data[0] & 0x0F;
00156         msg->param1 = data[1];
00157         msg->param2 = data[2];
00158         break;
00159     default:
00160         ESP_LOGE(TAG, "uart_read_bytes invalid message");
00161         return ESP_ERR_INVALID_ARG;
00162         break;
00163     }
00164     return ESP_OK;
00165 }

```

4.21 lib/midi/midi.h File Reference

MIDI UART Driver for ESP32.

```
#include <stdio.h>
#include <stdint.h>
#include "driver/gpio.h"
#include "driver/uart.h"
#include "esp_log.h"
```

Data Structures

- struct [midi_message_t](#)
MIDI Message.
- struct [midi_config_t](#)
MIDI UART Configuration.

Macros

- #define [MIDI_LOG_LEVEL](#) ESP_LOG_ERROR
- #define [MIDI_BYTE_SIZE_DEFAULT](#) 3
- #define [MIDI_BYTE_SIZE_SHORT](#) 2
- #define [MIDI_PITCH_BEND_MIN](#) (0)
- #define [MIDI_PITCH_BEND_MAX](#) (16383)
- #define [MIDI_PITCH_BEND_CENTER](#) (8192)

Typedefs

- typedef struct [midi_context_t](#) * [midi_handle_t](#)

Enumerations

- enum [midi_status_t](#) {
[MIDI_STATUS_NOTE_OFF](#) = 0x80 , [MIDI_STATUS_NOTE_ON](#) = 0x90 , [MIDI_STATUS_POLYPHONIC_KEY_PRESSURE](#) = 0xA0 , [MIDI_STATUS_CONTROL_CHANGE](#) = 0xB0 ,
[MIDI_STATUS_PROGRAM_CHANGE](#) = 0xC0 , [MIDI_STATUS_CHANNEL_PRESSURE](#) = 0xD0 ,
[MIDI_STATUS_PITCH_BEND](#) = 0xE0 }
MIDI Status Bytes.

Functions

- esp_err_t [midi_init](#) ([midi_handle_t](#) *out_handle, [midi_config_t](#) *out_cfg)
initializes MIDI and allocates driver resources
- esp_err_t [midi_exit](#) ([midi_handle_t](#) midi_handle)
Exits MIDI and frees all resources.
- esp_err_t [midi_write](#) ([midi_handle_t](#) midi_handle, [midi_message_t](#) *msg)
Writes MIDI Message to UART.
- esp_err_t [midi_read](#) ([midi_handle_t](#) midi_handle, [midi_message_t](#) *msg, TickType_t timeout)
Reads MIDI Message from UART.
- [midi_message_t note_off](#) (uint8_t channel, uint8_t key_num, uint8_t velocity)
- [midi_message_t note_on](#) (uint8_t channel, uint8_t key_num, uint8_t velocity)
- [midi_message_t poly_key_pressure](#) (uint8_t channel, uint8_t key_num, uint8_t value)
- [midi_message_t ctrl_change](#) (uint8_t channel, uint8_t controller_num, uint8_t value)
- [midi_message_t prg_change](#) (uint8_t channel, uint8_t program)
- [midi_message_t channel_pressure](#) (uint8_t channel, uint8_t value)
- [midi_message_t pitch_bend](#) (uint8_t channel, uint16_t value)

4.21.1 Detailed Description

MIDI UART Driver for ESP32.

Author

@s-grundner

Version

0.1

Date

2022-12-23

Copyright

Copyright (c) 2022

Definition in file [midi.h](#).

4.21.2 Macro Definition Documentation

4.21.2.1 MIDI_BYTE_SIZE_DEFAULT

```
#define MIDI_BYTE_SIZE_DEFAULT 3
```

Definition at line 23 of file [midi.h](#).

4.21.2.2 MIDI_BYTE_SIZE_SHORT

```
#define MIDI_BYTE_SIZE_SHORT 2
```

Definition at line 24 of file [midi.h](#).

4.21.2.3 MIDI_LOG_LEVEL

```
#define MIDI_LOG_LEVEL ESP_LOG_ERROR
```

Definition at line 21 of file [midi.h](#).

4.21.2.4 MIDI_PITCH_BEND_CENTER

```
#define MIDI_PITCH_BEND_CENTER (8192)
```

Definition at line 28 of file [midi.h](#).

4.21.2.5 MIDI_PITCH_BEND_MAX

```
#define MIDI_PITCH_BEND_MAX (16383)
```

Definition at line 27 of file [midi.h](#).

4.21.2.6 MIDI_PITCH_BEND_MIN

```
#define MIDI_PITCH_BEND_MIN (0)
```

Definition at line 26 of file [midi.h](#).

4.21.3 Typedef Documentation

4.21.3.1 midi_handle_t

```
typedef struct midi\_context\_t* midi\_handle\_t
```

Definition at line 85 of file [midi.h](#).

4.21.4 Enumeration Type Documentation

4.21.4.1 midi_status_t

```
enum midi\_status\_t
```

MIDI Status Bytes.

Parameters

<i>MIDI_STATUS_NOTE_OFF</i>	0x80, requires param2
<i>MIDI_STATUS_NOTE_ON</i>	0x90, requires param2
<i>MIDI_STATUS_POLYPHONIC_KEY_PRESSURE</i>	0xA0, param2 is not used
<i>MIDI_STATUS_CONTROL_CHANGE</i>	0xB0, requires param2
<i>MIDI_STATUS_PROGRAM_CHANGE</i>	0xC0, param2 is not used
<i>MIDI_STATUS_CHANNEL_PRESSURE</i>	0xD0, param2 is not used

Enumerator

MIDI_STATUS_NOTE_OFF	
MIDI_STATUS_NOTE_ON	
MIDI_STATUS_POLYPHONIC_KEY_PRESSURE	
MIDI_STATUS_CONTROL_CHANGE	
MIDI_STATUS_PROGRAM_CHANGE	
MIDI_STATUS_CHANNEL_PRESSURE	
MIDI_STATUS_PITCH_BEND	

Definition at line 41 of file [midi.h](#).

4.21.5 Function Documentation

4.21.5.1 channel_pressure()

```
midi_message_t channel_pressure (
    uint8_t channel,
    uint8_t value )
```

Definition at line 67 of file [midi_utils.c](#).

4.21.5.2 ctrl_change()

```
midi_message_t ctrl_change (
    uint8_t channel,
    uint8_t controller_num,
    uint8_t value )
```

Definition at line 47 of file [midi_utils.c](#).

4.21.5.3 midi_exit()

```
esp_err_t midi_exit (
    midi_handle_t midi_handle )
```

Exits MIDI and frees all resources.

Parameters

<i>midi_handle</i>	MIDI Handle to be freed
--------------------	-------------------------

Returns

esp_err_t

Definition at line 80 of file [midi.c](#).

4.21.5.4 midi_init()

```
esp_err_t midi_init (
    midi_handle_t * out_handle,
    midi_config_t * out_cfg )
```

initializes MIDI and allocates driver resources

Parameters

out	<i>out_handle</i>	MIDI Handle to be initialized
out	<i>out_cfg</i>	MIDI Configuration

Returns

esp_err_t

4.21.5.5 midi_read()

```
esp_err_t midi_read (
    midi_handle_t midi_handle,
    midi_message_t * msg,
    TickType_t timeout )
```

Reads MIDI Message from UART.

Parameters

<i>midi_handle</i>	MIDI Handle to pass parameters
<i>msg</i>	MIDI Message to be read

Returns

esp_err_t

Definition at line 133 of file [midi.c](#).

4.21.5.6 midi_write()

```
esp_err_t midi_write (
    midi_handle_t midi_handle,
    midi_message_t * msg )
```

Writes MIDI Message to UART.

Parameters

<i>midi_handle</i>	MIDI Handle to pass parameters
<i>msg</i>	MIDI Message to be sent

Returns

esp_err_t

Definition at line 92 of file [midi.c](#).

4.21.5.7 note_off()

```
midi_message_t note_off (
    uint8_t channel,
    uint8_t key_num,
    uint8_t velocity )
```

Definition at line 17 of file [midi_utils.c](#).

4.21.5.8 note_on()

```
midi_message_t note_on (
    uint8_t channel,
    uint8_t key_num,
    uint8_t velocity )
```

Definition at line 27 of file [midi_utils.c](#).

4.21.5.9 pitch_bend()

```
midi_message_t pitch_bend (
    uint8_t channel,
    uint16_t value )
```

Definition at line 77 of file [midi_utils.c](#).

4.21.5.10 poly_key_pressure()

```

midi_message_t poly_key_pressure (
    uint8_t channel,
    uint8_t key_num,
    uint8_t value )

```

Definition at line 37 of file [midi_utils.c](#).

4.21.5.11 prg_change()

```

midi_message_t prg_change (
    uint8_t channel,
    uint8_t program )

```

Definition at line 57 of file [midi_utils.c](#).

4.22 midi.h

[Go to the documentation of this file.](#)

```

00001
00012 #ifndef MIDI_DRIVER_H
00013 #define MIDI_DRIVER_H
00014
00015 #include <stdio.h>
00016 #include <stdint.h>
00017 #include "driver/gpio.h"
00018 #include "driver/uart.h"
00019 #include "esp_log.h"
00020
00021 #define MIDI_LOG_LEVEL ESP_LOG_ERROR
00022
00023 #define MIDI_BYTE_SIZE_DEFAULT 3
00024 #define MIDI_BYTE_SIZE_SHORT 2
00025
00026 #define MIDI_PITCH_BEND_MIN (0)
00027 #define MIDI_PITCH_BEND_MAX (16383)
00028 #define MIDI_PITCH_BEND_CENTER (8192)
00029
00041 typedef enum
00042 {
00043     MIDI_STATUS_NOTE_OFF = 0x80,
00044     MIDI_STATUS_NOTE_ON = 0x90,
00045     MIDI_STATUS_POLYPHONIC_KEY_PRESSURE = 0xA0,
00046     MIDI_STATUS_CONTROL_CHANGE = 0xB0,
00047     MIDI_STATUS_PROGRAM_CHANGE = 0xC0,
00048     MIDI_STATUS_CHANNEL_PRESSURE = 0xD0,
00049     MIDI_STATUS_PITCH_BEND = 0xE0,
00050 } midi_status_t;
00051
00060 typedef struct
00061 {
00062     uint8_t param1;
00063     midi_status_t status;
00064     uint8_t channel;
00065     uint8_t param2;
00066 } midi_message_t;
00067
00076 typedef struct
00077 {
00078     uart_port_t uart_num;
00079     uint baudrate;
00080     gpio_num_t rx_io;
00081     gpio_num_t tx_io;
00082 } midi_config_t;
00083
00085 typedef struct midi_context_t *midi_handle_t;

```

```

00086
00094 esp_err_t midi_init(midi_handle_t *out_handle, midi_config_t *out_cfg);
00095
00102 esp_err_t midi_exit(midi_handle_t midi_handle);
00103
00111 esp_err_t midi_write(midi_handle_t midi_handle, midi_message_t *msg);
00112
00120 esp_err_t midi_read(midi_handle_t midi_handle, midi_message_t *msg, TickType_t timeout);
00121
00122 // functions to configure midi messages
00123 midi_message_t note_off(uint8_t channel, uint8_t key_num, uint8_t velocity);
00124 midi_message_t note_on(uint8_t channel, uint8_t key_num, uint8_t velocity);
00125 midi_message_t poly_key_pressure(uint8_t channel, uint8_t key_num, uint8_t value);
00126 midi_message_t ctrl_change(uint8_t channel, uint8_t controller_num, uint8_t value);
00127 midi_message_t prg_change(uint8_t channel, uint8_t program);
00128 midi_message_t channel_pressure(uint8_t channel, uint8_t value);
00129 midi_message_t pitch_bend(uint8_t channel, uint16_t value);
00130
00131 #endif // MIDI_DRIVER_H

```

4.23 lib/midi/midi_utils.c File Reference

MIDI Message Utilities.

```
#include "midi.h"
```

Functions

- [midi_message_t note_off](#) (uint8_t channel, uint8_t key_num, uint8_t velocity)
- [midi_message_t note_on](#) (uint8_t channel, uint8_t key_num, uint8_t velocity)
- [midi_message_t poly_key_pressure](#) (uint8_t channel, uint8_t key_num, uint8_t value)
- [midi_message_t ctrl_change](#) (uint8_t channel, uint8_t controller_num, uint8_t value)
- [midi_message_t prg_change](#) (uint8_t channel, uint8_t program)
- [midi_message_t channel_pressure](#) (uint8_t channel, uint8_t value)
- [midi_message_t pitch_bend](#) (uint8_t channel, uint16_t value)

4.23.1 Detailed Description

MIDI Message Utilities.

Author

@s-grundner

Version

0.1

Date

2023-03-26

Copyright

Copyright (c) 2023

Definition in file [midi_utils.c](#).

4.23.2 Function Documentation

4.23.2.1 `channel_pressure()`

```
midi_message_t channel_pressure (
    uint8_t channel,
    uint8_t value )
```

Definition at line 67 of file [midi_utils.c](#).

4.23.2.2 `ctrl_change()`

```
midi_message_t ctrl_change (
    uint8_t channel,
    uint8_t controller_num,
    uint8_t value )
```

Definition at line 47 of file [midi_utils.c](#).

4.23.2.3 `note_off()`

```
midi_message_t note_off (
    uint8_t channel,
    uint8_t key_num,
    uint8_t velocity )
```

Definition at line 17 of file [midi_utils.c](#).

4.23.2.4 `note_on()`

```
midi_message_t note_on (
    uint8_t channel,
    uint8_t key_num,
    uint8_t velocity )
```

Definition at line 27 of file [midi_utils.c](#).

4.23.2.5 pitch_bend()

```
midi_message_t pitch_bend (
    uint8_t channel,
    uint16_t value )
```

Definition at line 77 of file [midi_utils.c](#).

4.23.2.6 poly_key_pressure()

```
midi_message_t poly_key_pressure (
    uint8_t channel,
    uint8_t key_num,
    uint8_t value )
```

Definition at line 37 of file [midi_utils.c](#).

4.23.2.7 prg_change()

```
midi_message_t prg_change (
    uint8_t channel,
    uint8_t program )
```

Definition at line 57 of file [midi_utils.c](#).

4.24 midi_utils.c

[Go to the documentation of this file.](#)

```
00001
00011 #include "midi.h"
00012
00013 // -----
00014 // MIDI UTILS
00015 // -----
00016
00017 midi_message_t note_off(uint8_t channel, uint8_t key_num, uint8_t velocity)
00018 {
00019     midi_message_t msg = {
00020         .status = MIDI_STATUS_NOTE_OFF,
00021         .channel = channel,
00022         .param1 = key_num,
00023         .param2 = velocity};
00024     return msg;
00025 }
00026
00027 midi_message_t note_on(uint8_t channel, uint8_t key_num, uint8_t velocity)
00028 {
00029     midi_message_t msg = {
00030         .status = MIDI_STATUS_NOTE_ON,
00031         .channel = channel,
00032         .param1 = key_num,
00033         .param2 = velocity};
00034     return msg;
00035 }
00036
00037 midi_message_t poly_key_pressure(uint8_t channel, uint8_t key_num, uint8_t value)
00038 {
00039     midi_message_t msg = {
```

```

00040         .status = MIDI_STATUS_POLYPHONIC_KEY_PRESSURE,
00041         .channel = channel,
00042         .param1 = key_num,
00043         .param2 = value};
00044     return msg;
00045 }
00046
00047 midi_message_t ctrl_change(uint8_t channel, uint8_t controller_num, uint8_t value)
00048 {
00049     midi_message_t msg = {
00050         .status = MIDI_STATUS_CONTROL_CHANGE,
00051         .channel = channel,
00052         .param1 = controller_num,
00053         .param2 = value};
00054     return msg;
00055 }
00056
00057 midi_message_t prg_change(uint8_t channel, uint8_t program)
00058 {
00059     midi_message_t msg = {
00060         .status = MIDI_STATUS_PROGRAM_CHANGE,
00061         .channel = channel,
00062         .param1 = program,
00063         .param2 = 0};
00064     return msg;
00065 }
00066
00067 midi_message_t channel_pressure(uint8_t channel, uint8_t value)
00068 {
00069     midi_message_t msg = {
00070         .status = MIDI_STATUS_CHANNEL_PRESSURE,
00071         .channel = channel,
00072         .param1 = value,
00073         .param2 = 0};
00074     return msg;
00075 }
00076
00077 midi_message_t pitch_bend(uint8_t channel, uint16_t value)
00078 {
00079     midi_message_t msg = {
00080         .status = MIDI_STATUS_PITCH_BEND,
00081         .channel = channel,
00082         .param1 = value & 0x7F,
00083         .param2 = (value >> 7) & 0x7F};
00084     return msg;
00085 }

```

4.25 src/gitcon.c File Reference

Gitcon Driver Source.

```

#include "gitcon.h"
#include "processed-data.h"

```

Macros

- #define [FLOAT_TO_UINT16\(x\)](#) ((uint16_t)((x)*32767.0f))
- #define [UINT16_TO_FLOAT\(x\)](#) ((float)(x) / 32767.0f)
- #define [SENSITIVITY](#) 0.5f
- #define [MIDI_LOWEST_NOTE](#) 21
- #define [MIDI_HIGHEST_NOTE](#) 108
- #define [MIDI_KEY_BOUNDARY\(x\)](#) ((x) < [MIDI_LOWEST_NOTE](#) || (x) > [MIDI_HIGHEST_NOTE](#))
- #define [CONCERT_A](#) 440.0f
- #define [CONCERT_A_NOTE](#) 69

Functions

- `esp_err_t gitcon_init (gitcon_context_t **out_handle)`
- `esp_err_t gitcon_exit (gitcon_handle_t handle)`
frees all resources

4.25.1 Detailed Description

Gitcon Driver Source.

Author

@s-grundner @Laurenz03

Version

0.1

Date

2022-12-23

Copyright

Copyright (c) 2022

Definition in file [gitcon.c](#).

4.25.2 Macro Definition Documentation

4.25.2.1 CONCERT_A

```
#define CONCERT_A 440.0f
```

Definition at line 26 of file [gitcon.c](#).

4.25.2.2 CONCERT_A_NOTE

```
#define CONCERT_A_NOTE 69
```

Definition at line 27 of file [gitcon.c](#).

4.25.2.3 FLOAT_TO_UINT16

```
#define FLOAT_TO_UINT16(  
    x ) ((uint16_t)((x)*32767.0f))
```

Definition at line 20 of file [gitcon.c](#).

4.25.2.4 MIDI_HIGHEST_NOTE

```
#define MIDI_HIGHEST_NOTE 108
```

Definition at line 24 of file [gitcon.c](#).

4.25.2.5 MIDI_KEY_BOUNDARY

```
#define MIDI_KEY_BOUNDARY(  
    x ) ((x) < MIDI_LOWEST_NOTE || (x) > MIDI_HIGHEST_NOTE)
```

Definition at line 25 of file [gitcon.c](#).

4.25.2.6 MIDI_LOWEST_NOTE

```
#define MIDI_LOWEST_NOTE 21
```

Definition at line 23 of file [gitcon.c](#).

4.25.2.7 SENSITIVITY

```
#define SENSITIVITY 0.5f
```

Definition at line 22 of file [gitcon.c](#).

4.25.2.8 UINT16_TO_FLOAT

```
#define UINT16_TO_FLOAT(  
    x ) ((float)(x) / 32767.0f)
```

Definition at line 21 of file [gitcon.c](#).

4.25.3 Function Documentation

4.25.3.1 gitcon_exit()

```
esp_err_t gitcon_exit (  
    gitcon_handle_t handle )
```

frees all resources

Parameters

<i>handle</i>	gitcon context handler
---------------	------------------------

Returns

ESP_OK on success

Definition at line 271 of file [gitcon.c](#).

4.25.3.2 gitcon_init()

```
esp_err_t gitcon_init (
    gitcon_context_t ** out_handle )
```

Note

DSP task: receives audio data from DMA task and sends audio data to MIDI task

MIDI task: receives midi messages from DSP task and sends them to MIDI UART

Definition at line 196 of file [gitcon.c](#).

4.26 gitcon.c

[Go to the documentation of this file.](#)

```
00001
00012 #include "gitcon.h"
00013 #include "processed-data.h"
00014
00015 static const char *TAG = "gitcon";
00016
00017 static TaskHandle_t midi_task_handle;
00018 static TaskHandle_t dsp_task_handle;
00019
00020 #define FLOAT_TO_UINT16(x) ((uint16_t)((x)*32767.0f))
00021 #define UINT16_TO_FLOAT(x) ((float)(x) / 32767.0f)
00022 #define SENSITIVITY 0.5f
00023 #define MIDI_LOWEST_NOTE 21
00024 #define MIDI_HIGHEST_NOTE 108
00025 #define MIDI_KEY_BOUNDARY(x) ((x) < MIDI_LOWEST_NOTE || (x) > MIDI_HIGHEST_NOTE)
00026 #define CONCERT_A 440.0f
00027 #define CONCERT_A_NOTE 69
00028
00029 // uncomment to enable debug output for better_serial_plotter software
00030 // #define DEBUG_BETTER_SERIAL_PLOTTER
00031
00032 // uncomment to enable debug output for dsp_task
00033 // #define DEBUG_DSP
00034
00035 // -----
00036 // static functions
00037 // -----
00038
00043 static void dsp_task(void *arg)
00044 {
00045     // gitcon driver context handler
00046     gitcon_handle_t gitcon_handle = (gitcon_handle_t)arg;
00047
00048     // audio buffer variables
00049     uint16_t *audio_buffer = NULL;
00050     float *audio_buffer_float = (float *)malloc(FFT_SIZE * sizeof(float));
```

```

00051     if (audio_buffer_float == NULL)
00052     {
00053         ESP_LOGE(TAG, "Could not allocate memory for audio_buffer_float");
00054         gitcon_exit(gitcon_handle);
00055         return;
00056     }
00057
00058     // fft variables
00059     float fft_buffer[FFT_SIZE];
00060     float magnitude[FFT_SIZE / 2];
00061     float frequency[FFT_SIZE / 2];
00062     unsigned char keyNR[FFT_SIZE / 2];
00063     float ratio = (float)F_SAMPLE_HZ / (float)FFT_SIZE;
00064
00065     // active notes resulting from fft
00066     midi_message_t *active_notes = (midi_message_t *)malloc(128 * sizeof(midi_message_t));
00067     if (active_notes == NULL)
00068     {
00069         ESP_LOGE(TAG, "Could not allocate memory for active_notes");
00070         gitcon_exit(gitcon_handle);
00071         return;
00072     }
00073
00074     // initialize active_notes
00075     for (int i = 0; i < 128; i++)
00076     {
00077         active_notes[i].channel = 0;
00078         active_notes[i].status = MIDI_STATUS_NOTE_OFF;
00079         active_notes[i].param1 = i;
00080         active_notes[i].param2 = 0;
00081     }
00082
00083     // window counter to sweep through the audio_buffer_float
00084     char window_counter = 0;
00085
00086     for (;;)
00087     {
00088         // delay to avoid overflow and to allow other tasks to run
00089         vTaskDelay(10 / portTICK_PERIOD_MS);
00090
00091         // get audio buffer from sampler
00092         if (xQueueReceive(gitcon_handle->sampler->dsp_queue, &audio_buffer, portMAX_DELAY) == pdFALSE)
00093             continue; // skip iteration if queue is empty
00094
00095         float *start_pos = audio_buffer_float;
00096         audio_buffer_float += (AUDIO_BUFFER_SIZE * window_counter); // move pointer to the next window
00097         window_counter = (window_counter + 1) % (FFT_WINDOW_SIZE); // increment window counter
00098
00099         // starting in a new window and fill the buffer with the new data
00100         for (int i = 0; i < AUDIO_BUFFER_SIZE; i++)
00101             audio_buffer_float[i] = UINT16_TO_FLOAT(audio_buffer[i]);
00102         audio_buffer_float = start_pos;
00103
00104 #ifdef DEBUG_BETTER_SERIAL_PLOTTER
00105     for (int i = 0; i < FFT_SIZE; i++)
00106         printf("%f\n", audio_buffer_float[i]); // for debugging in BetterSerialPlotter
00107 #endif
00108
00109     fft_config_t *fft_plan = fft_init(FFT_SIZE, FFT_REAL, FFT_FORWARD, audio_buffer_float,
00110 fft_buffer);
00111     if (fft_plan == NULL)
00112     {
00113         ESP_LOGE(TAG, "FFT plan could not be created");
00114         vTaskDelay(1000 / portTICK_PERIOD_MS);
00115         continue;
00116     }
00117     fft_execute(fft_plan);
00118
00119     bool invalid_key = false; // invalid key flag
00120     for (int k = 1; k < FFT_SIZE / 2; k++)
00121     {
00122         // detect fundamental frequencies
00123         frequency[k] = k * ratio;
00124         // convert to note number on piano roll
00125         keyNR[k] = (unsigned char)round(log2(frequency[k] / CONCERT_A) * 12 + CONCERT_A_NOTE) %
00126 128;
00127
00128         // calculate magnitude (absolute value of complex number)
00129         magnitude[k] = 2 * sqrt(pow(fft_buffer[2 * k], 2) + pow(fft_buffer[2 * k + 1], 2)) /
00130 FFT_SIZE;
00131     }
00132
00133     // calculate max magnitude for thresholding
00134     float max = 0;
00135     for (int i = 0; i < FFT_SIZE / 2; i++)
00136         max = (magnitude[i] > max) ? magnitude[i] : max;
00137
00138     if (max < 0.0005)

```

```

00142         max = 100;
00143
00144         // check if magnitudes pass a certain threshold
00145         for (int k = 1; k < FFT_SIZE / 2; k++)
00146         {
00147             // switch off notes that are not active
00148             if ((magnitude[k] < max * SENSITIVITY) || MIDI_KEY_BOUNDARY(keyNR[k]))
00149             {
00150                 active_notes[keyNR[k]].status = MIDI_STATUS_NOTE_OFF;
00151                 continue; // skip to next iteration threshold is not passed
00152             }
00153 #ifdef DEBUG_DSP
00154             ESP_LOGI(TAG, "keyNR: %d, magnitude: %f, frequency: %f", keyNR[k], magnitude[k],
00155                     frequency[k]);
00156 #endif
00157             active_notes[keyNR[k]].status = MIDI_STATUS_NOTE_ON;
00158             active_notes[keyNR[k]].param2 = (uint8_t)(magnitude[k] / max * 127);
00159         }
00160         // send saved notes to MIDI queue
00161         xQueueSend(gitcon_handle->midi_queue, &active_notes, portMAX_DELAY);
00162         fft_destroy(fft_plan);
00163     } // for(;;)
00164 } // dsp_task
00165
00166 static void midi_task(void *arg)
00167 {
00168     // gitcon driver context handler
00169     gitcon_handle_t gitcon_handle = (gitcon_handle_t)arg;
00170     midi_message_t *active_notes = NULL;
00171     midi_status_t previous_states[128] = {0};
00172     for (;;)
00173     {
00174         vTaskDelay(10 / portTICK_PERIOD_MS);
00175
00176         if (xQueueReceive(gitcon_handle->midi_queue, &active_notes, portMAX_DELAY) == pdFALSE)
00177             continue; // skip iteration if queue is empty
00178
00179         // send MIDI messages to UART
00180         for (size_t i = 0; i < 128; i++)
00181         {
00182             if (active_notes[i].status == previous_states[i])
00183                 continue; // continue if note has not changed
00184             // send message to MIDI UART
00185             ESP_ERROR_CHECK(midi_write(gitcon_handle->midi_handle, &active_notes[i]));
00186             previous_states[i] = active_notes[i].status;
00187         }
00188     } // for(;;)
00189 } // midi_task
00190
00191 // -----
00192 // non-static functions
00193 // -----
00194
00195 esp_err_t gitcon_init(gitcon_context_t **out_handle)
00196 {
00197     gitcon_context_t *gitcon_cfg = (gitcon_context_t *)malloc(sizeof(gitcon_context_t));
00198     if (!gitcon_cfg)
00199         return ESP_ERR_NO_MEM;
00200
00201     // create queue for audio data (passed into sampler)
00202     QueueHandle_t dsp_queue = xQueueCreate(10, sizeof(size_t *));
00203
00204     // create queue for midi messages
00205     gitcon_cfg->midi_queue = xQueueCreate(5, sizeof(midi_handle_t *));
00206     if (!gitcon_cfg->midi_queue)
00207         return ESP_ERR_NO_MEM;
00208
00209 #ifdef USE_MCP3201
00210     // -----
00211     // SPI
00212     // -----
00213     spi_bus_config_t bus_cfg = {
00214         .miso_io_num = SPI_MISO,
00215         .mosi_io_num = SPI_MOSI,
00216         .sclk_io_num = SPI_SCLK,
00217         .max_transfer_sz = 32,
00218     };
00219     spi_bus_initialize(SPI_DEV, &bus_cfg, DMA_CHAN);
00220
00221     // -----
00222     // MCP3201 (ADC)
00223     // -----
00224     mcp3201_handle_t mcp_handle;
00225     mcp3201_config_t mcp_cfg = {
00226         .host = SPI_DEV,
00227         .cs_io = SPI_CS,
00228         .miso_io = SPI_MISO,

```

```

00230     .mosi_io = SPI_MOSI};
00231     // initialize ADC and store in gitcon handle
00232     ESP_ERROR_CHECK(mcp3201_init(&mcp_handle, &mcp_cfg));
00233     gitcon_cfg->sampler = mcp3201_sampler_start(mcp_handle, dsp_queue, AUDIO_BUFFER_SIZE,
F_SAMPLE_HZ);
00234 #else
00235     gitcon_cfg->sampler = i2s_sampler_start(INTERNAL_ADC_CHANNEL, dsp_queue, AUDIO_BUFFER_SIZE,
F_SAMPLE_HZ);
00236 #endif
00237
00238     // -----
00239     // MIDI
00240     // -----
00241
00242     // Configure a handle parameter for MIDI
00243     midi_handle_t midi_handle;
00244     midi_config_t midi_cfg = {
00245         .uart_num = MIDI_UART,
00246         .baudrate = MIDI_BAUD,
00247         .rx_io = MIDI_RX,
00248         .tx_io = MIDI_TX};
00249     // Initialize MIDI and store in gitcon handle
00250     ESP_ERROR_CHECK(midi_init(&midi_handle, &midi_cfg));
00251     gitcon_cfg->midi_handle = midi_handle;
00252
00253     // -----
00254     // INIT RTOS
00255     // -----
00256
00257     ESP_LOGI(TAG, "Creating RTOS tasks...");
00259     if (xTaskCreatePinnedToCore(dsp_task, "dsp_task", 1 « 16, gitcon_cfg, 5, &dsp_task_handle, 1) ==
pdFALSE)
00260         return ESP_ERR_NO_MEM;
00261
00263     if (xTaskCreatePinnedToCore(midi_task, "midi_task", 2048, gitcon_cfg, 5, &midi_task_handle, 0) ==
pdFALSE)
00264         return ESP_ERR_NO_MEM;
00265
00266     // Pass final configuration to outer parameters
00267     *out_handle = gitcon_cfg;
00268     return ESP_OK;
00269 }
00270
00271 esp_err_t gitcon_exit(gitcon_handle_t handle)
00272 {
00273     ESP_ERROR_CHECK(midi_exit(handle->midi_handle));
00274
00275     // stop tasks
00276     vTaskDelete(dsp_task_handle);
00277     vTaskDelete(midi_task_handle);
00278
00279     // stop sampler
00280 #ifdef USE_MCP3201
00281     mcp3201_sampler_stop(handle->sampler->mcp_handle);
00282     ESP_ERROR_CHECK(mcp3201_exit(handle->sampler->mcp_handle));
00283 #else
00284     i2s_sampler_stop(handle->sampler);
00285 #endif
00286
00287     free(handle);
00288     return ESP_OK;
00289 }

```

4.27 src/gitcon.h File Reference

Gitcon Driver Header.

```
#include "config.h"
```

Data Structures

- struct [gitcon_context_t](#)
Gitcon Configuration.

Macros

- `#define GITCON_LOG_LEVEL ESP_LOG_ERROR`

Typedefs

- `typedef gitcon_context_t * gitcon_handle_t`

Functions

- `esp_err_t gitcon_init (gitcon_handle_t *out_handle)`
initializes gitcon device and installs peripheral drivers
- `esp_err_t gitcon_exit (gitcon_handle_t handle)`
frees all resources

4.27.1 Detailed Description

Gitcon Driver Header.

Author

@s-grundner @Laurenz03

Version

0.1

Date

2022-12-23

Copyright

Copyright (c) 2022

Definition in file [gitcon.h](#).

4.27.2 Macro Definition Documentation

4.27.2.1 GITCON_LOG_LEVEL

```
#define GITCON_LOG_LEVEL ESP_LOG_ERROR
```

Definition at line 17 of file [gitcon.h](#).

4.27.3 Typedef Documentation

4.27.3.1 gitcon_handle_t

```
typedef gitcon_context_t* gitcon_handle_t
```

Definition at line 37 of file [gitcon.h](#).

4.27.4 Function Documentation

4.27.4.1 gitcon_exit()

```
esp_err_t gitcon_exit (  
    gitcon_handle_t handle )
```

frees all resources

Parameters

<i>handle</i>	gitcon context handler
---------------	------------------------

Returns

ESP_OK on success

Definition at line 271 of file [gitcon.c](#).

4.27.4.2 gitcon_init()

```
esp_err_t gitcon_init (  
    gitcon_handle_t * out_handle )
```

initializes gitcon device and installs peripheral drivers

Parameters

out	<i>out_handle</i>	gitcon context handler
-----	-------------------	------------------------

Returns

esp_err_t ESP_OK on success, ESP_ERR_NO_MEM on memory allocation error

4.28 gitcon.h

[Go to the documentation of this file.](#)

```
00001
00012 #ifndef GITCON_H
00013 #define GITCON_H
00014
00015 #include "config.h"
00016
00017 #define GITCON_LOG_LEVEL ESP_LOG_ERROR
00018
00025 typedef struct
00026 {
00027     #ifdef USE_MCP3201
00028         mcp3201_sampler_t *sampler;
00029     #else
00030         i2s_sampler_t *sampler;
00031     #endif
00032     midi_handle_t midi_handle;
00033     QueueHandle_t midi_queue;
00034 } gitcon_context_t;
00035
00037 typedef gitcon_context_t *gitcon_handle_t;
00038
00045 esp_err_t gitcon_init(gitcon_handle_t *out_handle);
00046
00053 esp_err_t gitcon_exit(gitcon_handle_t handle);
00054
00055 #endif // GITCON_H
```

4.29 src/main.c File Reference

Main File for Gitcon Project.

```
#include "gitcon.h"
```

Macros

- #define `USER_LOCAL_LEVEL` ESP_LOG_ERROR
- #define `PROTOTYPE` 0

Functions

- void `app_main` (void)

4.29.1 Detailed Description

Main File for Gitcon Project.

Author

@s-grundner

Version

0.1

Date

2022-12-23

Copyright

Copyright (c) 2022

Definition in file [main.c](#).

4.29.2 Macro Definition Documentation

4.29.2.1 PROTOTYPE

```
#define PROTOTYPE 0
```

Definition at line 17 of file [main.c](#).

4.29.2.2 USER_LOCAL_LEVEL

```
#define USER_LOCAL_LEVEL ESP_LOG_ERROR
```

Definition at line 14 of file [main.c](#).

4.29.3 Function Documentation

4.29.3.1 app_main()

```
void app_main (
    void )
```

Definition at line 80 of file [main.c](#).

4.30 main.c

[Go to the documentation of this file.](#)

```
00001
00012 #include "gitcon.h"
00013
00014 #define USER_LOCAL_LEVEL ESP_LOG_ERROR
00015
00016 // when PROTOTYPE is 1, the prototype board is used
00017 #define PROTOTYPE 0
00018
00019 static const char *TAG = "main";
00020
00021 #if PROTOTYPE == 1
00022 #define DIP_POL 8
00023 #define DEBOUNCE_TIME_MS 50
00024
00025 static const char WHOLE_TONE[8] = {0, 2, 4, 5, 7, 9, 11, 12};
00026
00027 static const gpio_num_t DIP_IO[DIP_POL] = {
00028     GPIO_NUM_32,
00029     GPIO_NUM_33,
00030     GPIO_NUM_25,
00031     GPIO_NUM_26,
00032     GPIO_NUM_27,
00033     GPIO_NUM_14,
00034     GPIO_NUM_12,
00035     GPIO_NUM_13,
00036 };
00037
00038 typedef struct
00039 {
00040     QueueHandle_t midi_queue;
00041     size_t *current_dip;
00042     size_t *previous_dip;
00043     TimerHandle_t *timer;
00044 } dip_switch_t;
00045
00046 static void IRAM_ATTR dip_isr(void *args)
00047 {
00048     dip_switch_t *dip_switch = (dip_switch_t *)args;
00049     for (size_t i = 0; i < DIP_POL; i++)
00050     {
00051         // read DIP switch
00052         dip_switch->current_dip[i] = !gpio_get_level(DIP_IO[i]);
00053
00054         // continue if the dip state hasn't changed
00055         if (dip_switch->current_dip[i] == dip_switch->previous_dip[i])
00056             continue;
00057
00058         // send MIDI message
00059         midi_message_t msg = {
00060             .status = (midi_status_t) (dip_switch->current_dip[i]) ? MIDI_STATUS_NOTE_ON :
MIDI_STATUS_NOTE_OFF,
00061             .channel = 0,
00062             .param1 = 0x3C + WHOLE_TONE[i], // C4 + WHOLE_TONE[i]
00063             .param2 = 127};
00064         xQueueSendFromISR(dip_switch->midi_queue, &msg, NULL);
00065
00066         // disable interrupts for gpios to debounce
00067         ESP_ERROR_CHECK(gpio_intr_disable(DIP_IO[i]));
00068         dip_switch->previous_dip[i] = dip_switch->current_dip[i];
00069         xTimerStartFromISR(dip_switch->timer[i], NULL);
00070     }
00071 }
00072
00073 static void IRAM_ATTR debounce_task(TimerHandle_t debounce_timer)
00074 {
00075     int i = (int)pvTimerGetTimerID(debounce_timer);
00076     gpio_intr_enable(DIP_IO[i]);
00077 }
```

```

00078 #endif
00079
00080 void app_main(void)
00081 {
00082     gitcon_handle_t handle;
00083     if (gitcon_init(&handle) != ESP_OK)
00084     {
00085         ESP_LOGE(TAG, "gitcon_init failed");
00086         ESP_ERROR_CHECK(gitcon_exit(handle));
00087         return;
00088     }
00089     #if PROTOTYPE == 1
00090     xTimerHandle debounce_timers[DIP_POL];
00091
00092     for (size_t i = 0; i < DIP_POL; i++)
00093         debounce_timers[i] = xTimerCreate("dip_switch", pdMS_TO_TICKS(DEBOUNCE_TIME_MS), pdFALSE,
00094         (void *)i, debounce_task);
00095
00096     size_t current_dip[DIP_POL] = {0};
00097     size_t previous_dip[DIP_POL] = {0};
00098
00099     // setup debouncing for DIP Switches
00100     dip_switch_t dip_switch = {
00101         .current_dip = current_dip,
00102         .previous_dip = previous_dip,
00103         .timer = debounce_timers,
00104         .midi_queue = handle->midi_queue;
00105
00106     // setup interrupt for DIP switches
00107     ESP_ERROR_CHECK(gpio_install_isr_service(0));
00108     for (size_t i = 0; i < DIP_POL; i++)
00109     {
00110         gpio_config_t io_conf = {
00111             .pin_bit_mask = (1ULL << DIP_IO[i]),
00112             .mode = GPIO_MODE_INPUT,
00113             .pull_up_en = GPIO_PULLUP_ENABLE,
00114             .pull_down_en = GPIO_PULLDOWN_DISABLE,
00115             .intr_type = GPIO_INTR_ANYEDGE;
00116
00117         ESP_ERROR_CHECK(gpio_config(&io_conf));
00118         ESP_ERROR_CHECK(gpio_isr_handler_add(DIP_IO[i], dip_isr, &dip_switch));
00119     }
00120 #endif
00121     while (1)
00122         vTaskDelay(1000 / portTICK_PERIOD_MS);
00123 }

```

4.31 test/test_fft/test_fft.c File Reference

Unit Test for FFT.

```

#include <processed-data.h>
#include <fft.h>
#include <unity.h>

```

Macros

- #define [AUDIO_BUFFER_SIZE](#) 512
- #define [F_SAMPLE_HZ](#) 44100
- #define [FFT_WINDOW_SIZE](#) 2
- #define [FFT_SIZE](#) 4096

Functions

- void [setUp](#) (void)
- void [tearDown](#) (void)
- void [test_processed_data](#) (void)
- void [app_main](#) ()

4.31.1 Detailed Description

Unit Test for FFT.

Author

@s-grundner

Version

0.1

Date

2023-03-26

Copyright

Copyright (c) 2023

Definition in file [test_fft.c](#).

4.31.2 Macro Definition Documentation

4.31.2.1 AUDIO_BUFFER_SIZE

```
#define AUDIO_BUFFER_SIZE 512
```

Definition at line 15 of file [test_fft.c](#).

4.31.2.2 F_SAMPLE_HZ

```
#define F_SAMPLE_HZ 44100
```

Definition at line 16 of file [test_fft.c](#).

4.31.2.3 FFT_SIZE

```
#define FFT_SIZE 4096
```

Definition at line 18 of file [test_fft.c](#).

4.31.2.4 FFT_WINDOW_SIZE

```
#define FFT_WINDOW_SIZE 2
```

Definition at line 17 of file [test_fft.c](#).

4.31.3 Function Documentation

4.31.3.1 app_main()

```
void app_main (  
    void )
```

Definition at line 62 of file [test_fft.c](#).

4.31.3.2 setUp()

```
void setUp (  
    void )
```

Definition at line 20 of file [test_fft.c](#).

4.31.3.3 tearDown()

```
void tearDown (  
    void )
```

Definition at line 24 of file [test_fft.c](#).

4.31.3.4 test_processed_data()

```
void test_processed_data (  
    void )
```

Definition at line 28 of file [test_fft.c](#).

4.32 test_fft.c

[Go to the documentation of this file.](#)

```

00001
00011 #include <processed-data.h>
00012 #include <fft.h>
00013 #include <unity.h>
00014
00015 #define AUDIO_BUFFER_SIZE 512 // Size of buffer for FFT and sampler
00016 #define F_SAMPLE_HZ 44100    // Sample rate of FFT and sampler
00017 #define FFT_WINDOW_SIZE 2    // Amount of buffers to take for FFT
00018 #define FFT_SIZE 4096        // (AUDIO_BUFFER_SIZE * FFT_WINDOW_SIZE) Amount of samples to take for
    FFT
00019
00020 void setUp(void)
00021 {
00022 }
00023
00024 void tearDown(void)
00025 {
00026 }
00027
00028 void test_processed_data(void)
00029 {
00030     // fft variables
00031     float fft_buffer[FFT_SIZE];
00032     float magnitude[FFT_SIZE / 2];
00033     float frequency[FFT_SIZE / 2];
00034     unsigned char keyNR[FFT_SIZE / 2];
00035     float ratio = (float)F_SAMPLE_HZ / (float)FFT_SIZE;
00036
00037     fft_config_t *real_fft_plan = fft_init(FFT_SIZE, FFT_REAL, FFT_FORWARD, test_buffer, fft_buffer);
00038     TEST_ASSERT_NOT_NULL(real_fft_plan);
00039     fft_execute(real_fft_plan);
00040
00041     for (int k = 1; k < FFT_SIZE / 2; k++)
00042     {
00043         magnitude[k] = 2 * sqrt(pow(fft_buffer[2 * k], 2) + pow(fft_buffer[2 * k + 1], 2)) / FFT_SIZE;
00044         frequency[k] = k * ratio;
00045         keyNR[k] = (unsigned char)round(log2(frequency[k] / 440) * 12 + 69) % 128;
00046     }
00047
00048     float max = 0;
00049     for (int i = 0; i < FFT_SIZE / 2; i++)
00050         max = (magnitude[i] > max) ? magnitude[i] : max;
00051
00052     for (int k = 1; k < FFT_SIZE / 2; k++)
00053     {
00054         if (magnitude[k] >= max * 0.5)
00055         {
00056             TEST_ASSERT_EQUAL(45, keyNR[k]);
00057         }
00058     }
00059     fft_destroy(real_fft_plan);
00060 }
00061
00062 void app_main()
00063 {
00064     UNITY_BEGIN();
00065     RUN_TEST(test_processed_data);
00066     UNITY_END();
00067 }

```

4.33 test/test_midi/test_midi.c File Reference

Unit Tests for MIDI Driver.

```

#include <midi.h>
#include <unity.h>

```

Macros

- #define [MIDI_UART](#) (UART_NUM_1)
- #define [MIDI_BAUD](#) (115200)
- #define [MIDI_TX](#) (GPIO_NUM_26)
- #define [MIDI_RX](#) (GPIO_NUM_27)

Functions

- void [setUp](#) (void)
- void [tearDown](#) (void)
- void [test_midi_blink](#) (void)
- void [test_midi_bend](#) (void)
- void [test_midi_bent_note](#) (void)
- void [app_main](#) ()

4.33.1 Detailed Description

Unit Tests for MIDI Driver.

Author

@s-grundner

Version

0.1

Date

2023-03-26

Copyright

Copyright (c) 2023

Definition in file [test_midi.c](#).

4.33.2 Macro Definition Documentation

4.33.2.1 MIDI_BAUD

```
#define MIDI_BAUD (115200)
```

Definition at line [18](#) of file [test_midi.c](#).

4.33.2.2 MIDI_RX

```
#define MIDI_RX (GPIO_NUM_27)
```

Definition at line [20](#) of file [test_midi.c](#).

4.33.2.3 MIDI_TX

```
#define MIDI_TX (GPIO_NUM_26)
```

Definition at line 19 of file [test_midi.c](#).

4.33.2.4 MIDI_UART

```
#define MIDI_UART (UART_NUM_1)
```

Definition at line 17 of file [test_midi.c](#).

4.33.3 Function Documentation

4.33.3.1 app_main()

```
void app_main (  
    void )
```

Definition at line 87 of file [test_midi.c](#).

4.33.3.2 setUp()

```
void setUp (  
    void )
```

Definition at line 22 of file [test_midi.c](#).

4.33.3.3 tearDown()

```
void tearDown (  
    void )
```

Definition at line 32 of file [test_midi.c](#).

4.33.3.4 test_midi_bend()

```
void test_midi_bend (
    void )
```

Definition at line 46 of file [test_midi.c](#).

4.33.3.5 test_midi_bent_note()

```
void test_midi_bent_note (
    void )
```

Definition at line 58 of file [test_midi.c](#).

4.33.3.6 test_midi_blink()

```
void test_midi_blink (
    void )
```

Definition at line 37 of file [test_midi.c](#).

4.34 test_midi.c

[Go to the documentation of this file.](#)

```
00001
00012 #include <midi.h>
00013 #include <unity.h>
00014
00015 static midi_handle_t midi_handle;
00016
00017 #define MIDI_UART (UART_NUM_1)
00018 #define MIDI_BAUD (115200)
00019 #define MIDI_TX (GPIO_NUM_26)
00020 #define MIDI_RX (GPIO_NUM_27)
00021
00022 void setUp(void)
00023 {
00024     static midi_config_t midi_cfg = {
00025         .uart_num = MIDI_UART,
00026         .baudrate = MIDI_BAUD,
00027         .rx_io = MIDI_RX,
00028         .tx_io = MIDI_TX};
00029     ESP_ERROR_CHECK(midi_init(&midi_handle, &midi_cfg));
00030 }
00031
00032 void tearDown(void)
00033 {
00034     midi_exit(midi_handle);
00035 }
00036
00037 void test_midi_blink(void)
00038 {
00039     midi_message_t msg = note_on(0, 0x3C, 0x7F);
00040     TEST_ASSERT_EQUAL(ESP_OK, midi_write(midi_handle, &msg));
00041     vTaskDelay(1000 / portTICK_PERIOD_MS);
00042     msg.status = MIDI_STATUS_NOTE_OFF;
00043     TEST_ASSERT_EQUAL(ESP_OK, midi_write(midi_handle, &msg));
00044 }
00045
00046 void test_midi_bend(void)
```



```
00047 {
00048     midi_message_t msg = pitch_bend(0, MIDI_PITCH_BEND_MAX);
00049     TEST_ASSERT_EQUAL(ESP_OK, midi_write(midi_handle, &msg));
00050     vTaskDelay(1000 / portTICK_PERIOD_MS);
00051     msg = pitch_bend(0, MIDI_PITCH_BEND_MIN);
00052     TEST_ASSERT_EQUAL(ESP_OK, midi_write(midi_handle, &msg));
00053     vTaskDelay(1000 / portTICK_PERIOD_MS);
00054     msg = pitch_bend(0, MIDI_PITCH_BEND_CENTER);
00055     TEST_ASSERT_EQUAL(ESP_OK, midi_write(midi_handle, &msg));
00056 }
00057
00058 void test_midi_bent_note(void)
00059 {
00060     midi_message_t msg = note_on(0, 0x3C, 0x7F);
00061     TEST_ASSERT_EQUAL(ESP_OK, midi_write(midi_handle, &msg));
00062
00063     // ramp up and down pitch bend in 1 second
00064     midi_message_t msg_bend = pitch_bend(0, MIDI_PITCH_BEND_CENTER);
00065     for (int i = MIDI_PITCH_BEND_CENTER; i <= MIDI_PITCH_BEND_MAX; i += 128)
00066     {
00067         msg_bend = pitch_bend(0, i);
00068         TEST_ASSERT_EQUAL(ESP_OK, midi_write(midi_handle, &msg_bend));
00069         vTaskDelay(4 / portTICK_PERIOD_MS);
00070     }
00071     for (int i = MIDI_PITCH_BEND_MAX; i >= MIDI_PITCH_BEND_MIN; i -= 128)
00072     {
00073         msg_bend = pitch_bend(0, i);
00074         TEST_ASSERT_EQUAL(ESP_OK, midi_write(midi_handle, &msg_bend));
00075         vTaskDelay(2 / portTICK_PERIOD_MS);
00076     }
00077     for (int i = MIDI_PITCH_BEND_MIN; i <= MIDI_PITCH_BEND_CENTER; i += 128)
00078     {
00079         msg_bend = pitch_bend(0, i);
00080         TEST_ASSERT_EQUAL(ESP_OK, midi_write(midi_handle, &msg_bend));
00081         vTaskDelay(4 / portTICK_PERIOD_MS);
00082     }
00083     msg.status = MIDI_STATUS_NOTE_OFF;
00084     TEST_ASSERT_EQUAL(ESP_OK, midi_write(midi_handle, &msg));
00085 }
00086
00087 void app_main()
00088 {
00089     UNITY_BEGIN();
00090
00091     RUN_TEST(test_midi_blink);
00092     RUN_TEST(test_midi_bend);
00093     RUN_TEST(test_midi_bent_note);
00094
00095     UNITY_END();
00096 }
```


Index

ADC_CLK
 mcp3201.h, [85](#)

ADC_RES
 config.h, [20](#)

ADC_RES_BITS
 config.h, [20](#)

app_main
 main.c, [118](#)
 test_fft.c, [122](#)
 test_midi.c, [125](#)

AUDIO_BUFFER_SIZE
 config.h, [21](#)
 test_fft.c, [121](#)

baudrate
 midi_config_t, [14](#)

buffer
 i2s_sampler_t, [8](#)
 mcp3201_sampler_t, [12](#)

buffer_pos
 i2s_sampler_t, [9](#)
 mcp3201_sampler_t, [12](#)

buffer_size
 i2s_sampler_t, [9](#)
 mcp3201_sampler_t, [12](#)

cfg
 mcp3201_context_t, [11](#)
 midi_context_t, [16](#)

channel
 midi_message_t, [17](#)

channel_pressure
 midi.h, [101](#)
 midi_utils.c, [106](#)

CONCERT_A
 gitcon.c, [109](#)

CONCERT_A_NOTE
 gitcon.c, [109](#)

config.h
 ADC_RES, [20](#)
 ADC_RES_BITS, [20](#)
 AUDIO_BUFFER_SIZE, [21](#)
 DMA_CHAN, [21](#)
 F_SAMPLE_HZ, [21](#)
 FFT_SIZE, [21](#)
 FFT_WINDOW_SIZE, [21](#)
 INTERNAL_ADC_CHANNEL, [21](#)
 INTERNAL_ADC_IO, [22](#)
 INTERNAL_ADC_UNIT, [22](#)
 MIDI_BAUD, [22](#)
 MIDI_RX, [22](#)
 MIDI_TX, [22](#)
 MIDI_UART, [22](#)
 SPI_CS, [23](#)
 SPI_DEV, [23](#)
 SPI_MISO, [23](#)
 SPI_MOSI, [23](#)
 SPI_SCLK, [23](#)

cs_io
 mcp3201_config_t, [10](#)

ctrl_change
 midi.h, [101](#)
 midi_utils.c, [106](#)

direction
 fft_config_t, [5](#)

DMA_CHAN
 config.h, [21](#)

dma_chan
 mcp3201_config_t, [10](#)

dma_queue
 i2s_sampler_t, [9](#)
 mcp3201_sampler_t, [13](#)

dsp_queue
 i2s_sampler_t, [9](#)
 mcp3201_sampler_t, [13](#)

F_SAMPLE_HZ
 config.h, [21](#)
 test_fft.c, [121](#)

fft
 fft.c, [65](#)
 fft.h, [77](#)

fft.c
 fft, [65](#)
 fft4, [65](#)
 fft8, [65](#)
 fft_destroy, [65](#)
 fft_execute, [66](#)
 fft_init, [66](#)
 fft_primitive, [66](#)
 ifft, [66](#)
 ifft_primitive, [66](#)
 irfft, [67](#)
 LARGE_BASE_CASE, [64](#)
 rfft, [67](#)
 split_radix_fft, [67](#)
 TWO_PI, [64](#)
 USE_SPLIT_RADIX, [65](#)

fft.h

- fft, 77
- fft4, 78
- fft8, 78
- FFT_BACKWARD, 76
- FFT_COMPLEX, 77
- fft_destroy, 78
- fft_direction_t, 76
- fft_execute, 78
- FFT_FORWARD, 76
- fft_init, 78
- FFT_OWN_INPUT_MEM, 76
- FFT_OWN_OUTPUT_MEM, 76
- fft_primitive, 79
- FFT_REAL, 77
- fft_type_t, 77
- ifft, 79
- ifft_primitive, 79
- irfft, 79
- rfft, 80
- split_radix_fft, 80
- fft4
 - fft.c, 65
 - fft.h, 78
- fft8
 - fft.c, 65
 - fft.h, 78
- FFT_BACKWARD
 - fft.h, 76
- FFT_COMPLEX
 - fft.h, 77
- fft_config_t, 5
 - direction, 5
 - flags, 5
 - input, 6
 - output, 6
 - size, 6
 - twiddle_factors, 6
 - type, 6
- fft_destroy
 - fft.c, 65
 - fft.h, 78
- fft_direction_t
 - fft.h, 76
- fft_execute
 - fft.c, 66
 - fft.h, 78
- FFT_FORWARD
 - fft.h, 76
- fft_init
 - fft.c, 66
 - fft.h, 78
- FFT_OWN_INPUT_MEM
 - fft.h, 76
- FFT_OWN_OUTPUT_MEM
 - fft.h, 76
- fft_primitive
 - fft.c, 66
 - fft.h, 79
- FFT_REAL
 - fft.h, 77
- FFT_SIZE
 - config.h, 21
 - test_fft.c, 121
- fft_type_t
 - fft.h, 77
- FFT_WINDOW_SIZE
 - config.h, 21
 - test_fft.c, 121
- flags
 - fft_config_t, 5
- FLOAT_TO_UINT16
 - gitcon.c, 109
- gitcon.c
 - CONCERT_A, 109
 - CONCERT_A_NOTE, 109
 - FLOAT_TO_UINT16, 109
 - gitcon_exit, 110
 - gitcon_init, 111
 - MIDI_HIGHEST_NOTE, 110
 - MIDI_KEY_BOUNDARY, 110
 - MIDI_LOWEST_NOTE, 110
 - SENSITIVITY, 110
 - UINT16_TO_FLOAT, 110
- gitcon.h
 - gitcon_exit, 116
 - gitcon_handle_t, 116
 - gitcon_init, 116
 - GITCON_LOG_LEVEL, 115
- gitcon_context_t, 7
 - midi_handle, 7
 - midi_queue, 7
 - sampler, 7
- gitcon_exit
 - gitcon.c, 110
 - gitcon.h, 116
- gitcon_handle_t
 - gitcon.h, 116
- gitcon_init
 - gitcon.c, 111
 - gitcon.h, 116
- GITCON_LOG_LEVEL
 - gitcon.h, 115
- host
 - mcp3201_config_t, 10
- i2s_sampler.c
 - i2s_sampler_start, 59
 - i2s_sampler_stop, 60
 - READER_TIMEOUT_MS, 59
 - READER_TIMEOUT_TICKS, 59
 - RESAMPLE_DENOMINATOR, 59
- i2s_sampler.h
 - i2s_sampler_start, 62
 - i2s_sampler_stop, 63
- i2s_sampler_start

- i2s_sampler.c, [59](#)
- i2s_sampler.h, [62](#)
- i2s_sampler_stop
 - i2s_sampler.c, [60](#)
 - i2s_sampler.h, [63](#)
- i2s_sampler_t, [8](#)
 - buffer, [8](#)
 - buffer_pos, [9](#)
 - buffer_size, [9](#)
 - dma_queue, [9](#)
 - dsp_queue, [9](#)
- ifft
 - fft.c, [66](#)
 - fft.h, [79](#)
- ifft_primitive
 - fft.c, [66](#)
 - fft.h, [79](#)
- include/config.h, [19](#), [24](#)
- include/processed-data.h, [24](#), [58](#)
- input
 - fft_config_t, [6](#)
- INTERNAL_ADC_CHANNEL
 - config.h, [21](#)
- INTERNAL_ADC_IO
 - config.h, [22](#)
- INTERNAL_ADC_UNIT
 - config.h, [22](#)
- irfft
 - fft.c, [67](#)
 - fft.h, [79](#)
- LARGE_BASE_CASE
 - fft.c, [64](#)
- lib/audio/i2s_sampler.c, [58](#), [60](#)
- lib/audio/i2s_sampler.h, [61](#), [63](#)
- lib/fft/fft.c, [64](#), [68](#)
- lib/fft/fft.h, [75](#), [80](#)
- lib/mcp3201/mcp3201.c, [81](#), [83](#)
- lib/mcp3201/mcp3201.h, [84](#), [88](#)
- lib/mcp3201/mcp3201_sampler.c, [88](#), [91](#)
- lib/midi/midi.c, [93](#), [96](#)
- lib/midi/midi.h, [98](#), [104](#)
- lib/midi/midi_utils.c, [105](#), [107](#)
- main.c
 - app_main, [118](#)
 - PROTOTYPE, [118](#)
 - USER_LOCAL_LEVEL, [118](#)
- mcp3201.c
 - mcp3201_context_t, [82](#)
 - mcp3201_exit, [82](#)
 - mcp3201_init, [82](#)
 - mcp3201_read, [83](#)
- mcp3201.h
 - ADC_CLK, [85](#)
 - mcp3201_exit, [86](#)
 - mcp3201_handle_t, [86](#)
 - mcp3201_init, [86](#)
 - mcp3201_read, [87](#)
 - mcp3201_sampler_start, [87](#)
 - mcp3201_sampler_stop, [87](#)
- mcp3201_config_t, [9](#)
 - cs_io, [10](#)
 - dma_chan, [10](#)
 - host, [10](#)
 - miso_io, [10](#)
 - mosi_io, [10](#)
- mcp3201_context_t, [11](#)
 - cfg, [11](#)
 - mcp3201.c, [82](#)
 - mcp3201_sampler.c, [90](#)
 - ongoing_transaction, [11](#)
 - spi, [11](#)
- mcp3201_exit
 - mcp3201.c, [82](#)
 - mcp3201.h, [86](#)
- mcp3201_handle_t
 - mcp3201.h, [86](#)
- mcp3201_init
 - mcp3201.c, [82](#)
 - mcp3201.h, [86](#)
- mcp3201_read
 - mcp3201.c, [83](#)
 - mcp3201.h, [87](#)
- mcp3201_sampler.c
 - mcp3201_context_t, [90](#)
 - mcp3201_sampler_start, [90](#)
 - mcp3201_sampler_stop, [91](#)
 - READER_TIMEOUT_MS, [89](#)
 - READER_TIMEOUT_TICKS, [90](#)
 - RESAMPLE_DENOMINATOR, [90](#)
- mcp3201_sampler_start
 - mcp3201.h, [87](#)
 - mcp3201_sampler.c, [90](#)
- mcp3201_sampler_stop
 - mcp3201.h, [87](#)
 - mcp3201_sampler.c, [91](#)
- mcp3201_sampler_t, [12](#)
 - buffer, [12](#)
 - buffer_pos, [12](#)
 - buffer_size, [12](#)
 - dma_queue, [13](#)
 - dsp_queue, [13](#)
 - mcp_handle, [13](#)
- mcp_handle
 - mcp3201_sampler_t, [13](#)
- midi.c
 - MIDI_BYTE_SIZE_DEFAULT, [94](#)
 - MIDI_BYTE_SIZE_SHORT, [94](#)
 - midi_context_t, [94](#)
 - midi_exit, [94](#)
 - midi_init, [95](#)
 - midi_read, [95](#)
 - midi_write, [95](#)
- midi.h
 - channel_pressure, [101](#)
 - ctrl_change, [101](#)

- MIDI_BYTE_SIZE_DEFAULT, 99
- MIDI_BYTE_SIZE_SHORT, 99
- midi_exit, 101
- midi_handle_t, 100
- midi_init, 102
- MIDI_LOG_LEVEL, 99
- MIDI_PITCH_BEND_CENTER, 99
- MIDI_PITCH_BEND_MAX, 100
- MIDI_PITCH_BEND_MIN, 100
- midi_read, 102
- MIDI_STATUS_CHANNEL_PRESSURE, 101
- MIDI_STATUS_CONTROL_CHANGE, 101
- MIDI_STATUS_NOTE_OFF, 101
- MIDI_STATUS_NOTE_ON, 101
- MIDI_STATUS_PITCH_BEND, 101
- MIDI_STATUS_POLYPHONIC_KEY_PRESSURE, 101
- MIDI_STATUS_PROGRAM_CHANGE, 101
- midi_status_t, 100
- midi_write, 102
- note_off, 103
- note_on, 103
- pitch_bend, 103
- poly_key_pressure, 103
- prg_change, 104
- MIDI_BAUD
 - config.h, 22
 - test_midi.c, 124
- MIDI_BYTE_SIZE_DEFAULT
 - midi.c, 94
 - midi.h, 99
- MIDI_BYTE_SIZE_SHORT
 - midi.c, 94
 - midi.h, 99
- midi_config_t, 13
 - baudrate, 14
 - rx_io, 14
 - tx_io, 14
 - uart_num, 14
- midi_context_t, 15
 - cfg, 16
 - midi.c, 94
- midi_exit
 - midi.c, 94
 - midi.h, 101
- midi_handle
 - gitcon_context_t, 7
- midi_handle_t
 - midi.h, 100
- MIDI_HIGHEST_NOTE
 - gitcon.c, 110
- midi_init
 - midi.c, 95
 - midi.h, 102
- MIDI_KEY_BOUNDARY
 - gitcon.c, 110
- MIDI_LOG_LEVEL
 - midi.h, 99
- MIDI_LOWEST_NOTE
 - gitcon.c, 110
- midi_message_t, 16
 - channel, 17
 - param1, 17
 - param2, 17
 - status, 17
- MIDI_PITCH_BEND_CENTER
 - midi.h, 99
- MIDI_PITCH_BEND_MAX
 - midi.h, 100
- MIDI_PITCH_BEND_MIN
 - midi.h, 100
- midi_queue
 - gitcon_context_t, 7
- midi_read
 - midi.c, 95
 - midi.h, 102
- MIDI_RX
 - config.h, 22
 - test_midi.c, 124
- MIDI_STATUS_CHANNEL_PRESSURE
 - midi.h, 101
- MIDI_STATUS_CONTROL_CHANGE
 - midi.h, 101
- MIDI_STATUS_NOTE_OFF
 - midi.h, 101
- MIDI_STATUS_NOTE_ON
 - midi.h, 101
- MIDI_STATUS_PITCH_BEND
 - midi.h, 101
- MIDI_STATUS_POLYPHONIC_KEY_PRESSURE
 - midi.h, 101
- MIDI_STATUS_PROGRAM_CHANGE
 - midi.h, 101
- midi_status_t
 - midi.h, 100
- MIDI_TX
 - config.h, 22
 - test_midi.c, 124
- MIDI_UART
 - config.h, 22
 - test_midi.c, 125
- midi_utils.c
 - channel_pressure, 106
 - ctrl_change, 106
 - note_off, 106
 - note_on, 106
 - pitch_bend, 106
 - poly_key_pressure, 107
 - prg_change, 107
- midi_write
 - midi.c, 95
 - midi.h, 102
- miso_io
 - mcp3201_config_t, 10
- mosi_io
 - mcp3201_config_t, 10

- note_off
 - midi.h, 103
 - midi_utils.c, 106
- note_on
 - midi.h, 103
 - midi_utils.c, 106
- ongoing_transaction
 - mcp3201_context_t, 11
- output
 - fft_config_t, 6
- param1
 - midi_message_t, 17
- param2
 - midi_message_t, 17
- pitch_bend
 - midi.h, 103
 - midi_utils.c, 106
- poly_key_pressure
 - midi.h, 103
 - midi_utils.c, 107
- prg_change
 - midi.h, 104
 - midi_utils.c, 107
- processed-data.h
 - test_buffer, 41
- PROTOTYPE
 - main.c, 118
- READER_TIMEOUT_MS
 - i2s_sampler.c, 59
 - mcp3201_sampler.c, 89
- READER_TIMEOUT_TICKS
 - i2s_sampler.c, 59
 - mcp3201_sampler.c, 90
- RESAMPLE_DENOMINATOR
 - i2s_sampler.c, 59
 - mcp3201_sampler.c, 90
- rfft
 - fft.c, 67
 - fft.h, 80
- rx_io
 - midi_config_t, 14
- sampler
 - gitcon_context_t, 7
- SENSITIVITY
 - gitcon.c, 110
- setUp
 - test_fft.c, 122
 - test_midi.c, 125
- size
 - fft_config_t, 6
- spi
 - mcp3201_context_t, 11
- SPI_CS
 - config.h, 23
- SPI_DEV
 - config.h, 23
- SPI_MISO
 - config.h, 23
- SPI_MOSI
 - config.h, 23
- SPI_SCLK
 - config.h, 23
- split_radix_fft
 - fft.c, 67
 - fft.h, 80
- src/gitcon.c, 108, 111
- src/gitcon.h, 114, 117
- src/main.c, 117, 119
- status
 - midi_message_t, 17
- tearDown
 - test_fft.c, 122
 - test_midi.c, 125
- test/test_fft/test_fft.c, 120, 123
- test/test_midi/test_midi.c, 123, 126
- test_buffer
 - processed-data.h, 41
- test_fft.c
 - app_main, 122
 - AUDIO_BUFFER_SIZE, 121
 - F_SAMPLE_HZ, 121
 - FFT_SIZE, 121
 - FFT_WINDOW_SIZE, 121
 - setUp, 122
 - tearDown, 122
 - test_processed_data, 122
- test_midi.c
 - app_main, 125
 - MIDI_BAUD, 124
 - MIDI_RX, 124
 - MIDI_TX, 124
 - MIDI_UART, 125
 - setUp, 125
 - tearDown, 125
 - test_midi_bend, 125
 - test_midi_bent_note, 126
 - test_midi_blink, 126
- test_midi_bend
 - test_midi.c, 125
- test_midi_bent_note
 - test_midi.c, 126
- test_midi_blink
 - test_midi.c, 126
- test_processed_data
 - test_fft.c, 122
- twiddle_factors
 - fft_config_t, 6
- TWO_PI
 - fft.c, 64
- tx_io
 - midi_config_t, 14
- type
 - fft_config_t, 6

uart_num
 midi_config_t, [14](#)
UINT16_TO_FLOAT
 gitcon.c, [110](#)
USE_SPLIT_RADIX
 fft.c, [65](#)
USER_LOCAL_LEVEL
 main.c, [118](#)