

---

## NAME

**mint** — determine melodic intervals between successive pitches

## SYNOPSIS

**mint** [-acd] [-b *regex*] [-s *regex*] [*inputfile*] [> *outputfile.mnt*]

## DESCRIPTION

The **mint** command determines the distance (interval) between successive pitches. Output pitch intervals are expressed as a diatonic interval size plus interval quality; a leading plus or minus sign indicates whether the interval is ascending or descending. By way of illustration, **mint** will change a sequence of *\*\*pitch* data tokens — such as *C4*, *A4*, *E4* — to the interval sequence *+M6*, *-P4*. Each pitch-related input spine is transformed to a corresponding *\*\*mint* output spine.

The **mint** command determines melodic intervals only for pitch tokens *within* individual spines. Pitch intervals *across* spines are not determined by **mint**.

No interval is calculated for the first pitch token; initial pitches are simply echoed in the output — appearing in square brackets. These initial pitches are referred to as *offset* values, since they indicate the starting value from which subsequent intervals are calculated. Offset values can prove useful in attempting to reconstruct the input, but the user may wish to eliminate offset values in subsequent processing (see below).

The **mint** command is able to handle multiple-stops. Data-flow interruptions such as the occurrence of barlines can be handled using the **-s** option. By defining *regular expression* patterns, the user may select which types of data tokens should be ignored by **mint**. (See EXAMPLES below.)

Note that the output spine generated by **mint** preserves the same record-type structure as the input, and so may readily be pasted with the input file using the Humdrum **assemble** command.

The **mint** command is able to accept any of the pitch-related representations listed below. For descriptions of the various input representations refer to Section 2 (*Representation Reference*) of this reference manual.

It is recommended that output files produced using the **mint** command should be given names with the distinguishing *.mnt* extension.

<b>**kern</b>	core pitch/duration representation
<b>**pitch</b>	American National Standards Institute pitch notation (e.g. “A#4”)
<b>**solfg</b>	French solfège system (fixed ‘doh’)
<b>**Tonh</b>	German pitch system

*Input representations processed by **mint**.*

## OPTIONS

The **mint** command provides the following options:

- a** output absolute pitch interval without distinguishing ascending/descending
- b *regexp*** break; do not calculate difference for tokens matching *regexp*;  
restart interval calculations with next pitch token
- c** output compound intervals as non-compound intervals
- d** output diatonic interval size only, without the interval quality
- h** displays a help screen summarizing the command syntax
- s *regexp*** skip; completely ignore tokens matching *regexp*;  
(echo in output only)

Options are specified in the command line.

By default, **mint** distinguishes ascending and descending intervals by prepending a plus or minus sign. If the **-a** option is invoked, then only unsigned (absolute) intervals are output.

The “skip” function takes precedence over the “break” function, so input strings matching both the skip (-s) and break (-b) regular expressions cause a skip rather than a break.

## EXAMPLES

The various aspects of the **mint** command are best illustrated using a set of examples. Consider the following input:

```

**kern
=1
8c
8g
4.b-
.
8e
=2
4f
8r
8C
4FF
=3
*-

```

Using the default invocation, the **mint** command transforms the above input as follows:

```

**mint
=1
[c]
+P5
+m3
.
-d5
=2
+m2
r
-P11
-P5
=3
*-

```

The leading or “offset” pitch ‘c’ has been echoed in square brackets in the third record. This represents the initial pitch from which subsequent pitch distances are measured. This offset value reflects the type of input given to **mint**, thus if the input format is **\*\*kern** the offset pitch will be recorded in the same representation. Note that for absolute pitch units: ‘c’ (**\*\*kern**) equals ‘C4’ (**\*\*pitch**) equals ‘do4’ (**\*\*solfg**) equals ‘C4’ (**\*\*Tonh**).

The subsequent output value (+P5) indicates a melodic interval of an ascending perfect fifth. This is followed by an ascending minor third (+m3) followed by a descending diminished fifth (-d5).

Notice that the null-token in the sixth record has been echoed. Null-tokens have no effect on interval calculations and are treated as though they are non-existent. In addition, **mint** correctly echoes (and ignores) both rests and kern-like barlines. Note that pitch intervals spanning a rest continues to be calculated and that intervals greater than an octave remain as “compound intervals.”

Depending on the application, users may wish to suppress the calculation of intervals across rests. In this case, the **mint** command can be invoked using the “break” (-b) option. Consider the command: `mint -b 'r' input > output.mnt`. Each time **mint** encounters a data token that matches the letter ‘r’ it echoes the input token and begins looking for a new offset value. Applied to the earlier input file, this command produces the following output:



```

**mint
=1
[c]
+P5
+m3
.
-d5
=2
+m2
8r
[C]
-P5
=3
*-

```

Notice that the descending perfect eleventh spanning the rest has been eliminated, and a new pitch offset value 'C' has been echoed in the corresponding output.

The **-s** (skip) option can be used to allow the user to selectively identify records that should not be involved in processing. For example, the command

```
mint -s '^[^4]+$' input > output.mnt
```

will cause any data token not matching the number 4 to be skipped during processing. Given the sample input, intervals will be calculated only between quarter-notes and dotted quarter-notes:

```

**mint
=1
8c
8g
[b-]
.
8e
=2
-P4
8r
8C
-P15
=3
*-

```

Using the skip option, the user may calculate melodic intervals between pitches in strong metric positions, or pitches that have been marked as structural tones.

The **mint** command is also able to process numerical data tokens containing multiple-stops. Consider the following following sample input:

```

**pitch
C4
B3 D4
A3 C4 F4
C4
C4
*—

```

Notice the presence of the double- and triple-stops in the third and fourth records. Using the default invocation, the **mint** command processes this input as follows:

```

**mint
[C4]
-m2 +M2
-M2 (+m2) (-M2) +m3
+m3 P1 -P4
P1
*—

```

The leading or offset value [C4] has been echoed in the second record. (The user might wish to eliminate such offset values via the **humshed** command; see below.) The third record in both the input and output contain double-stops. In the output, the first value of the double-stop (-m2) represents the pitch interval between C4 and B3. The second value in the double-stop (M2) represents the difference between C4 and D4. In short, **mint** traces both possible difference “paths.”

In processing successive multiple-stops **mint** does not calculate all of the possible permutations. For example, in the case of two consecutive triple-stops, **mint** will calculate three intervals corresponding to the first notes in both triple-stops, the second notes, and the third notes.

Where the number of multiple-stops changes, **mint** operates under some special conventions. Consider, for example, the case of a double-stop followed by a triple-stop: the pitches P+Q followed by X+Y+Z. All of the possible (interval) differences might be enumerated as follows: X-P, Y-P, Z-P, X-Q, Y-Q and Z-Q. **Mint** first calculates the “outer” interval distances (X-P and Z-Q). It then calculates a permuted set of “inner” intervals (Y-P and Y-Q). The remaining intervals are considered unlikely or implausible and are not calculated by **mint**.

In the above example, moving from the double-stop to the triple stop between records three and four generates two “outer” interval distances (B3 to A3 → -M2; D4 to F4 → +m3), as well as the permuted “inner” intervals (B3 to C4 → +m2; D4 to C4 → -M2). Both the resulting inner intervals are printed in parentheses. A similar process occurs when moving from records four to five. Three intervals may be traced from the 3 initial pitches to the subsequent single pitch.

Depending on the goal, the presence of the parentheses makes it easy for the user to eliminate the inner intervals using the Humdrum stream-editor **humshed**. For example, the command

```
humsed 's/([^\)]*) //g' input > output
```

can be used to eliminate inner intervals. Alternatively, the command

```
humsed 's/[()]/g' input > output
```

can be used to eliminate the parentheses surrounding the inner intervals. Offset values can be transformed to null-tokens using the command

```
humsed 's/\[[^\]]*\]/./g' input > output
```

records containing offset values can be eliminated using the command

```
humsed '/\[.*\]/d' input > output
```

One final example illustrates how several spines can be processed concurrently by **mint**.

```
!! J.S. Bach, keyboard Sinfonia No. 13
**Tonh          **pitch          **kern
*M3/8           *M3/8           *M3/8
*a:             *a:             *a:
=7              =7              =7
A3              G4              16ee
.               .               16ff#
H3              F#4             8.dd#
H2              .               .
.               .               16ee
=8              =8              =8
E3              E4              4ee
.               F#4             .
.               G4              .
.               A4              .
r               B4              [8gg
.               C#5             .
=9              =9              =9
r               [D5             16gg]
.               .               16bb-
.               .               16aa
.               .               16gg
.               .               16ff
.               .               16ee
=10             =10             =10
r               D5              [4.ff
*-              *-              *-
```

The following command invokes the **-a** and **-d** options. Indications of the direction of interval movement (ascending/descending) have been removed, and the diatonic interval sizes are output without the associated interval qualities (major/minor/perfect/diminished, augmented).

```
mint -a -d input | humsed 's/\[[^\t]*\]/./g' > output
```

Notice in the corresponding output that the initial offset pitches have been changed to a null tokens (due to the **humsed** command).

```
!! J.S. Bach, keyboard Sinfonia No. 13
**mint          **mint          **mint
*M3/8           *M3/8           *M3/8
*a:            *a:            *a:
=7             =7             =7
.              .              .
.              .              2
2              2              3
8              .              .
.              .              2
=8            =8            =8
4              2              1
.              2              .
.              2              .
.              2              .
r              2              3
.              2              .
=9            =9            =9
r              2              1
.              .              3
.              .              2
.              .              2
.              .              2
.              .              2
.              .              2
=10           =10           =10
r              1              2
*_            *_            *_
```

## PORTABILITY

DOS 2.0 and up, with the MKS Toolkit. OS/2 with the MKS Toolkit. UNIX systems supporting the *Korn* shell or *Bourne* shell command interpreters, and revised *awk* (1985).

## SEE ALSO

**\*\*hint** (2), **hint** (4), **humsed** (4), **kern** (2), **\*\*mint** (2), **recode** (4), **regexp** (6), **\*\*semit**s (2), **semit**s (4), **solfg** (2), **\*\*Tonh** (2), **xdelta** (4), **ydelta** (4)