

hako.sty ver. 1.13

arhako.sty ver.0.88

使用例

tDB

2004/12/25

概要

センター試験の問題で使われている解答を記入すべきハコを作るためのパッケージです。
このマクロ集のマクロについてのご質問, バグ報告, 修正・追加の提案等は

<http://emath.s40.xrea.com/>

の掲示板へどうぞ。

はじめに

センター試験では $1024 = 2^{\boxed{\text{アイ}}}$ のように、解答を書くべき所に必要な桁数分の文字を含んだハコを用意する、という方法が採られています。このハコを $\text{T}_{\text{E}}\text{X}$ で記述するのに私は `\hako{アイ}` などと引数の中にハコに入れる文字を記述していました。Waver さんがこれを見かねて、ハコの中の文字を自動的にインクリメントし、それを相互参照するマクロを作ってくださいました。そのマクロをもとにいろいろな機能を付け加えて、現在の `hako.sty` ができ上がってきました。Waver さんのお陰で、大変楽ができるようになりました。

有り難うございます。 > Waver さん。

目次

1	基本的用法	1
2	相互参照	3
3	ハコの中に複数の文字	4
4	複数の文字ハコへの相互参照	4
5	ハコの文字種 (1)	6
6	ハコの文字種 (2)	7
7	ハコの文字の初期値変更	8
8	ハコの文字の書体変更	10
9	ハコの枠線太さの変更	11
10	ハコの横巾変更 (1)	12
11	ハコの横巾変更 (2)	13
12	ハコの文字位置	14
13	ハコの高さ変更	15
14	中が空白のハコ (ハコの文字を直接指定)	16
15	答を埋め込む (1)	17
16	答を埋め込む (2)	17
17	埋め込んだ答えの表示形式	18
18	ハコの中を網掛け	21
19	ハコいろいろ	23
19.1	ハコの文字の間に縦仕切線	23
19.2	破線の仕切	24
19.3	先頭文字を で囲む	24
20	¥HAKO — 高速版 ¥Hako	25
21	コマンドリファレンス (1)	27
22	コマンドリファレンス (2)	29

23	附 記号を付与しないハコ	29
23.1	¥karaHako	30
23.2	¥maskHako	30

1 基本的用法

まずは、基本的な使い方です。このスタイルファイルは初期化が必要です。`\begin{document}`の直後に

```
\hakosyokika
```

という一行を書きます。これで内部変数などが初期化され、`hako.sty` が使用可能になります。あとは、ハコを作りたい所で、`\Hako` と書くだけです。例えば、

基本的用法

```
\documentclass{jarticle}
\usepackage{emath,hako}

\begin{document}
\hakosyokika
二次不等式 $(x-1)(x-2)<0$ の解は $\Hako{<x<\Hako}$ である。
\end{document}
```

をタイプセットすると

二次不等式 $(x-1)(x-2) < 0$ の解は ア $< x <$ イ である。

となります。

すなわち、コマンド `\Hako` が登場するたびにハコが作られ、その中の文字が

ア、イ、ウ、エ、オ、カ、キ、……

と五十音順にインクリメントされていきます。

注 1 上の例で、コマンド `\Hako` と不等号 `<` の間に `{}` がありますが、この意味について説明します。

コマンド `\Hako` はオプション引数を取りますが、そのオプション文字として

`[, <, (, /, ", '`

が使われています。したがって、`\Hako<x` と書きますと、不等号 `<` が、コマンド `\Hako` のオプション引数の記号と解釈されてしまいます。それを防ぐために `\Hako` と不等号 `<` の間に `{}` を入れています。さぼって、`\Hako` と不等号の間に半角のスペースを入れてもたいていは大丈夫なようにしてありますが、特殊な場面で不都合が生ずることがありますので、`{}` をお勧めします。

注 2 `\hakosyokika` は必ず実行してください。これを実行していない場合は、エラー

！ コマンド：`hakosyokika` が実行されていません。

が発生します。

注 3 古いバージョン (ver.0.63 より前) では、

`\hakosyokika` ではなく、`\hakobansyokika`

でした。ver.0.63以降では、\hakobansyokika では、ハコのカウンタのみを初期化することとし、\hakosyokika ですべての内部変数を初期化すると、使い分けることになりました。ただし、\hakobansyokika が文書の最初に呼ばれたときは \hakosyokika と同じ働きをするようにしてありますから、今まで作成された文書を書き換える必要は少ないでしょう。

注 4 以下、具体例を挙げている部分は、

角の丸い囲みの中に

ソースリスト

影つきの囲みの中に

それをタイプセットした結果

を記述します。

2 相互参照

二次方程式 $x^2 - 3x + 2 = 0$ の解は $x =$, である。
という問題では、解が一意に定まりません。そこで、

ただし $<$ とする。

というただし書きをつけたいです。これを実現するために、ハコにラベルを付け、それを参照する機能があります。

まず、ラベルを付けるには `\Hako` にオプション引数でラベル名を指定します：

```
\Hako[ラベル名]
```

次にそれを参照するには、コマンド

```
\refHako{ラベル名}
```

または

```
\refHako*{ラベル名}
```

を用います。両者の違いは、前者がハコの中の文字だけを、後者は枠を付けて参照します。先の例を完成させます。

— [...] オプションと `\refHako*` —

```
二次方程式  $x^2 - 3x + 2 = 0$  の解は  $x =$  \Hako[L02a],  
\ \Hako[L02i] $ である。ただし,  
$ \refHako*{L02a} < \refHako*{L02i} $ とする。
```

をタイプセットすると

```
二次方程式  $x^2 - 3x + 2 = 0$  の解は  $x =$  ,  である。ただし,   $<$   と  
する。
```

となります。

3 ハコの中に複数の文字

センター試験では、2桁以上の答を要求するハコでは、ハコの中に桁数分の文字を用意する方法が採られています。例えば、

2^{50} は

アイ

 桁の整数である。ただし、 $\log_{10} 2 = 0.3010$ とする。

これを実現するために、`\Hako` にオプション引数でハコの中に入れる文字数を指定する機能があります。今度のオプションはオプション文字が前節のものとは異なります。

`\Hako<文字数>`

次のように使います。

— <..>オプション —

2^{50} は `\Hako<2>` 桁の整数である。
ただし、 $\log_{10} 2 = 0.3010$ とする。

をタイプセットすると

2^{50} は

アイ

 桁の整数である。ただし、 $\log_{10} 2 = 0.3010$ とする。

となります。

4 複数の文字ハコへの相互参照

複数の文字をもつハコに付けたラベルによる参照は、ハコの中の文字全体への参照となります。しかし、ときにはハコの個別の文字への参照をすることもあります。その機能は第2の [..] オプションで実現されます。すなわち、

`\Hako<文字数>[全体ラベル][個別ラベル]`

という書式です。個別ラベルは、文字数分のラベルをコンマで区切って記述します。例えば

`\Hako<3>[Lb1][L1,L2,L3]`

と記述すると

<code>\refHako{Lb1}</code>	アイウ
<code>\refHako{L1}</code>	ア
<code>\refHako{L2}</code>	イ
<code>\refHako{L3}</code>	ウ

と、参照されます。

具体例を挙げます。これは1997年度の大学入試センター試験の問題です。


```

\begin{caprm}
\sankaku{ABC}の外心を O，直線 BO と外接円の交点を D とする。
また、垂心を H，直線 AH と直線 BC の交点を E とする。

次の文中の \refHako*{L04ai} ~ \refHako*{L04oka} に
当てはまるものを，記号 A ~ E のうちから選べ。
(\refHako*{L04ai} の \refHako{L04a} と \refHako{L04i}，
\refHako*{L04ue} の \refHako{L04u} と \refHako{L04e} について
は、解答の順序を問わない。)
\[ AH \heikou \Hako<2>[L04ai] [L04a,L04i]，
\quad CH \heikou \Hako<2>[L04ue] [L04u,L04e] \]
であるから、四角形 AH\Hako<2>[L04oka] は平行四辺形である。
\end{caprm}

```

をタイプセットすると

△ABC の外心を O，直線 BO と外接円の交点を D とする。また、垂心を H，直線 AH と直線 BC の交点を E とする。

次の文中の アイ ~ オカ に当てはまるものを，記号 A ~ E のうちから選べ。(アイ のアとイ， ウエ のウとエについては、解答の順序を問わない。)

AH // アイ， CH // ウエ

であるから、四角形 AHオカ は平行四辺形である。

となります。

注 個別ラベルをつけるときは，必ず全体ラベルをつけなければなりません。

5 ハコの文字種 (1)

ハコの中の文字は、デフォルトでは

ア, イ, ウ, エ, オ, カ, キ, …

と、カタカナが五十音順に現れます。これを

a, b, c, d, e, …

などと英字をアルファベット順に並べたいといった、ハコの中の文字種を変更するためにコマンド `\hakomozisyu{.}` が用意してあります。このコマンドの引数として使える文字は次の通りです。

ア	: ア, イ, ウ, エ, オ, …	あ	: あ, い, う, え, お, …
イ	: イ, 口, 八, 二, ホ, …	い	: い, ろ, は, に, ほ, …
1	: 1, 2, 3, 4, 5, …	m	: ①, ②, ③, ④, ⑤, …
i	: i, ii, iii, iv, v, …	I	: I, II, III, IV, V, …
a	: a, b, c, d, e, …	A	: A, B, C, D, E, …

注 その他の文字はそのまま出力されますから、引数に '(1)' と指定すれば、

(1), (2), (3), (4), (5), …

と、ナンバリングされます。一例を挙げます。

— `¥hakomozisyu` —

```
\hakosyokika\hakomozisyu{(イ)}%  
2点$(2,0)$, $(0,3)$を通る直線は$\Hako x+\Hako y=6$である。
```

をタイプセットすると

```
2点 (2,0), (0,3) を通る直線は (イ) x + (口) y = 6 である。
```

となります。

注 アイウ… は、49文字しかありません。これを越えて `\Hako` を使用すると

```
! Undefined control sequence.  
\GenericError ...  
#4 \errhelp \@err@ ...
```

あるいは

```
! LaTeX Error: Counter too large.
```

等のエラーが発生します。

6 ハコの文字種 (2)

一文書中ですべてのハコの文字種が一種類のときは、前節の方法で処理できますが、一つの問題で2種類以上の文字種を用いる場合の対策です。

このときは、メインの方の文字種を前節の方法で指定します。他の系列の文字種が登場するハコに対しては、\Hako のオプションで処理します。

\Hako/文字種/

/../ の引数は前節の \hakomozisyu の引数と同じです。少し長い例です。

```

\hakosyokika
\begin{caprm}
次の文中の空欄 \refHako*{L06a}～\refHako*{L06ku} にあてはまる
整数を 0～9 から選んで該当する解答欄 (省略) にマークし,
空欄\refHako*{L06A}と\refHako*{L06I}には, あてはまる $a$ の
式を該当する解答欄 (省略) にマークせよ. \par
正三角形 ABC と同一平面上の点 P が不等式 $AP^2 > BP^2 + CP^2$ を満た
すとき, 辺 BC の中点を原点, 直線 BC を $x$ 軸, 辺 BC の垂直二等分線
を $y$ 軸とする座標系を設定して点 P の存在する範囲を求めよう.
正三角形の 1 辺の長さを $2a$, 頂点 B の座標を $(-a, 0)$,
頂点 A の $y$ 座標を正とすると, 頂点 A, C の座標は
A$(0, \Hako[L06A]/あ)$, C$(\Hako[L06I]/あ/, 0)$となる.
点 P$(x, y)$ が上記の不等式を満たすとき, $x, y$ は不等式
\[ \Hako[L06a]x^2 + \Hako y^2 + \Hako x
+ \Hako \sqrt{\Hako} ay - a^2 < 0 \]
を満たすことになる.
したがって不等式 $AP^2 > BP^2 + CP^2$ を満たす点 P は $\Hako a$ を半
径とし, 点 $(\Hako, -\sqrt{\Hako[L06ku]}a)$ を中心とする円の内部の点である.
\end{caprm}

```

をタイプセットすると

次の文中の空欄 ～ にあてはまる整数を 0～9 から選んで該当する解答欄 (省略) にマークし, 空欄 と には, あてはまる a の式を該当する解答欄 (省略) にマークせよ.

正三角形 ABC と同一平面上の点 P が不等式 $AP^2 > BP^2 + CP^2$ を満たすとき, 辺 BC の中点を原点, 直線 BC を x 軸, 辺 BC の垂直二等分線を y 軸とする座標系を設定して点 P の存在する範囲を求めよう. 正三角形の 1 辺の長さを $2a$, 頂点 B の座標を $(-a, 0)$, 頂点 A の y 座標を正とすると, 頂点 A, C の座標は $A(0, \text{あ})$, $C(\text{い}, 0)$ となる. 点 $P(x, y)$ が上記の不等式を満たすとき, x, y は不等式

$$\text{ア} x^2 + \text{イ} y^2 + \text{ウ} x + \text{エ} \sqrt{\text{オ}} ay - a^2 < 0$$

を満たすことになる. したがって不等式 $AP^2 > BP^2 + CP^2$ を満たす点 P は a を半径とし, 点 $(\text{キ}, -\sqrt{\text{ク}}a)$ を中心とする円の内部の点である.

となります.

7 ハコの文字の初期値変更

デフォルトでは, ハコの中の文字は五十音順に先頭の ‘ア’ から出現します. これを途中の ‘ケ’ から始めて

ケ, コ, サ, シ, ス, セ, ソ, タ,.....

としたいときの話です。ハコの中の文字をインクリメントするのに L^AT_EX のカウンタを用いています。デフォルトのカウンタ名は hakobanaiu です。この値を変更することでハコの中の文字を自由に変更することができます。先の例では、‘ケ’ は五十音の 9 番目の文字ですから、

```
\setcounter{hakobanaiu}{8}%
```

と、1 つ小さい値を指定します (\Hako が呼び出される時に、カウンタは 1 つ増やされてからそれを文字に翻訳します。カウンタについての詳しいことは、L^AT_EX についての参考文献をご覧ください。)

ハコのカウンタは前節の文字種ごとに別のものが用意されています。

文字種 : カウンタ名

ア : hakobanaiu

イ : hakobaniru

あ : hakobanAIU

い : hakobanIRO

l : hakobanara

i : hakobanrma

I : hakobanRMA

a : hakobanalp

A : hakobanALP

m : hakobanmar

一例を挙げます。

ハコ文字のカウンタ

```
\hakosyokika
\hakomozisyu{m}%
\setcounter{hakobanmar}{9}%
不等式
\[ \log_{\frac{1}{2}}(2-x) \leq 2\log_{\frac{1}{2}}x \]
を解くと、
\[ \Hako{x} < x \leq \Hako \text{ となる。} \]
```

をタイプセットすると

不等式

$$\log_{\frac{1}{2}}(2-x) \leq 2\log_{\frac{1}{2}}x$$

を解くと、

$$\textcircled{10} < x \leq \textcircled{11} \text{ となる。}$$

となります。

8 ハコの文字の書体変更

ハコの中の文字は、デフォルトではゴシックという書体を用いています。これを別の書体に変更するには、コマンド `\hakosyotai{..}` を用います。例えば、明朝体にしてみましょう。

— ¥hakosyotai —

```
\hakosyokika
\hakomozisyu{イ}
\hakosyotai{\textmc}
\sankaku{ABC}において、辺 BC, CA, AB の長さを
それぞれ  $a, b, c$  とし  $\text{kaku}{A}$  の大きさを  $A$  で表す。
C から AB に垂線 CH を引く。  $\text{sankaku}{ACH}$  において、
 $\text{CH}=\text{Hako}$ ,  $\text{AH}=\text{Hako}$ 
したがって、  $\text{BH}=\text{Hako}$ 
そこで、  $\text{sankaku}{BCH}$  に三平方の定理を適用することにより、
余弦定理  $a^2=\text{Hako}$  が導かれる。
```

をタイプセットすると

$\triangle ABC$ において、辺 BC, CA, AB の長さをそれぞれ a, b, c とし $\angle A$ の大きさを A で表す。
C から AB に垂線 CH を引く。 $\triangle ACH$ において、 $CH = \text{イ}$, $AH = \text{ロ}$ したがって、
 $BH = \text{ハ}$ ところで、 $\triangle BCH$ に三平方の定理を適用することにより、余弦定理 $a^2 = \text{ニ}$
が導かれる。

となります。

9 ハコの枠線太さの変更

ハコの枠線の太さは、デフォルトでは 1pt としてあります。これを変更するにはコマンド `\hakosenhaba` を用います。L^AT_EX の `\framebox` のデフォルト値は 0.4pt です。

```
\hakosenhaba
\hakosenhaba{0.4pt}%
$x$ の整式
\[ x^3+4ax^2+(4-b)x+c \]
を $x^2+2ax+2a$ で割ったときの余りは
\[ \left(\Hako-b-\Hako a-\Hako a^2\right)x+c-\Hako a^2\]
である。この余りが $-2x+7$ になるような整数 $a, b, c$ のうち、
$b$ が正となるものは
\[ a=\Hako, \quad b=\Hako, \quad c=\Hako \]
および
\[ a=\Hako<2>, \quad b=\Hako, \quad c=\Hako<2>\text{ { である。}}\]
```

をタイプセットすると

x の整式

$$x^3 + 4ax^2 + (4 - b)x + c$$

を $x^2 + 2ax + 2a$ で割ったときの余りは

$$\left(\boxed{\text{ア}} - b - \boxed{\text{イ}} a - \boxed{\text{ウ}} a^2 \right) x + c - \boxed{\text{エ}} a^2$$

である。この余りが $-2x + 7$ になるような整数 a, b, c のうち、 b が正となるものは

$$a = \boxed{\text{オ}}, \quad b = \boxed{\text{カ}}, \quad c = \boxed{\text{キ}}$$

および

$$a = \boxed{\text{クケ}}, \quad b = \boxed{\text{コ}}, \quad c = \boxed{\text{サシ}} \text{ である。}$$

となります。

10 ハコの横巾変更 (1)

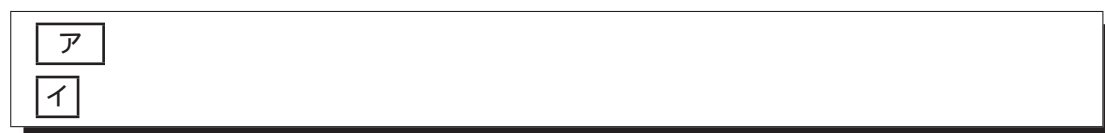
ハコの中の文字と枠線との間はデフォルトで 3pt の余白が取られる他に，左右あわせて約 1zw (全角 1 文字分)，正しくは可変長で

```
1zw plus .5zw minus .05zw
```

の余白を付けています．これを変更するには，コマンド `\hakoxyohaku{.}` を用います．デフォルトの状態と，この余白を取り去った状態を比較してみます．

```
\hakoxyohaku —————  
  
\Hako \par  
\hakoxyohaku{0pt}\Hako
```

をタイプセットすると



となります．

注 ハコの横巾の指定には，グルーが含まれています．グルーについて詳しいことは，参考文献—
例えば「LaTeX 自由自在」等を参照してください．

11 ハコの横巾変更 (2)

文書中のすべてのハコの横巾を変更するには、前節の方法を採りますが、一部のハコの横巾を変更したいときは (...) オプションを用います。

`\Hako` (横余白サイズ)

分数で、分母と分子の桁数が違うときは、ハコの横巾が違ってしまいます。このようなとき、このオプションを用いて分母子のハコの横巾を揃えることができます。

— (...) オプション —

`$$\bunsuu{\Hako<2>}{\Hako}$$` では分母子のハコの横巾が不揃いです。

分母のハコに余白を調整するオプションをつけると、
`$$\bunsuu{\Hako<2>}{\Hako(2zw)}$$` と揃えることができます。

をタイプセットすると

$$\frac{\boxed{\text{アイ}}}{\boxed{\text{ウ}}}$$
 では分母子のハコの横巾が不揃いです。

分母のハコに余白を調整するオプションをつけると、
$$\frac{\boxed{\text{エオ}}}{\boxed{\text{カ}}}$$
 と揃えることができます。

となります。

12 ハコの文字位置

ハコの中の文字位置はデフォルトでは「中央」となっています。これを変更するコマンドは `\hakomoziiti{..}` です。引数は次のものが有効です。

- l : 左寄せ
- c : 中央 (デフォルト)
- % r : 右寄せ
- ^ : 左寄せ, かつ上付き
- L : 左外, かつ上付き

一例を示します。

```
\hakomoziiti
\hakosyokika
\hakomozisyu{1}%
\hakomoziiti{L}%
\hakosyotai{\sffamily}%
\setcounter{hakobanara}{10}%
\hakoxyohaku{0pt}%
\hakosenhaba{.4pt}%
```

二点 $(-1, -1)$, $(1, 5)$ を通る直線の方程式は $y = \text{\Hako } x + \text{\Hako}$ である。

をタイプセットすると

二点 $(-1, -1)$, $(1, 5)$ を通る直線の方程式は $y = \text{\Hako} x + \text{\Hako}$ である。

となります。

なお、特定のハコだけ文字位置を指定したいときは、第 2 の (..) オプションを uses。このときは第 1 の (..) オプションは必ずつけなければなりません。例えば

第 2 の (..) オプション

```
\hakosyokika
\Hako
\Hako(2zw)(1)
\Hako
```

をタイプセットすると

ア イ ウ

となります。

13 ハコの高さ変更

ハコの高さはデフォルトでは

```
\vphantom{あ}
```

すなわち全角文字の高さ，となっています．さらに，その上下に各 3pt の余白がつきます．これを
変更するには，`\hakosityuu` コマンドを使用します．例えば

—¥hakosityuu—

```
\hakosyokika  
$\bunsuu{x}{y}=\Hako$
```

のようなとき，左辺と右辺の高さを揃えたければ，

```
\hakosityuu{\vphantom{$\bunsuu{x}{y}$}}%  
$\bunsuu{x}{y}=\Hako$
```

をタイプセットすると

$$\frac{x}{y} = \boxed{\text{ア}}$$

のようなとき，左辺と右辺の高さを揃えたければ，

$$\frac{x}{y} = \boxed{\text{イ}}$$

となります．

直接高さを指定するには，

—¥hakotakasa—

```
\hakosyokika  
\hakotakasa[4mm]{10mm}% 高さ 10mm，深さ 4mm と指定する．  
$\bunsuu{x}{y}=\Hako$
```

をタイプセットすると

$$\frac{x}{y} = \boxed{\text{ア}}$$

となります．

14 中が空白のハコ（ハコの文字を直接指定）

次の を補え .

などと、中が空白のハコを作りたいときもあります . 上のハコは

```
\Hako(2zw minus Opt)""
```

として "" オプションを用いています . ハコの横巾を調整するには、(..) オプションを使用します .

"..." オプションは Ver.0.64 から追加されたオプションで、ハコの文字を直接指定したいときに用います . 例えば

—"..."オプション—

```
\hakosyokika
```

```
\Hako
```

```
\Hako"例題"
```

```
\Hako
```

```
\verb/"..."/オプションを用いた\verb/\Hako/では、  
カウンタは進みません .
```

をタイプセットすると

ア	例題	イ
---	----	---

"..."オプションを用いた\Hako では、カウンタは進みません .

となります .

中が空白のハコを作る別法として、`*karaHako` コマンドを用いることもできます（後述）

15 答を埋め込む(1)

箱の答を本文中に書いておき、問題の後にまとめてタイプセットすることができます。これは '...' オプションで、'...' の中に答を記入しておきます。ただし、答は数式モードになります。

まずは、一例をごらん頂きましょう。

--- '...' オプション ---

```
\hakosyokika

$$0 \leq x < 2\pi$$

のとき  $2\sqrt{3}\cos^2 x + \sin 2x - 2\sqrt{3} < 0$ 
\vspace{.8ex}をみたす  $x$  の範囲は
 $\Hako'\bunsuu{\pi}{6}' < x < \Hako[L-o]'\pi'$  と
 $\Hako[L-ka]'\bunsuu{7\pi}{6}' < x$ 
 $< \Hako'2\pi'$  である。 \vspace{.8ex}

(ただし,
 $\refHako*[L-o] < \refHako*[L-ka]$ 
とする。) \bigskip

\textbf{答}

\inputHakoKaiFile
```

をタイプセットすると

$0 \leq x < 2\pi$ のとき $2\sqrt{3}\cos^2 x + \sin 2x - 2\sqrt{3} < 0$ をみたす x の範囲は $< x <$
と $< x <$ である。
(ただし, $<$ とする.)

答
 $\text{ア} = \frac{\pi}{6}$ $\text{イ} = \pi$ $\text{ウ} = \frac{7\pi}{6}$ $\text{エ} = 2\pi$

となります。

すなわち、問題の中で、コマンド `¥Hako` のオプション引数 '...' の中に記述された答が問題の後にタイプセットされます。

注1. 答は、別ファイル(同一ファイル名で、拡張子が .hka)に書き出されます。それを

`¥inputHakoKaiFile`

というコマンドで読み込みます。

16 答を埋め込む(2)

センター試験では、答の桁数分だけ八コの中に文字が現れます。これを

`¥Hako<2>'43'`

等のように答と桁数の両方を記述するのは無駄ですね．そこで，‘...’ オプションをつけてその内容から桁数がカウントできる場合は<...>オプションを省略することができるようにしました．そのおまじないが

¥centermodetrue

です．その例をごらん頂きましょう．

¥centermodetrue

```
\hakosyokika
\centermodetrue
$y=x^2-2kx+8k+9$のグラフが$x$軸と共有点をもたないとき，
定数$k$のとりうる値の範囲は
$\Hako'-1' < k < \Hako'9'$ である．
```

```
\textbf{答}
```

```
\inputHakoKaiFile
```

をタイプセットすると

$y = x^2 - 2kx + 8k + 9$ のグラフが x 軸と共有点をもたないとき，定数 k のとりうる値の範囲は $< k <$ である．
 答
 アイ = -1 ウ = 9

となります．

すなわち ¥hakosyokika に続けて，¥centermodetrue としますと，ハコに入れるべき文字数は，‘...’ オプションで与えられる答の文字数となります．この場合，<...>オプションをつけても無視されます．

なお，デフォルトは ¥centermodetrue の状態です．前節の例は，この状態で実行されています．

17 埋め込んだ答えの表示形式

埋め込んだ答えを表示する形式は

ア = 1, イウ = 23

という行形式のほか，

ア	イウ
1	23

という表形式を新設したことにあります．

この形式を利用するには，

```
\HakoKaiKata{t}
```

と宣言します。

次に一例です。

まず、今までの形式から

従来の形式

```
\hacosyokika
\centermodetrue
\openHakoKaiFile
\[ y=\Hako'2'x+\Hako'34' \]
\closeHakoKaiFile
\begin{Kaitou}
\inputHakoKaiFile
\end{Kaitou}
```

$$y = \boxed{\text{ア}}x + \boxed{\text{イウ}}$$

【解答】

$$\text{ア} = 2 \quad \text{イウ} = 34$$

解答を表形式にするには`\HakoKaiKata{t}`を宣言します。下の例では、`\inputHakoKaiFile`の直前に宣言していますが、文書全体でこの形式をとるときは、文書の冒頭で

```
\hacosyokika
\HakoKaiKata{t}
```

などとすればよいでしょう。

表形式

```
\hacosyokika
\centermodetrue
\openHakoKaiFile
\[ y=\Hako'2'x+\Hako'34' \]
\closeHakoKaiFile
\begin{Kaitou}
\HakoKaiKata{t}
\inputHakoKaiFile
\end{Kaitou}
```

$$y = \boxed{\text{ア}}x + \boxed{\text{イウ}}$$

【解答】

ア	イウ
2	34

答えに分数などが含まれ、高さが異なるときはその調整が必要です。

—— 高さが異なる場合 ——

```
\hakosyokika
\openHakoKaiFile
\[ a=\Hako'\bunsuu12',~b=\Hako'\sqrt3'
\]
\closeHakoKaiFile
\begin{Kaitou}
\HakoKaiKata{t}
\inputHakoKaiFile
\end{Kaitou}
```

$$a = \boxed{\text{ア}}, b = \boxed{\text{イ}}$$

【解答】

ア	イ
$\frac{1}{2}$	$\sqrt{3}$

対策は支柱を立てることです。書式は

`\HakoKaiSityuu[深さ]{高さ}`

です。上の例で支柱を立ててみます。

—— 高さが異なる場合の調整 ——

```
\hakosyokika
\openHakoKaiFile
\[ a=\Hako'\bunsuu12',~b=\Hako'\sqrt3'
\]
\closeHakoKaiFile
\begin{Kaitou}
\HakoKaiKata{t}
\HakoKaiSityuu[1zh]{1.5zh}
\inputHakoKaiFile
\end{Kaitou}
```

$$a = \boxed{\text{ア}}, b = \boxed{\text{イ}}$$

【解答】

ア	イ
$\frac{1}{2}$	$\sqrt{3}$

¥Hako の答え文字列を修飾するには、¥hakovaisyotai を用います。

—— 答えを赤で ——

```
\openHakoKaiFile
$(x+a)^2=x^2+\Hako'2a'x+\Hako'a^2'$
\closeHakoKaiFile

\hakovaisyotai{\color{red}\ensuremath}
【答】
\inputHakoKaiFile
```

$$(x+a)^2 = x^2 + \boxed{\text{ウ}}x + \boxed{\text{エ}}$$

【答】ウ = $2a$ エ = a^2

デフォルトは

`\hakovaisyotai{\ensuremath}`

が実行された状態で、答えは数式モードに入る、というのは今まで通りです。

この ¥ensuremath を外してしまえば、答えはテキストモードで表されます。

— 答えを ¥texttt で —

```
\openHakoKaiFile
つぎの\karaHako に下の選択肢の中から
最も適切なものを選び，記号で答えよ。
\[ (x+t)^2=x^2+\Hako'(c)'x+\Hako'(d)'
\]
```

選択肢：

```
\begin{jquote}
\begin{edaenumerate}[\ttfamily(a)]
  \item 0
  \item $t$
  \item $2t$
  \item $t^2$
\end{edaenumerate}
\end{jquote}
\closeHakoKaiFile
```

```
\hakovaisyotai{\ttfamily}
```

【答】

```
\HakoKaiKata{t}
\inputHakoKaiFile
```

つぎの に下の選択肢の中から最も適切なものを選び，記号で答えよ。

$$(x+t)^2 = x^2 + \text{オ} x + \text{カ}$$

選択肢：

- (a) 0 (b) t
(c) $2t$ (d) t^2

【答】

オ	カ
(c)	(d)

18 ハコの中を網掛け

ハコを2種類区別したいときに，一方を網掛けする，という出題をする大学もあります。

そのために，`\amiHako` というコマンドを作りました。ただし，これは `hako.sty` ではなく，`arhako.sty` という別ファイルにしていますから，それを読み込む必要があります。また，読み込む順序も問題で `hako.sty` を先に，`arhako.sty` をその後に読み込む必要があります。さらに，このパッケージは `ascmac.sty` を必要とします。

一例を挙げます。

```

\documentclass{jarticle}
\usepackage{amssymb,ascmac,emath,hako,arhako}

\begin{document}
\hakosyokika
次の\Hako/ /にあてはまる数値を答えよ .
また , \amiHako/ /にあてはまる記号を下の選択肢から選び ,
その番号を答えよ .

 $x$ についての方程式 $4x^2-2ax+\sqrt{3}=0$ の2つの解が
 $\sin\theta$ ,  $\cos\theta$ であるという .
ただし ,  $0\text{Deg}\leq\theta\leq 45\text{Deg}$ で ,  $a$ は定数である .
このとき

$$[ a^2=\Hako\amiHako\Hako\sqrt{\Hako} ]$$

したがって ,  $0\text{Deg}\leq\theta\leq 45\text{Deg}$ に注意すると

$$[a=\Hako\amiHako\Hako\sqrt{\Hako},$$


$$\quad\theta={\Hako\Hako}\text{Deg}]$$

である .

\begin{itemize}\itemindent2zw\item[【選択肢】]
\renewcommand{\labelenumi}{\maru{\theenumi}}%
\begin{edaenumerate}<4>
\ttfamily\edaitemindent2zw%
\edaitem{+}
\edaitem{-}
\edaitem{\times}
\edaitem{\div}
\end{edaenumerate}
\end{itemize}
\end{document}

```

をタイプセットすると

次の にあてはまる数値を答えよ．また， にあてはまる記号を下の選択肢から選び，その番号を答えよ．

x についての方程式 $4x^2 - 2ax + \sqrt{3} = 0$ の 2 つの解が $\sin \theta, \cos \theta$ であるという．ただし， $0^\circ \leq \theta \leq 45^\circ$ で， a は定数である．このとき

$$a^2 = \text{ア} \text{イ} \text{ウ} \sqrt{\text{エ}}$$

したがって， $0^\circ \leq \theta \leq 45^\circ$ に注意すると

$$a = \text{オ} \text{カ} \text{キ} \sqrt{\text{ク}}, \quad \theta = \text{ケ} \text{コ}^\circ$$

である．

【選択肢】

1 +

2 -

3 ×

4 ÷

となります．

コマンド `\amiHako` は，網掛けフラッグを立てて，`\Hako` を呼び出しています．従って，オプション引数の使い方等は `\Hako` と同様で，相互参照なども可能ですが，参照も網掛けのハコにしたいときは `\refamiHako*` を用います．

19 ハコいろいろ

19.1 ハコの文字の間に縦仕切線

ハコの中に複数の文字が入るとき，その文字の間に縦の仕切線を入れる流儀もあります．そのためのコマンドが `\renHako<文字数>` です．例えば，

```

\renHako
\hakoyohaku{2pt}%
$\bunsuu{\renHako<3>}{\renHako<4>}$, あるいは
$\bunsuu{\renHako<2>}{\Hako}$ などと用います．
```

をタイプセットすると

ア	イ	ウ	
エ	オ	カ	キ

, あるいは

ク	ケ
コ	


などを用います．

となります．このコマンドの用い方は，`\Hako` とほぼ同様です．ただし，このコマンドは `hako.sty` ではなく，`arhako.sty` で定義されていますから，`hako.sty` に引き続き `arhako.sty` を読み込む必要があります．なお，このコマンドは `\hakoxyohaku{0pt}` を内部で実行します．

19.2 破線の仕切

仕切り線を破線にすることも可能です。


```
— *sikirisen —  
\hakosyokika  
\sikirisen{\hasen}{0.4pt}{あ}  
\renHako<3>
```



新機能を利用するには、まず、`arhako.sty` を `[tpic]` オプションをつけて読み込みます：

```
\usepackage[tpic]{arhako}
```

次いで、コマンド `*sikirisen` により、仕切り線の形状を指定します。`*sikirisen` の書式は：

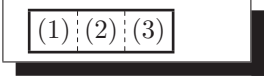


```
\sikirisen#1#2#3  
#1 : 仕切り線を引く描画関数  
      ex: \drawline(default), \hasen, \hasen[L=2pt,G=1pt],....  
#2 : 仕切り線の太さ  
#3 : ハコに入る, 最大の高さ・深さを持つ文字
```

破線をもう少し細かくしてみます。

破線を細かく

```
\hakosyokika  
\hakosyotai{\normalfont}  
\hakomozisyu{(1)}  
\sikirisen{\hasen[L=1.5pt,G=1.5pt]}{.2pt}{(1)}  
\renHako<3>
```



注 この機能は `tpic-specials` を用いています。したがって、これをサポートしていない `dvi-ware` では使用できません。

19.3 先頭文字を で囲む

まずは、具体例をご覧ください。

○で囲んだ文字には符号(+, -), 不等号(<, ≤)があてはまる。求めた数値が0の場合は+0とする。

$0 < x < 1$ …… ①, $|x - a| < 2$ …… ②とする。ここで, ①を満たすどのような x についても②が満たされるとき, この実数 a の範囲は

アイ ウ a エ カ

である。また, ①を満たすある x について②が満たされるとき, 実数 a の範囲は

キ ケ a コ シ

である。

【解答】

<input type="checkbox"/> アイ	<input type="checkbox"/> ウ	<input type="checkbox"/> エ	<input type="checkbox"/> カ	<input type="checkbox"/> キ	<input type="checkbox"/> ケ	<input type="checkbox"/> コ	<input type="checkbox"/> シ
-1	≤	≤	+2	-2	<	<	+3

[2005 東洋大学]

\centermodetrue のとき,

ハコに入る文字列の先頭を で囲み,

ここには符号など数字以外のものが入ることを明示する

という表現法に対応するためのコマンドが\tmHako です。

¥tmHako

```
\hakosyokika
\centermodetrue
\openHakoKaiFile
\Hako'12'

\tmHako'-345'

\Hako'-67'
\closeKaiFile

\verb+-----+

\HakoKaiKata{t}
\inputHakoKaiFile
```

アイ		
<input type="checkbox"/> エオカ		
キクケ		

アイ	<input type="checkbox"/> エオカ	キクケ
12	-345	-67

20 ¥HAKO — 高速版 ¥Hako

¥Hako が多くのオプション引数をとるようになったため, 処理速度が低下してきました。そこで使用頻度の高いオプションだけをとるコマンド ¥HAKO を作りました。つけることのできるオプションは

- <.> 文字個数オプション

- [...] ラベルオプション (全体ラベルのみ)
- (...) 横巾変更オプション

の3個だけです。その他のオプションをつけたいときは、従来の \Hako を使用してください。

\HAKO と \Hako の混在使用は可能で、番号は両者で共通にインクリメントされます。その一例を見ていただきましょう。下の例は、2文字を含むハコにおいて、個別ラベル (第2の [...] オプション) を付ける2箇所だけ \Hako を使用し、他のハコには \HAKO を用いています。また、 \HAKO で付けたラベルを用いての参照は、 \Hako に対すると同じく \refHako または $\refHako*$ を用います。

— \HAKO と \Hako の混用 —

\backslash hakosyokika

点 A を中心とする半径 2 の円と、点 B を中心とする半径 3 の円が点 C で外接している。点 D は半径 2 の円上に、また点 E は半径 3 の円上にあり、直線 DE は二つの円の共通接線となっている。

点 C における二つの円の共通接線と直線 DE との交点を F とし、直線 DA と直線 EC の交点を G とする。 \backslash par

このとき、 $\text{\text{DE}}=\text{\HAKO}\sqrt{\text{\HAKO}}$ である。 \backslash par

また、 $\text{\text{AD}} \parallel \text{\text{BE}}$ から、

$\text{\triangle}\text{\Hako}<2>[\text{Lue}][\text{Lu},\text{Le}]\text{\text{C}}$

と $\text{\sankaku}\{\text{AGC}\}$ は相似であるので、線分 AG と

線分 $\text{\Hako}<2>[\text{Loka}][\text{Lo},\text{Lka}]$ の長さは等しくなる。

($\text{\refHako}*\{\text{Lue}\}$ の $\text{\refHako}\{\text{Lu}\}$ と $\text{\refHako}\{\text{Le}\}$ 、

$\text{\refHako}*\{\text{Loka}\}$ の $\text{\refHako}\{\text{Lo}\}$ と $\text{\refHako}\{\text{Lka}\}$ については、解答の順序を問わない。) \backslash par

\backslash par

したがって、点 G は点 A を中心とする半径 2 の円上にあり、

$\text{\kaku}\{\text{GCD}\}=\text{\HAKO}<2>\text{\Deg}$ となる。

をタイプセットすると

点 A を中心とする半径 2 の円と、点 B を中心とする半径 3 の円が点 C で外接している。点 D は半径 2 の円上に、また点 E は半径 3 の円上にあり、直線 DE は二つの円の共通接線となっている。点 C における二つの円の共通接線と直線 DE との交点を F とし、直線 DA と直線 EC の交点を G とする。

このとき、 $DE = \boxed{\text{ア}} \sqrt{\boxed{\text{イ}}}$ である。

また、 $AD \parallel BE$ から、 $\triangle \boxed{\text{ウエ}}C$ と $\triangle AGC$ は相似であるので、線分 AG と線分 $\boxed{\text{オカ}}$ の長さは等しくなる ($\boxed{\text{ウエ}}$ のウとエ、 $\boxed{\text{オカ}}$ のオとカについては、解答の順序を問わない。)

したがって、点 G は点 A を中心とする半径 2 の円上にあり、 $\angle GCD = \boxed{\text{キク}}^\circ$ となる。

となります。

21 コマンドリファレンス (1)

\Hako<ハコの中の文字数>(ハコの幅)(位置指定)

[全体ラベル][個別ラベル]"ハコの中の文字"

'ハコの答'/ハコの中の文字種/

第1引数 <ハコの中の文字数(デフォルトは1)>

第2引数 (ハコの横幅の余白サイズ)

第3引数 (位置指定)

c : 中央(デフォルト)

l : 左寄せ

^ : 上付き(必然的に左寄せ)

| : 文字の右に縦罫線(必然的に左寄せ)

L : 左外上付き

b : ハコの baseline をハコの下枠線とする .

t : ハコの baseline をハコの上枠線とする .

第4引数 [全体ラベル]

第5引数 [個別ラベル]

第1引数で複数の文字を指定した場合, その数だけラベルを, で区切って並べる .

第6引数 "ハコの中の文字"

カウンタの値とは無関係に, ハコの中に入る文字を直接指定する .

第7引数 'ハコの答'

ハコに入れるべき答を記述しておいて, 問題の後ろにタイプセットする .

第8引数 /ハコの中の文字種/

ア : ア, イ, ウ, エ, オ,..... (デフォルト)

イ : イ, ロ, ハ, ニ, ホ,.....

あ : あ, い, う, え, お,.....

い : い, ろ, は, に, ほ,.....

1 : 1, 2, 3, 4, 5,.....

i : i, ii, iii, iv, v,.....

I : I, II, III, IV, V,.....

a : a, b, c, d, e,.....

A : A, B, C, D, E,.....

m : 丸付き数字

一 : 一, 二, 三, 四, 五,.....

注1 . その他の文字はそのまま出力されますから ,

/(1)/ と指定すれば ,

(1),(2),(3),(4),(5),.....

と, ナンバリングされます .

注2 . デフォルトは /ア/ ですが ,

\hakomozisyu{い}

などと、宣言すれば、以降デフォルトの文字種が /い/ に変更されます。
すなわち、単に \Hako で
い, ろ, は, に, ほ,
と出力されます。

\refHako/ハコの中の文字種/{ラベル}

\Hako でつけた”ラベル”を取り出します。

\refHako*(ハコの幅)(位置指定)/ハコの中の文字種/{ラベル}

\Hako でつけた”ラベル”をハコの枠をつけた形で取り出します。(ハコの幅)(位置指定) オプションは \Hako と同様です。

22 コマンドリファレンス (2)

`\hakosyokika`

`hako.sty` の内部変数などを初期化します。このパッケージを使用するときは、最初に呼ぶ必要があります。

`\hakobansyokika`

ハコのカウンタをすべて初期化 (0) します。

`\hakomozisyu{..}`

ハコの中の文字種を変更します (第 5 節参照)

`\hakosenhaba{..}`

ハコの枠線の太さを変更します (第 9 節参照)

`\hakosityuu{..}`

ハコの高さを指定します。(第 13 節参照))

`\hakosyotai{..}`

ハコの中の文字の書体を設定します (第 8 節参照)

`\hakoxyohaku{..}`

ハコの横余白量を設定します (第 10 節参照)

`\hakoyohaku{..}`

ハコの中の文字と枠線の間 (上下左右) の余白量を設定します。デフォルト値は、`\fboxsep` と同じく 3pt です。

23 附 記号を付与しないハコ

`hako.sty` では、ハコに

ア, イ, ウ, ……

と記号を付与し、それを相互参照することを目的としていますが、空のハコを使いたい、あるいは答えを埋め込みたいというご要望もあります。それに答えるのが

`\karaHako`

`\maskHako`

です。

23.1 ¥karaHako

次の \karaHako を補え。

をタイプセットすると

次の を補え。

となります。

¥karaHako の書式は

```
\karaHako(#1)[#2][#3]
#1 : 横幅 (デフォルト = 2zw)
#2 : 高さ (デフォルト = 1zh)
#3 : 深さ (デフォルト = 0zh)
```

23.2 ¥maskHako

¥karaHako では、空白のハコを表示するだけですが、答えを埋め込んで置き、ハコの中を空白にしておくか、答えを表示するかは、¥ifmaskhako の真偽値でコントロールしよう、というのが ¥maskHako です。ただし、このコマンドは color パッケージを必要とします。

次の \karaHako を補え。

$f(x)=x^n$ の導関数は
 $\backslash[f'(x)=\backslashmaskHako(4zw)[2zh][1zh]\{nx^{\{n-1\}}\}\backslash$
である。

新しいコマンドは ¥maskHako で、その書式は

次の を補え。

$f(x) = x^n$ の導関数は

$f'(x) =$

である。

`\maskHako(#1)[#2][#3]#4`
 #1 : ハコの横幅
 #2 : ハコの高さ
 #3 : ハコの深さ
 #4 : 答え (数式モード内と仮定されています)

`\ifmaskhako`
`\maskhakottrue` (デフォルト) のときは
 #1,#2,#3 で指定されたサイズの空ハコを作る。
 #1,#2,#3 を指定しないときは #4 が納まるハコを作る。
`\maskhakofalse` とすると
 #1,#2,#3 で指定されたサイズのハコ中央に #4 を示す。

です。デフォルトでは `\maskhakottrue` すなわち、答えは表示せず、ハコの枠のみを示します。
`\maskhakofalse` と宣言しますと

`\ifmaskhako`
`\maskhakofalse`
 次の `\karaHako` を補え。

 $f(x) = x^n$ の導関数は
`\[f'(x) = \maskHako(4zw)[2zh][1zh]{nx^{n-1}}\]`
 である。

次の を補え。
 $f(x) = x^n$ の導関数は
 $f'(x) = \span style="border: 1px solid black; display: inline-block; padding: 2px 10px;">nx^{n-1}$
 である。

と、ハコの中に答えが表示されます。

答えを赤で表示したい、というとき `\color` 文で指定します。

`\ifmaskhako` 解答を赤で埋め込む `\maskhakofalse`
`\[f'(x) = \maskHako(4zw)[2zh][1zh]{\color{red}2x}\]`

$(x^2)' = \span style="border: 1px solid black; display: inline-block; padding: 2px 10px; color: red;">2x$

いちいち `\color{red}` と記述するのは面倒だというなら、`\everymaskHako` を用います。

¥everymaskHako

```
\maskhakofalse
\everymaskHako{\color{red}}
次の\karaHako を補え。
\begin{align*}
{(x^2)'}&=&\maskHako(4zw)[2zh][1zh]{2x} \\
{(x^3)'}&=&\maskHako(4zw)[2zh][1zh]{3x^2}
\end{align*}
```

次の を補え。

$$(x^2)' = \boxed{2x}$$

$$(x^3)' = \boxed{3x^2}$$

この指定法はすべての ¥maskHako に対して働きます。デフォルト状態に戻したいときは

```
\everymaskHako{\relax}
```

とします。

さて、マスクできるでしょうか。

¥maskhakottrue も有効

```
\maskhakottrue
\everymaskHako{\color{red}}
次の\karaHako を補え。
\begin{align*}
{(x^2)'}&=&\maskHako(4zw)[2zh][1zh]{2x} \\
{(x^3)'}&=&\maskHako(4zw)[2zh][1zh]{3x^2}
\end{align*}
```

次の を補え。

$$(x^2)' = \boxed{}$$

$$(x^3)' = \boxed{}$$

蛇足ながら ¥color を使用するには、color パッケージが必要です。

太字指定 (`\bm`) は、少々面倒です。

¥bm 指定

```
\maskhakofalse
\everymaskHako[$]{\bm}
次の\karaHako を補え。
\begin{align*}
{(x^2)'}&=&\maskHako(4zw)[2zh][1zh]{2x}\backslash
{(x^3)'}&=&\maskHako(4zw)[2zh][1zh]{3x^2}
\end{align*}
```

次の を補え。

$$(x^2)' = \boxed{2x}$$

$$(x^3)' = \boxed{3x^2}$$

すなわち

```
\everymaskHako[#1]#2
\maskHako{なんたら}
```

は次のように展開されます：

```
#2{\ensuremath{なんたら}}#1
```

すなわち，上の例では

```
$_\bm{\ensuremath{2x}}$
```

と展開されます。

最後に，赤で太字の指定です。

赤で太字

```
\maskhakofalse
\everymaskHako[$]{\color{red}\bm}
次の\karaHako を補え。
\begin{align*}
{(x^2)'}&=&\maskHako(4zw)[2zh][1zh]{2x}\backslash
{(x^3)'}&=&\maskHako(4zw)[2zh][1zh]{3x^2}
\end{align*}
```

次の を補え。

$$(x^2)' = \boxed{2x}$$

$$(x^3)' = \boxed{3x^2}$$

マスクの確認をしましょうか。

—— 赤で太字 ——

```
\maskhakotrue
\everymaskHako[$]{\color{red}$\bm}
次の\karaHako を補え。
\begin{align*}
  \{(x^2)'\}&=&\maskHako(4zw)[2zh][1zh]{2x} \\
  \{(x^3)'\}&=&\maskHako(4zw)[2zh][1zh]{3x^2}
\end{align*}
```

次の を補え。

$$(x^2)' = \boxed{}$$

$$(x^3)' = \boxed{}$$