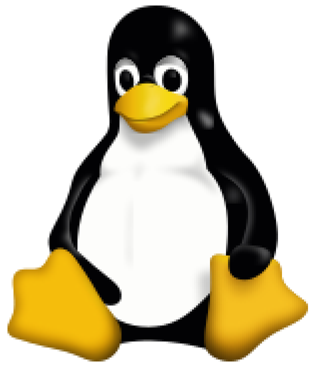
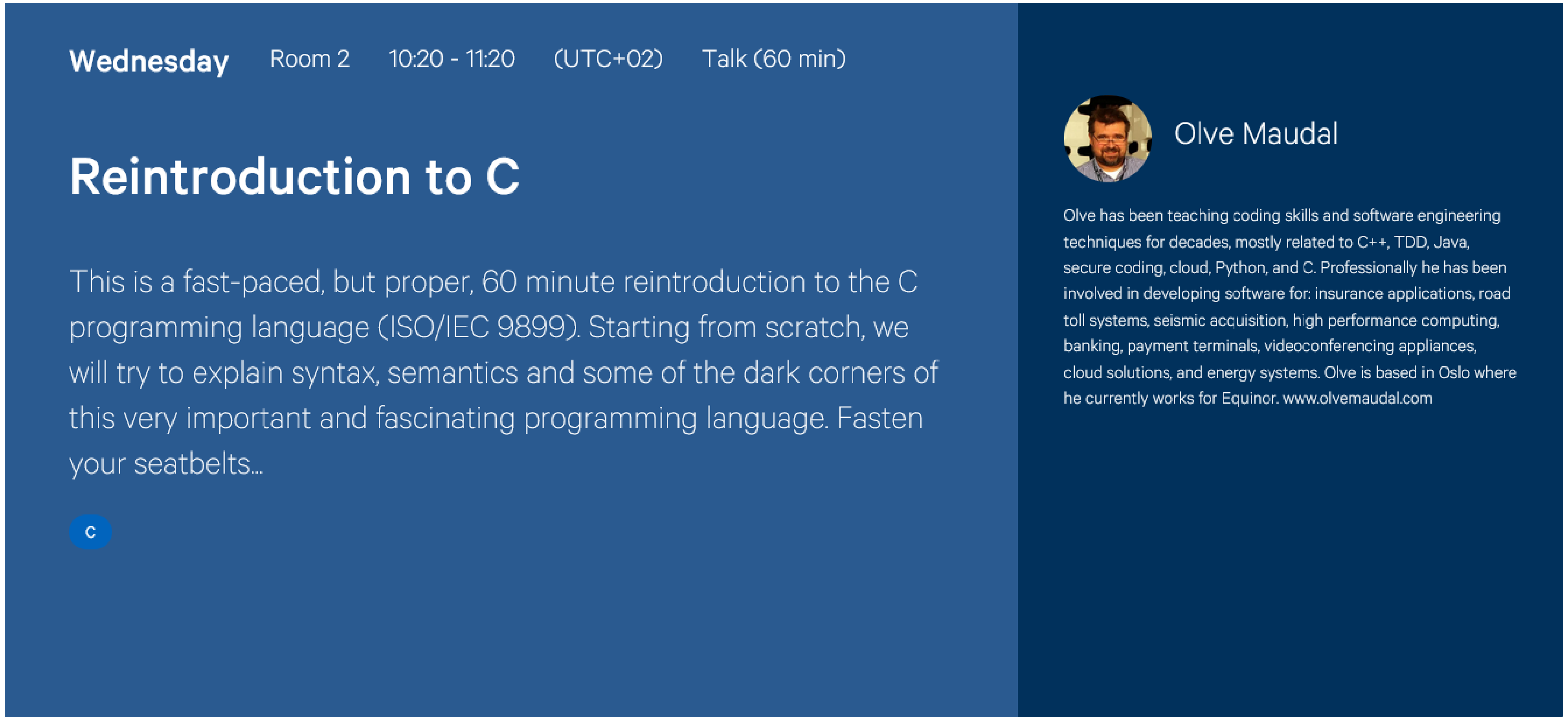
Reintroduction to

by Olve Maudal, a 60 minute session @ NDC TechTown 2023, September 20, Kongsberg, Norway.

Why C ?



NDC { TechTown }



Wednesday Room 2 10:20 - 11:20 (UTC+02) Talk (60 min)

Reintroduction to C

This is a fast-paced, but proper, 60 minute reintroduction to the C programming language (ISO/IEC 9899). Starting from scratch, we will try to explain syntax, semantics and some of the dark corners of this very important and fascinating programming language. Fasten your seatbelts...

C

Olve Maudal

Olve has been teaching coding skills and software engineering techniques for decades, mostly related to C++, TDD, Java, secure coding, cloud, Python, and C. Professionally he has been involved in developing software for: insurance applications, road toll systems, seismic acquisition, high performance computing, banking, payment terminals, videoconferencing appliances, cloud solutions, and energy systems. Olve is based in Oslo where he currently works for Equinor. www.olvemaudal.com

https://ndctechtown.com

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
Hello
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>


int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>

// your stuff ...
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

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

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

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

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
#include <stdio.h>

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
#include <stdio.h>

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
//
// relevant stuff from stdio.h (often found in /usr/include)
//
extern int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
//
// relevant stuff from stdio.h (often found in /usr/include)
//
extern int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
//
// relevant stuff from stdio.h (often found in /usr/include)
//
extern int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
//
// relevant stuff from stdio.h (often found in /usr/include)
//
extern int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
//
// relevant stuff from stdio.h (often found in /usr/include)
//
extern int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == (-1))
        return 1;
    return 0;
}
```

```c
//
// relevant stuff from stdio.h (often found in /usr/include)
//
extern int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == (-1))
        return 1;
    return 0;
}
```

```
//
// relevant stuff from stdio.h (often found in /usr/include)
//
extern int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == (-1))
        return 1;
    return 0;
}
```

```c
//
// relevant stuff from stdio.h (often found in /usr/include)
//
int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == (-1))
        return 1;
    return 0;
}
```

```c
//
// relevant stuff from stdio.h (often found in /usr/include)
//
int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == (-1))
        return 1;
    return 0;
}
```

```
//
// relevant stuff from stdio.h (often found in /usr/include)
//
int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == (-1))
        return 1;
    return 0;
}
```

```
//
// relevant stuff from stdio.h (often found in /usr/include)
//
int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
//
// relevant stuff from stdio.h (often found in /usr/include)
//
int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
//
// relevant stuff from stdio.h (often found in /usr/include)
//
int puts(const char *s);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (puts("Hello") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
//
// relevant stuff from stdio.h (often found in /usr/include)
//
int printf(const char * restrict format, ...);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (printf("Hello\n") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```c
//
// relevant stuff from stdio.h (often found in /usr/include)
//
int printf(const char * restrict format, ...);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (printf("Hello\n") == EOF)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
//
// relevant stuff from stdio.h (often found in /usr/include)
//
int printf(const char * restrict format, ...);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (printf("Hello\n") != 6)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
//
// relevant stuff from stdio.h (often found in /usr/include)
//
int printf(const char * restrict format, ...);
#define EOF (-1)

/*
 * relevant stuff from stdlib.h (often found in /usr/include)
 */
#define EXIT_SUCCESS 0
#define EXIT_FAILURE 1

/* your stuff ... */
int main(void)
{
    if (printf("Hello\n") != 6)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

```
Hello
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int ret_code = printf("Hello\n");
    if (ret_code == EOF)
       if (ferror(stdout)) {
          fprintf(stderr, "printf() failed in file %s at line # %d\n", __FILE__,__LINE__-3);
          perror("printf()");
          exit(EXIT_FAILURE);
       }
    if (ret_code != 6) {
        fprintf(stderr, "Unexpected return value: %d\n", ret_code);
        exit(2);
    }
    return EXIT_SUCCESS;
}
```

```
Hello
```

```c
#include <stdio.h>

int main(void)
{
    printf("Hello\n");
    return 0;
}
```

```c
#include <stdio.h>

int main(void)
{
    printf("Hello\n");
}
```

```c
#include <stdio.h>

int main(void)
{
    printf("Hello\n");
}
```

```
Hello
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    int letter_H = 0x48;   // hexadecimal constant
    int letter_E = 101;    // decimal constant
    int letter_L = 0154;   // octal constant
    int letter_O = 'o';    // character constant (aka character literal)
    int cntrl_NL = '\n';   // simple escape sequence, representing ASCII 0x0a

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)

int main(void)
{
    int letter_H = 0x48;   // hexadecimal constant
    int letter_E = 101;    // decimal constant
    int letter_L = 0154;   // octal constant
    int letter_O = 'o';    // character constant (aka character literal)
    int cntrl_NL = '\n';   // simple escape sequence, representing ASCII 0x0a

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    int letter_H = 0x48;   // hexadecimal constant
    int letter_E = 101;    // decimal constant
    int letter_L = 0154;   // octal constant
    int letter_O = 'o';    // character constant (aka character literal)
    int cntrl_NL = '\n';   // simple escape sequence, representing ASCII 0x0a

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    int letter_H = 0x48;   // hexadecimal constant
    int letter_E = 101;    // decimal constant
    int letter_L = 0154;   // octal constant
    int letter_O = 'o';    // character constant (aka character literal)
    int cntrl_NL = '\n';   // simple escape sequence, representing ASCII 0x0a

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```
Hello
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    int letter_H = 0x48;   // hexadecimal constant
    int letter_E = 101;    // decimal constant
    int letter_L = 0154;   // octal constant
    int letter_O = 'o';    // character constant (aka character literal)
    int cntrl_NL = '\n';   // simple escape sequence, representing ASCII 0x0a

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    int letter_H = 0x48;   // hexadecimal constant
    int letter_E = 101;    // decimal constant
    int letter_L = 0154;   // octal constant
    int letter_O = 'o';    // character constant (aka character literal)
    int cntrl_NL = '\n';   // simple escape sequence, representing ASCII 0x0a

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    unsigned long long int letter_H =  72;  /* at least 64 bit. */
    unsigned long int      letter_E = 101;  /* at least 32 bit  */
    unsigned int           letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    unsigned short int     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    unsigned char          cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    unsigned long long int letter_H =  72;  /* at least 64 bit. */
    unsigned long int      letter_E = 101;  /* at least 32 bit  */
    unsigned int           letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    unsigned short int     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    unsigned char          cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    unsigned long long int letter_H =  72;  /* at least 64 bit. */
    unsigned long int      letter_E = 101;  /* at least 32 bit  */
    unsigned int           letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    unsigned short int     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    unsigned char          cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```
Hello
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    unsigned long long int letter_H =  72;  /* at least 64 bit. */
    unsigned long int      letter_E = 101;  /* at least 32 bit  */
    unsigned int           letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    unsigned short int     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    unsigned char          cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```
#include <stdio.h>   // int putchar(int)


int main(void)
{
    signed long long int letter_H =  72;  /* at least 64 bit. */
    signed long int      letter_E = 101;  /* at least 32 bit  */
    signed int           letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    signed short int     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    signed char          cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    signed long long int letter_H =  72;  /* at least 64 bit. */
    signed long int      letter_E = 101;  /* at least 32 bit  */
    signed int           letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    signed short int     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    signed char          cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```
#include <stdio.h>   // int putchar(int)


int main(void)
{
    signed long long int letter_H =  72;  /* at least 64 bit. */
    signed long int      letter_E = 101;  /* at least 32 bit  */
    signed int           letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    signed short int     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    signed char          cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    long long letter_H =  72;  /* at least 64 bit. */
    long      letter_E = 101;  /* at least 32 bit  */
    int       letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    short     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    char      cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    long long letter_H =  72;  /* at least 64 bit. */
    long      letter_E = 101;  /* at least 32 bit  */
    int       letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    short     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    char      cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    long long letter_H =  72;  /* at least 64 bit. */
    long      letter_E = 101;  /* at least 32 bit  */
    int       letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    short     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    char      cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```
Hello
```

```
#include <stdio.h>   // int putchar(int)


int main(void)
{
    long long letter_H =  72;  /* at least 64 bit. */
    long      letter_E = 101;  /* at least 32 bit  */
    int       letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    short     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    char      cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)


int main(void)
{
    long long letter_H =  72;  /* at least 64 bit. */
    long      letter_E = 101;  /* at least 32 bit  */
    int       letter_L = 108;  /* at least 16 bit. Could be, say, 36 bits... */
    short     letter_O = 111;  /* at least 16 bit. Could be, say, 24 bits... */
    char      cntrl_NL =  10;  /* at least 8 bit. Could be, say, 9 or 32 ... */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```
#include <stdio.h>   // int putchar(int)
#include <complex.h>

int main(void)
{
    long double         letter_H =  72.0L;      /* often 128 bits, but at least 80 bits */
    double              letter_E = 101.0;       /* often 64 bits */
    float               letter_L = 108.f;       /* often 32 bits */
    complex             letter_O = 111.  + 0*I; /* often 128 bits (2*64) */
    long double complex cntrl_NL =  10.0 + 0*I; /* often 256 bits, but at least 160 bits */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```
#include <stdio.h>   // int putchar(int)
#include <complex.h>

int main(void)
{
    long double          letter_H =  72.0L;      /* often 128 bits, but at least 80 bits */
    double               letter_E = 101.0;       /* often 64 bits */
    float                letter_L = 108.f;       /* often 32 bits */
    complex              letter_O = 111.  + 0*I; /* often 128 bits (2*64) */
    long double complex cntrl_NL =  10.0 + 0*I; /* often 256 bits, but at least 160 bits */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```c
#include <stdio.h>   // int putchar(int)
#include <complex.h>

int main(void)
{
    long double         letter_H =  72.0L;      /* often 128 bits, but at least 80 bits */
    double              letter_E = 101.0;       /* often 64 bits */
    float               letter_L = 108.f;       /* often 32 bits */
    complex             letter_O = 111.  + 0*I; /* often 128 bits (2*64) */
    long double complex cntrl_NL =  10.0 + 0*I; /* often 256 bits, but at least 160 bits */

    putchar(letter_H);
    putchar(letter_E);
    putchar(letter_L);
    putchar(letter_L);
    putchar(letter_O);
    putchar(cntrl_NL);
}
```

```
Hello
```

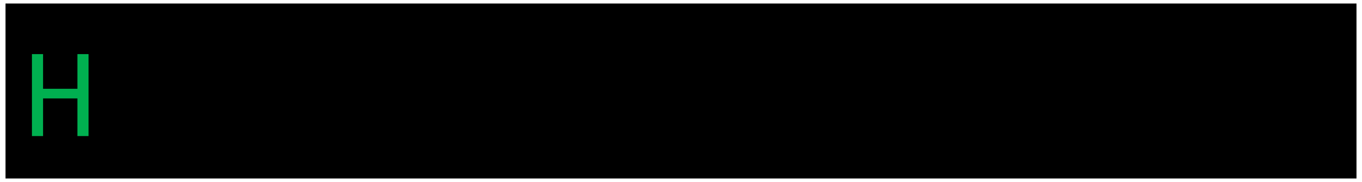
```c
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    putchar(a[0]);
    putchar(a[1]);
    putchar(a[2]);
    putchar(a[3]);
    putchar(a[4]);
    putchar(a[5]);
}
```

```c
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    putchar(a[0]);
    putchar(a[1]);
    putchar(a[2]);
    putchar(a[3]);
    putchar(a[4]);
    putchar(a[5]);
}
```

```
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{






}
```

[redacted]

```
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    putchar(*a);





}
```



```
H
```

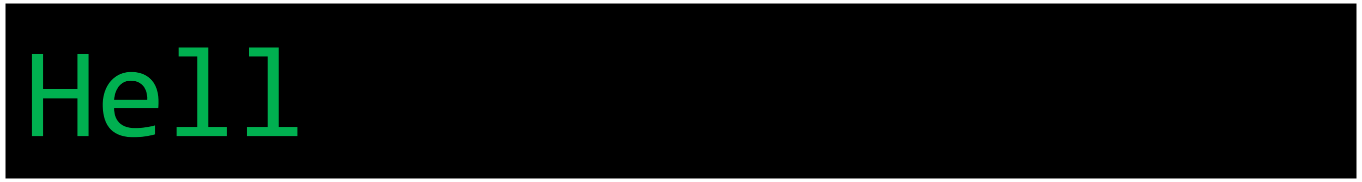
```
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    putchar(*a);
    putchar(a[1]);




}
```

```
He
```

```c
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    putchar(*a);
    putchar(a[1]);
    putchar(*(a + 2));



}
```

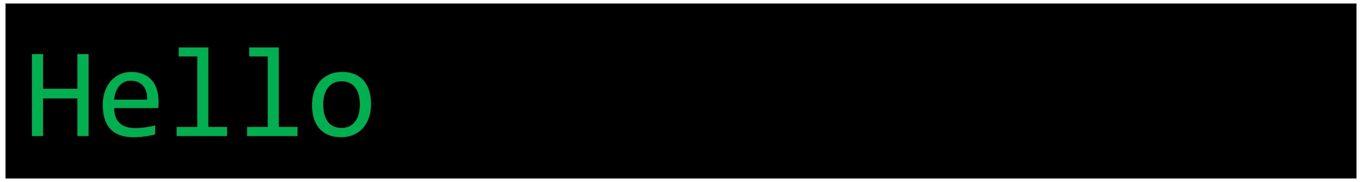
```
Hel
```

```c
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    putchar(*a);
    putchar(a[1]);
    putchar(*(a + 2));
    putchar(*(3 + a));


}
```



```
Hell
```

```
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    putchar(*a);
    putchar(a[1]);
    putchar(*(a + 2));
    putchar(*(3 + a));
    putchar(4[a]);

}
```



```
Hello
```

```c
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    putchar(*a);
    putchar(a[1]);
    putchar(*(a + 2));
    putchar(*(3 + a));
    putchar(4[a]);
    putchar(*(&a[2] + 3));
}
```

```
Hello
```

```
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{






}
```

[redacted]

```
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    const int * p = a + 6;







}
```

[redacted]

```c
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    const int * p = a + 6;




}
```

```
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    const int * p = a + 6;





}
```

[redacted]

```c
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    const int * p = a + 6;
    putchar(p[-6]);





}
```

```
H
```

```c
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    const int * p = a + 6;
    putchar(p[-6]);
    putchar(*(p - 5));




}
```

```
He
```

```
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    const int * p = a + 6;
    putchar(p[-6]);
    putchar(*(p - 5));
    putchar(*(-4 + p));



}
```

```
Hel
```

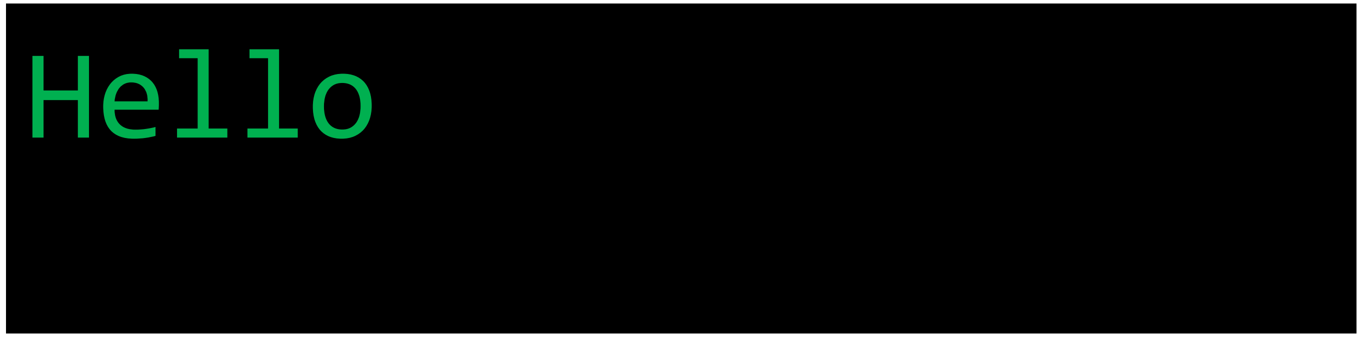
```
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    const int * p = a + 6;
    putchar(p[-6]);
    putchar(*(p - 5));
    putchar(*(-4 + p));
    putchar((-3)[p]);


}
```

```
Hell
```

```c
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    const int * p = a + 6;
    putchar(p[-6]);
    putchar(*(p - 5));
    putchar(*(-4 + p));
    putchar((-3)[p]);
    putchar(*&p[-2]);

}
```

```
Hello
```

```c
extern int putchar(int);

static int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

int main(void)
{
    const int * p = a + 6;
    putchar(p[-6]);
    putchar(*(p - 5));
    putchar(*(-4 + p));
    putchar((-3)[p]);
    putchar(*&p[-2]);
    putchar(*&*(p - 1));
}
```



```
Hello
```

```
#include <stdio.h>
#include <stddef.h>  // typedef <some implementation-defined unsigned integer type> size_t

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
    putchar(a[i]);
    ++i;
    putchar(a[i]);
    ++i;
    putchar(a[i]);
    ++i;
    putchar(a[i]);
    ++i;
    putchar(a[i]);
    ++i;
    putchar(a[i]);
    ++i;
}
```

```
#include <stdio.h>
#include <stddef.h>  // typedef <some implementation-defined unsigned integer type> size_t

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
    putchar(a[i]);
    ++i;
    putchar(a[i]);
    ++i;
    putchar(a[i]);
    ++i;
    putchar(a[i]);
    ++i;
    putchar(a[i]);
    ++i;
    putchar(a[i]);
    ++i;
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
again:
    putchar(a[i]);
    ++i;
    if (i < 6)
        goto again;
}
```

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

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
again: {
        putchar(a[i]);
        ++i;
    } if (i < 6)
        goto again;
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
    again: {
        putchar(a[i]);
        ++i;
    } if (i < 6) goto again;
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
    do {
        putchar(a[i]);
        ++i;
    } while (i < 6);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
    while (i < 6) {
        putchar(a[i]);
        ++i;
    }
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
    while (i < 6) {
        putchar(a[i]);
        ++i;
    }
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
    while (i < 6) {
        putchar(a[i]);
        ++i;
    }
}
```

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

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
    for (            ; i < 6;    ) {
        putchar(a[i]);
        ++i;
    }
}
```

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

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t i = 0;
    for (            ; i < 6;    ) {
        putchar(a[i]);
        ++i;
    }
}
```

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

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

    for (size_t i = 0; i < 6;    ) {
        putchar(a[i]);
        ++i;
    }
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

    for (size_t i = 0; i < 6;    ) {
        putchar(a[i]);
        ++i;
    }
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

    for (size_t i = 0; i < 6; ++i) {
        putchar(a[i]);

    }
}
```

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

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

    for (size_t i = 0; i < 6; ++i) {
        putchar(a[i]);

    }
}
```

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

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

    for (size_t i = 0; i < 6; ++i) {
        putchar(a[i]);
    }
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

    for (size_t i = 0; i < 6; ++i) {
        putchar(a[i]);
    }
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

    for (size_t i = 0; i < 6; ++i)
        putchar(a[i]);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};

    for (size_t i = 0; i < 6; ++i)
        putchar(a[i]);
}
```

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

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    for (size_t i = 0; i < n; ++i)
        putchar(a[i]);
}
```

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

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    for (size_t i = 0; i < n; ++i)
        putchar(a[i]);
}
```

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

int main(void)
{
    int a[6] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    for (size_t i = 0; i < n; ++i)
        putchar(a[i]);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    for (size_t i = 0; i < n; ++i)
        putchar(a[i]);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    for (size_t i = 0; i < n; ++i)
        putchar(a[i]);
}
```

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


int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    for (size_t i = 0; i < n; ++i)
        putchar(a[i]);
}
```

```c
#include <stdio.h>
#include <stdlib.h>
```

```c
int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    for (size_t i = 0; i < n; ++i)
        putchar(a[i]);
}
```

```c
#include <stdio.h>
#include <stdlib.h>








int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    for (size_t i = 0; i < n; ++i)
        putchar(a[i]);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * s, size_t n)
{
    for (size_t i = 0; i < n; ++i)
        putchar(s[i]);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, n);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * s, size_t n)
{
    for (size_t i = 0; i < n; ++i)
        putchar(s[i]);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, n);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * s, size_t n)
{
    for (size_t i = 0; i < n; ++i)
        putchar(s[i]);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, n);
}
```

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

static void myputchars(const int * s, size_t n)
{
    const int * begin = s;
    const int * end = s + n;
    for (const int * it = begin; it != end; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, n);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * s, size_t n)
{
    const int * begin = s;
    const int * end = s + n;
    for (const int * it = begin; it != end; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, n);
}
```

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

static void myputchars(const int * s, size_t n)
{
    const int * begin = s;
    const int * end = s + n;
    for (const int * it = begin; it != end; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, n);
}
```

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

static void myputchars(const int * begin, const int * end)
{
    for (const int * it = begin; it != end; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, a + n);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * begin, const int * end)
{
    for (const int * it = begin; it != end; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, a + n);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * begin, const int * end)
{
    for (const int * it = begin; it != end; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, a + n);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * begin, const int * end)
{
    for (const int * it = begin; it != end; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, a + n);
}
```

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

static void myputchars(const int * begin, const int * end)
{
    for (const int * it = begin; it != end; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, a + n);
}
```

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

static void myputchars(const int * begin, const int * end)
{
    for (const int * it = begin; *it != 0x00; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, a + n);
}
```

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

static void myputchars(const int * begin, const int * end)
{
    for (const int * it = begin; *it; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, a + n);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * begin, const int * end)
{
    for (const int * it = begin; *it; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a, a + n);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * begin)
{
    for (const int * it = begin; *it; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a);
}
```

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

static void myputchars(const int * begin)
{
    for (const int * it = begin; *it; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    size_t n = sizeof a / sizeof a[0];
    myputchars(a);
}
```

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

static void myputchars(const int * begin)
{
    for (const int * it = begin; *it; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * begin)
{
    for (const int * it = begin; *it; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

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

static void myputchars(const int * it)
{
    for (                      ; *it; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

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

static void myputchars(const int * it)
{
    for (                      ; *it; ++it)
        putchar(*it);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * it)
{
    for (                          ; *it;      )
        putchar(*it++);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * it)
{
    for (                      ; *it;     )
        putchar(*it++);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

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

static void myputchars(const int * it)
{
    for (                             ; *it;      )
        putchar(*it++);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

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

static void myputchars(const int * it)
{
    while (*it)
        putchar(*it++);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

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

static void myputchars(const int * it)
{
    while (*it)
        putchar(*it++);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputchars(const int * it)
{
    while (*it)
        putchar(*it++);
}

int main(void)
{
    int a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

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

static void myputchars(const char * it)
{
    while (*it)
        putchar(*it++);
}

int main(void)
{
    char a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

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

static void myputchars(const char * it)
{
    while (*it)
        putchar(*it++);
}

int main(void)
{
    char a[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x0a, 0x00};
    myputchars(a);
}
```

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

static void myputchars(const char * it)
{
    while (*it)
        putchar(*it++);
}

int main(void)
{
    char a[] = {'H', 'e', 'l', 'l', 'o', '\n', '\0'};
    myputchars(a);
}
```

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

static void myputchars(const char * it)
{
    while (*it)
        putchar(*it++);
}

int main(void)
{
    char a[] = "Hello\n"; // {'H', 'e', 'l', 'l', 'o', '\n', '\0'};
    myputchars(a);
}
```

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

static void myputchars(const char * it)
{
    while (*it)
        putchar(*it++);
}

int main(void)
{
    char a[] = "Hello\n";
    myputchars(a);
}
```

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

static void myputchars(const char * it)
{
    while (*it)
        putchar(*it++);
}

int main(void)
{
    char a[] = "Hello\n";
    myputchars(a);
}
```

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

static void myputchars(const char * it)
{
    while (*it)
        putchar(*it++);
    putchar('\n');
}

int main(void)
{
    char a[] = "Hello";
    myputchars(a);
}
```

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

static void myputchars(const char * it)
{
    while (*it)
        putchar(*it++);
    putchar('\n');
}

int main(void)
{
    char a[] = "Hello";
    myputchars(a);
}
```

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

static void myputs(const char * it)
{
    while (*it)
        putchar(*it++);
    putchar('\n');
}

int main(void)
{
    char a[] = "Hello";
    myputs(a);
}
```

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

static void myputs(const char * it)
{
    while (*it)
        putchar(*it++);
    putchar('\n');
}

int main(void)
{
    char a[] = "Hello";
    myputs(a);
}
```

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

static void myputs(const char * s)
{
    while (*s)
        putchar(*s++);
    putchar('\n');
}

int main(void)
{
    char a[] = "Hello";
    myputs(a);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputs(const char * s)
{
    while (*s)
        putchar(*s++);
    putchar('\n');
}

int main(void)
{
    char a[] = "Hello";
    myputs(a);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputs(const char * s)
{
    while (*s)
        putchar(*s++);
    putchar('\n');
}

int main(void)
{
    char a[] = "Hello";
    myputs(a);
}
```

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

static void myputs(const char * s)
{
    while (*s)
        putchar(*s++);
    putchar('\n');
}

int main(void)
{
    myputs("Hello");
}
```

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

static void myputs(const char * s)
{
    while (*s)
        putchar(*s++);
    putchar('\n');
}

int main(void)
{
     puts("Hello");
}
```

```c
#include <stdio.h>
#include <stdlib.h>

static void myputs(const char * s)
{
    while (*s)
        putchar(*s++);
    putchar('\n');
}

int main(void)
{
     puts("Hello");
}
```

```c
#include <stdio.h>
#include <stdlib.h>


int main(void)
{
     puts("Hello");
}
```

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
     puts("Hello");
}
```

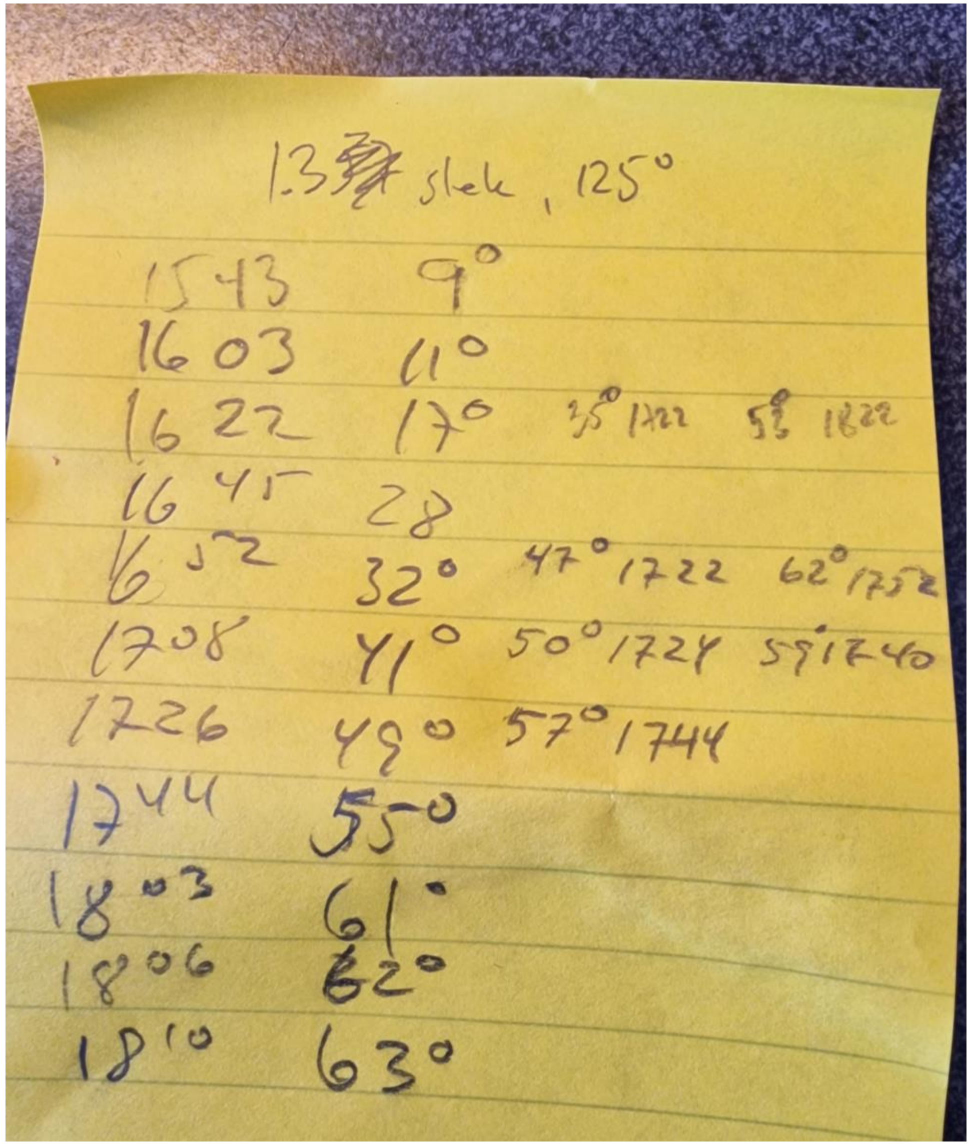
```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
     puts("Hello");
}
```

```c
#include <stdio.h>

int main(void)
{
     puts("Hello");
}
```

```c
#include <stdio.h>

int main(void)
{
     puts("Hello");
}
```

```c
#include <stdio.h>

int main(void)
{
     puts("Hello");
}
```

```
Hello
```

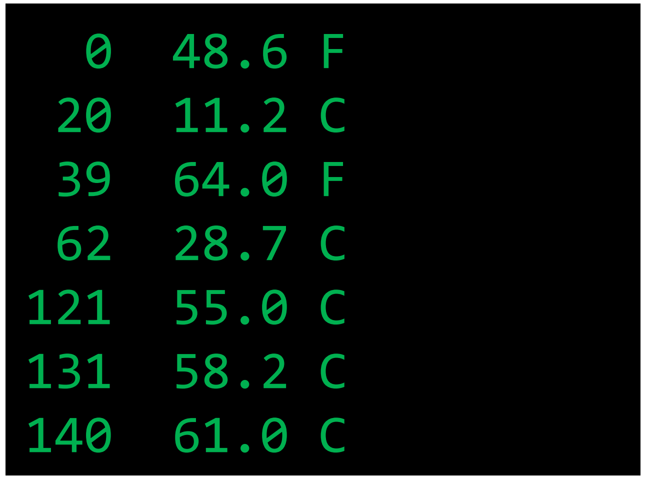


1.3[illegible] stek , 125°

1543	9°		
1603	11°		
1622	17°	35° 1722	53° 1822
1645	28		
1652	32°	47° 1722	62° 1752
1708	41°	50° 1724	59° 1740
1726	49°	57° 1744	
1744	55°		
1803	61°		
1806	62°		
1810	63°		

```c
#include <stdio.h>

struct tempsample {
    int time;
    double temp;
    char scale;
};

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```

```c
#include <stdio.h>

struct tempsample {
    int time;
    double temp;
    char scale;
};

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```



```
  0  48.6 F
 20  11.2 C
 39  64.0 F
 62  28.7 C
121  55.0 C
131  58.2 C
140  61.0 C
```

```c
#include <stdio.h>

struct tempsample {
    int time;
    double temp;
    char scale;
};

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i) {
        if (samples[i].scale != 'C')
            continue;
        if (samples[i].time > 131)
            break;
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
    }
}
```

```
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i) {
        if (samples[i].scale != 'C')
            continue;
        if (samples[i].time > 131)
            break;
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
    }
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i) {
        if (samples[i].scale != 'C')
            continue;
        if (samples[i].time > 131)
            break;
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
    }
}
```

```
 20  11.2 C
 62  28.7 C
121  55.0 C
131  58.2 C
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i) {
        if (samples[i].scale != 'C')
            continue;
        if (samples[i].time > 131)
            break;
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
    }
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i) {
        if (samples[i].scale != 'C')
            continue;
        if (samples[i].time > 131)
            break;
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
    }
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
    }
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
        {131, 58.2, 'C'},
        {140, 61.0, 'C'}
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
    }
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```

```
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 39, 64.0, 'F'},
        { 62, 28.7, 'C'},
        {121, 55.0, 'C'},
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 62, 28.7, 'C'},
        { 39, 64.0, 'F'},
        {121, 55.0, 'C'},
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```

```
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        {  0, 48.6, 'F'},
        { 20, 11.2, 'C'},
        { 62, 28.7, 'C'},
        { 39, 64.0, 'F'},
        {121, 55.0, 'C'},
    };

    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```

```
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        [0] = {.time =   0, .temp = 48.6, .scale = 'F'},
        [1] = {.time =  20, .temp = 11.2, .scale = 'C'},
        [3] = {.time =  62, .temp = 28.7, .scale = 'C'},
        [2] = {.time =  39, .temp = 64.0, .scale = 'F'},
        [4] = {.time = 121, .temp = 55.0, .scale = 'C'},
    };
    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```

```
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

int main(void)
{
    struct tempsample samples[] = {
        [0] = {.time =   0, .temp = 48.6, .scale = 'F'},
        [1] = {.time =  20, .temp = 11.2, .scale = 'C'},
        [3] = {.time =  62, .temp = 28.7, .scale = 'C'},
        [2] = {.time =  39, .temp = 64.0, .scale = 'F'},
        [4] = {.time = 121, .temp = 55.0, .scale = 'C'},
    };
    size_t nsamples = sizeof samples / sizeof *samples;
    for (size_t i = 0; i < nsamples; ++i)
        printf("%3d %5.1lf %c\n", samples[i].time, samples[i].temp, samples[i].scale);
}
```

```
  0  48.6 F
 20  11.2 C
 39  64.0 F
 62  28.7 C
121  55.0 C
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void printsample(struct tempsample sample)
{
    printf("%3d %5.1lf %c\n", sample.time, sample.temp, sample.scale);
}

int main(void)
{
    printsample((struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'});
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void printsample(struct tempsample sample)
{
    printf("%3d %5.1lf %c\n", sample.time, sample.temp, sample.scale);
}

int main(void)
{
    printsample((struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'});
}
```

```
 39  17.8 C
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void printsample(struct tempsample sample)
{
    printf("%3d %5.1lf %c\n", sample.time, sample.temp, sample.scale);
}

int main(void)
{
    printsample((struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'});
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void printsample(struct tempsample sample)
{
    printf("%3d %5.1lf %c\n", sample.time, sample.temp, sample.scale);
}

int main(void)
{
    printsample((struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'});
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void printsample(struct tempsample * sample)
{
    printf("%3d %5.1lf %c\n", (*sample).time, (*sample).temp, (*sample).scale);
}

int main(void)
{
    printsample(&(struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'})
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void printsample(struct tempsample * sample)
{
    printf("%3d %5.1lf %c\n", (*sample).time, (*sample).temp, (*sample).scale);
}

int main(void)
{
    printsample(&(struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'})
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void printsample(struct tempsample * sample)
{
    printf("%3d %5.1lf %c\n", sample->time, sample->temp, sample->scale);
}

int main(void)
{
    printsample(&(struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'})
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void printsample(struct tempsample * sample)
{
    printf("%3d %5.1lf %c\n", sample->time, sample->temp, sample->scale);
}

int main(void)
{
    printsample(&(struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'})
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void printsample(const struct tempsample * sample)
{
    printf("%3d %5.1lf %c\n", sample->time, sample->temp, sample->scale);
}

int main(void)
{
    printsample(&(struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'})
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void printsample(const struct tempsample * sample)
{
    printf("%3d %5.1lf %c\n", sample->time, sample->temp, sample->scale);
}

int main(void)
{
    printsample(&(struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'})
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * sample)
{
    printf("%3d %5.1lf %c\n", sample->time, sample->temp, sample->scale);
}

int main(void)
{
    tempsample_print(&(struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'});
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * sample)
{
    printf("%3d %5.1lf %c\n", sample->time, sample->temp, sample->scale);
}

int main(void)
{
    tempsample_print(&(struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'});
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * sample)
{
    printf("%3d %5.1lf %c\n", sample->time, sample->temp, sample->scale);
}

int main(void)
{
    tempsample_print(&(struct tempsample){.time = 39, .temp = 17.8, .scale = 'C'});
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    if (sample->scale == 'C') {
        temp = sample->temp;
    } else if (sample->scale == 'F') {
        temp = (sample->temp - 32) * 5 / 9;
    } else {
        printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    printf("%.2lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    if (sample->scale == 'C') {
        temp = sample->temp;
    } else if (sample->scale == 'F') {
        temp = (sample->temp - 32) * 5 / 9;
    } else {
        printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    printf("%.2lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    if (sample->scale == 'C') {
        temp = sample->temp;
    } else if (sample->scale == 'F') {
        temp = (sample->temp - 32) * 5 / 9;
    } else {
        printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    printf("%.2lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```
11.28 C
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    if (sample->scale == 'C') {
        temp = sample->temp;
    } else if (sample->scale == 'F') {
        temp = (sample->temp - 32) * 5 / 9;
    } else {
        printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    printf("%.2lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    switch (sample->scale) {
        case 'C': temp = sample->temp; break;
        case 'F': temp = (sample->temp - 32) * 5 / 9; break;
        case 'R':
        case 'K': printf("Feature not implemented\n"); break;
        default:  printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    printf("%.2lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    switch (sample->scale) {
        case 'C': temp = sample->temp; break;
        case 'F': temp = (sample->temp - 32) * 5 / 9; break;
        case 'R': 
        case 'K': printf("Feature not implemented\n"); break;
        default:  printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    printf("%.2lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    switch (sample->scale) {
        case 'C': temp = sample->temp; break;
        case 'F': temp = (sample->temp - 32) * 5 / 9; break;
        case 'R': /* fallthrough */
        case 'K': printf("Feature not implemented\n"); break;
        default:  printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    printf("%.2lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    switch (sample->scale) {
        case 'C': temp = sample->temp; break;
        case 'F': temp = (sample->temp - 32) * 5 / 9; break;
        case 'R': 
        case 'K': printf("Feature not implemented\n"); break;
        default:  printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    printf("%.2lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    switch (sample->scale) {
        case 'C': temp = sample->temp; break;
        case 'F': temp = (sample->temp - 32) * 5 / 9; break;
        case 'R':
        case 'K': printf("Feature not implemented\n"); break;
        default:  printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    printf("%.2lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    switch (sample->scale) {
        case 'C': temp = sample->temp; break;
        case 'F': temp = (sample->temp - 32) * 5 / 9; break;
        case 'R':
        case 'K': printf("Feature not implemented\n"); break;
        default:  printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    printf("%.2lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```
#include <stdio.h>

enum tempscale { CELSIUS, FAHRENHEIT, KELVIN, RANKINE };

struct tempsample { int time; double temp; enum tempscale scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    switch (sample->scale) {
        case CELSIUS:    temp = sample->temp; break;
        case FAHRENHEIT: temp = (sample->temp - 32) * 5 / 9; break;
        case KELVIN:
        case RANKINE:    printf("Feature not implemented\n"); break;
        default:         printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = FAHRENHEIT};
    printf("%.1lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```c
#include <stdio.h>

enum tempscale { CELSIUS, FAHRENHEIT, KELVIN, RANKINE };

struct tempsample { int time; double temp; enum tempscale scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    switch (sample->scale) {
        case CELSIUS:    temp = sample->temp; break;
        case FAHRENHEIT: temp = (sample->temp - 32) * 5 / 9; break;
        case KELVIN:
        case RANKINE:    printf("Feature not implemented\n"); break;
        default:         printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = FAHRENHEIT};
    printf("%.1lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```c
#include <stdio.h>

enum tempscale { CELSIUS, FAHRENHEIT, KELVIN, RANKINE };

struct tempsample { int time; double temp; enum tempscale scale; };

double tempsample_temp_in_celcisus(const struct tempsample * sample)
{
    double temp = 0.0;
    switch (sample->scale) {
        case CELSIUS:    temp = sample->temp; break;
        case FAHRENHEIT: temp = (sample->temp - 32) * 5 / 9; break;
        case KELVIN:
        case RANKINE:    printf("Feature not implemented\n"); break;
        default:         printf("Huh?\n");
    }
    return temp;
}

int main(void)
{
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = FAHRENHEIT};
    printf("%.1lf C\n", tempsample_temp_in_celcisus(&sample));
}
```

```
11.3 C
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static void demo(void) {
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    tempsample_print(&sample);
}

int main(void) {
    demo();
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static void demo(void) {
    struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    tempsample_print(&sample);
}

int main(void) {
    demo();
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};

static void demo(void) {
    tempsample_print(&sample);
}

int main(void) {
    demo();
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};

static void demo(void) {
    tempsample_print(&sample);
}

int main(void) {
    demo();
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};

static void demo(void) {
    tempsample_print(&sample);
}

int main(void) {
    demo();
}
```

```
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};

static void demo(void) {
    tempsample_print(&sample);
}

int main(void) {
    demo();
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};

static void demo(void) {
    tempsample_print(&sample);
}

int main(void) {
    demo();
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};

static void demo(void) {
    tempsample_print(&sample);
}

int main(void) {
    demo();
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static void demo(void) {
    static struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    tempsample_print(&sample);
}

int main(void) {
    demo();
}
```

```
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static void demo(void) {
    static struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    tempsample_print(&sample);
}

int main(void) {
    demo();
}
```

```
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static struct tempsample * demo(void) {
    static struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    return &sample;
}

int main(void) {
    struct tempsample * sample = demo();
    tempsample_print(sample);
}
```

```c
#include <stdio.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static struct tempsample * demo(void) {
    static struct tempsample sample = {.time = 20, .temp = 52.3, .scale = 'F'};
    return &sample;
}

int main(void) {
    struct tempsample * sample = demo();
    tempsample_print(sample);
}
```

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

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static struct tempsample * demo(void) {
    struct tempsample * sample = malloc(sizeof sample);
    if (sample == NULL)
        exit(EXIT_FAILURE);
    *sample = (struct tempsample){.time = 20, .temp = 52.3, .scale = 'F'};
    return sample;
}

int main(void) {
    struct tempsample * sample = demo();
    tempsample_print(sample);
    free(sample);
}
```

```c
#include <stdio.h>
#include <stdlib.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static struct tempsample * demo(void) {
    struct tempsample * sample = malloc(sizeof sample);
    if (sample == NULL)
        exit(EXIT_FAILURE);
    *sample = (struct tempsample){.time = 20, .temp = 52.3, .scale = 'F'};
    return sample;
}

int main(void) {
    struct tempsample * sample = demo();
    tempsample_print(sample);
    free(sample);
}
```

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

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static void demo(void) {
    struct tempsample * sample = malloc(sizeof sample);
    if (sample == NULL)
        exit(EXIT_FAILURE);
    *sample = (struct tempsample){.time = 20, .temp = 52.3, .scale = 'F'};
    tempsample_print(sample);
    free(sample);
}

int main(void) {
    demo();
}
```

```c
#include <stdio.h>
#include <stdlib.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static void demo(void) {
    struct tempsample * sample = malloc(sizeof sample);
    if (sample == NULL)
        exit(EXIT_FAILURE);
    *sample = (struct tempsample){.time = 20, .temp = 52.3, .scale = 'F'};
    tempsample_print(sample);
    free(sample);
}

int main(void) {
    demo();
}
```

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

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static void demo(void) {
    struct tempsample * sample = malloc(sizeof sample);
    // do stuff... goto cleanup after dealing with error
cleanup:
    // clean up stuff...
    free(sample);
}

int main(void) {
    demo();
}
```

```c
#include <stdio.h>
#include <stdlib.h>

struct tempsample { int time; double temp; char scale; };

static void tempsample_print(const struct tempsample * s) {
    printf("%3d %5.1lf %c\n", s->time, s->temp, s->scale);
}

static void demo(void) {
    struct tempsample * sample = malloc(sizeof sample);
    // do stuff... goto cleanup after dealing with error
cleanup:
    // clean up stuff...
    free(sample);
}

int main(void) {
    demo();
}
```

```
 20  52.3 F
```

```c
#include <stdio.h>

struct tempsample {
    int time;
    double temp;
    char scale;
};

int main(void)
{
    printf("%zu\n", sizeof(int));
    printf("%zu\n", sizeof(double));
    printf("%zu\n", sizeof(char));
    printf("%zu\n", sizeof(struct tempsample));
}
```

```c
#include <stdio.h>

struct tempsample {
    int time;
    double temp;
    char scale;
};

int main(void)
{
    printf("%zu\n", sizeof(int));
    printf("%zu\n", sizeof(double));
    printf("%zu\n", sizeof(char));
    printf("%zu\n", sizeof(struct tempsample));
}
```

On my dev PC

```
4
```

```c
#include <stdio.h>

struct tempsample {
    int time;
    double temp;
    char scale;
};

int main(void)
{
    printf("%zu\n", sizeof(int));
    printf("%zu\n", sizeof(double));
    printf("%zu\n", sizeof(char));
    printf("%zu\n", sizeof(struct tempsample));
}
```

On my dev PC

```
4
8
```

```c
#include <stdio.h>

struct tempsample {
    int time;
    double temp;
    char scale;
};

int main(void)
{
    printf("%zu\n", sizeof(int));
    printf("%zu\n", sizeof(double));
    printf("%zu\n", sizeof(char));
    printf("%zu\n", sizeof(struct tempsample));
}
```

On my dev PC

```
4
8
1
```

```c
#include <stdio.h>

struct tempsample {
    int time;
    double temp;
    char scale;
};

int main(void)
{
    printf("%zu\n", sizeof(int));
    printf("%zu\n", sizeof(double));
    printf("%zu\n", sizeof(char));
    printf("%zu\n", sizeof(struct tempsample));
}
```

On my dev PC

```
4
8
1
24
```

```
#include <stdio.h>

int main(void)
{
    int v[] = {0, 2, 4, 6, 8};
    int i = 1;
    int n = i + v[++i] + v[++i];
    printf("%d\n", n);
}
```

```c
#include <stdio.h>

int main(void)
{
    int v[] = {0, 2, 4, 6, 8};
    int i = 1;
    int n = i + v[++i] + v[++i];
    printf("%d\n", n);
}
```

[redacted]

```c
#include <stdio.h>

int main(void)
{
    int v[] = {0, 2, 4, 6, 8};
    int i = 1;
    int n = i + v[++i] + v[++i];
    printf("%d\n", n);
}
```

```
$ gcc tour.c && ./a.out
12
```

```c
#include <stdio.h>

int main(void)
{
    int v[] = {0, 2, 4, 6, 8};
    int i = 1;
    int n = i + v[++i] + v[++i];
    printf("%d\n", n);
}
```

```
$ gcc tour.c && ./a.out
12
$ icc tour.c && ./a.out
13
```

```c
#include <stdio.h>

int main(void)
{
    int v[] = {0, 2, 4, 6, 8};
    int i = 1;
    int n = i + v[++i] + v[++i];
    printf("%d\n", n);
}
```

```
$ gcc tour.c && ./a.out
12
$ icc tour.c && ./a.out
13
$ clang tour.c && ./a.out
11
```

!