```c
//
// Created by hfwei on 2022/11/10.
//

#include <stdio.h>

int main() {
  // char c = 150;
  unsigned char c = 150;
  int i = 900;

  printf("i / c = %d\n", i / c);

  return 0;
}
```

```c
/**
 * file: compare.c
 *
 * See https://randomascii.wordpress.com/2012/02/25/comparing-floating-point-numbers-2012-edition/
 */

#include <float.h>
#include <math.h>
#include <stdio.h>
#include <stdbool.h>

bool IsEqual(double x, double y);

int main() {
  printf("%d\n", IsEqual(DBL_MAX, DBL_MAX - 100));

  printf("%.50f\n", DBL_MAX - (DBL_MAX - 100));

  return 0;
}

bool IsEqual(double x, double y) {
  return fabs(x - y) <= DBL_EPSILON;
}
```

```c
//
// Created by hfwei on 2022/11/10.
//

#include <stdio.h>
#include <limits.h>
int main() {
  double pi = 3.14159;

  double pi_frac = pi - (int) pi;
  printf("The fractional part of pi is %f.\n", pi_frac);

  int num = 1000000000;
  printf("LLONG_MAX = %lld\n", LLONG_MAX);
  // long long llint = num * num;
  long long llint = (long long) num * num;
  printf("i = %lld\n", llint);

  return 0;
}
```

```c
//
// Created by hfwei on 2022/11/9.
//

#include <stdio.h>
#include <float.h>

int main() {
  // 3.402823e+38
  printf("FLT_MAX = %e\n", FLT_MAX);
  // 1.175494e-38
  printf("FLT_MIN = %e\n", FLT_MIN);
  // 1.401298e-45
  printf("FLT_TRUE_MIN = %e\n", FLT_TRUE_MIN);
  // 1.192093e-07
  printf("FLT_EPSILON = %e\n\n", FLT_EPSILON);

  // %lf for scanf; %lf for printf since C99
  // 1.797693e+308
  printf("DBL_MAX = %e\n", DBL_MAX);
  // 2.225074e-308
  printf("DBL_MIN = %e\n", DBL_MIN);
  // 4.940656e-324
  printf("DBL_TRUE_MIN = %e\n", DBL_TRUE_MIN);
  // 2.220446e-16
  printf("DBL_EPSILON = %e\n\n", DBL_EPSILON);

  return 0;
}
```

```c
//
// Created by hfwei on 2022/11/10.
//

#include <limits.h>
#include <stdio.h>

int SquareInt(int num);
double SquareDouble(double num);

int main() {
  // then i = 3; narrowing conversion (still in the range)
  int i = 3.14159;

  // narrowing conversion
  int square_int = SquareInt(3.14);
  printf("SquareInt(3.14) = %d\n", square_int);

  // narrowing conversion
  int square_double = SquareDouble(3.14);
  printf("SquareDouble(3.14) = %d\n", square_double);

  // out of the range: undefined behavior!!!
  int j = UINT_MAX;
  printf("j = UINT_MAX = %d\n", j);

  return 0;
}

int SquareInt(int num) {
  return num * num;
}

double SquareDouble(double num) {
  return num * num;
}
```

```c
// file: limits.h

#include <stdio.h>
#include <limits.h>

int main() {
  printf("INT_MIN = %d\n", INT_MIN);
  printf("INT_MAX = %d\n\n", INT_MAX);

  printf("UINT_MIN = %u\n", 0U);
  printf("UINT_MAX = %u\n\n", UINT_MAX);

  // TODO:
  short sh = SHRT_MIN;
  printf("SHRT_MIN = %hd\n", sh);
  printf("SHRT_MAX = %hd\n\n", SHRT_MAX);

  printf("LONG_MIN = %ld\n", LONG_MIN);
  printf("LONG_MAX = %ld\n\n", LONG_MAX);

  printf("ULONG_MIN = %lu\n", 0UL);
  printf("ULONG_MAX = %lu\n\n", ULONG_MAX);

  // long long int: >= 64 bits
  printf("LLONG_MIN = %lld\n", LLONG_MIN);
  printf("LLONG_MAX = %lld\n\n", LLONG_MAX);

  printf("ULONG_LONG_MIN = %llu\n", 0ULL);
  printf("ULONG_LONG_MAX = %llu\n\n", ULONG_LONG_MAX);

  return 0;
}
```

File - D:\cpl\cpl-coding-0\2022-CPL\7-data-types\int-overflow.c

```c
//
// Created by hfwei on 2022/11/10.
//

#include <stdio.h>
#include <limits.h>
int main() {
  printf("UINT_MAX = %u\n", UINT_MAX);

  unsigned int max = UINT_MAX;
  unsigned int one = 1U;
  unsigned int two = 2U;

  printf("max + one = %u\n", max + one);
  printf("one - two = %u\n", one - two);

  return 0;
}
```

```c
/**
 * file: loop.c
 *
 * Created by hengxin on 11/21/21.
 */

#include <stdio.h>

int main() {
  /**
   * Do not use a counter of type float/double,
   * although it works on some platforms.
   *
   * 0.1 cannot be exactly represented in machines.
   */
  for (double x = 0.1; x <= 1.0; x += 0.1) {
    printf("%.20f\n", x);
  }

  return 0;
}
```

```
# 7-data-types

## `int-limits.c`

## `int-overflow.c`

## `unsigned.c`

## `timing.c`

## `char.c`

## `implicit-inversion.c`

## `explicit-inversion.c`

## `float-limits.c`

## `sums.c`

## `loop.c`

## `compare.c`
```

```c
/**
 * file: sums.c
 * See https://randomascii.wordpress.com/2012/02/25/comparing-floating-point-numbers-2012-edition/
 *
 * Created by hengxin on 11/21/21.
 */

#include <stdio.h>

int main() {
  // 0.1: 0.0 0011 0011 0011
  float f = 0.1F;
  float sum = 0.0F;

  for (int i = 0; i < 10; ++i) {
    sum += f;
  }

  float product = f * 10;

  printf("sum = %.15f\nmul = %.15f\n",
         sum, product);

  return 0;
}
```

```c
//
// Created by hfwei on 2022/11/10.
//

#include <stdio.h>
#include <time.h>

long long Fib(int n);

int main() {
  int n;
  scanf("%d", &n);

  clock_t start = clock();

  printf("Fib(%d) = %lld\n", n, Fib(n));

  clock_t end = clock();
  double time = ((double) end - start) / CLOCKS_PER_SEC;
  printf("time = %f (sec)\n", time);

  return 0;
}

long long Fib(int n) {
  if (n == 0) {
    return 0;
  }

  if (n == 1) {
    return 1;
  }

  return Fib(n - 1) + Fib(n - 2);
}
```

```c
//
// Created by hfwei on 2022/11/10.
//

#include <stdio.h>

int main() {
  int array[] = {0, 1, 2, 3, 4};
  int i = -1;

  size_t size = sizeof array;
  printf("The size of the array is %zu\n", size);
  // long long size = (long long) sizeof array;

  if (i <= size) {
    printf("i <= sizeof array\n");
  } else {
    printf("i > sizeof array\n");
  }

  return 0;
}
```