

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

文部科学省後援

情報システム試験

令和4年度前期 情報検定

<実施 令和4年9月11日(日)>

プログラミングスキル

(説明時間 10:00~10:10)

(試験時間 10:10~11:40)

- ・試験問題は試験開始の合図があるまで開かないでください。
- ・解答用紙(マークシート)への必要事項の記入は、試験開始の合図と同時に行いますので、それまで伏せておいてください。
- ・試験開始の合図の後、次のページを開いてください。<受験上の注意>が記載されています。必ず目を通してから解答を始めてください。
- ・試験問題は、すべてマークシート方式です。正解と思われるものを1つ選び、解答欄の○をHBの黒鉛筆でぬりつぶしてください。2つ以上ぬりつぶすと、不正解になります。
- ・辞書、参考書類の使用および筆記用具の貸し借りは一切禁止です。
- ・電卓の使用が認められます。ただし、下記の機種については使用が認められません。

<使用を認めない電卓>

1. 電池式(太陽電池を含む)以外の電卓
2. 文字表示領域が複数行ある電卓(計算状態表示の一行は含まない)
3. プログラムを組み込む機能がある電卓
4. 電卓が主たる機能ではないもの
 - *パソコン(電子メール専用機等を含む)、携帯電話(PHS)、スマートフォン、タブレット、電子手帳、電子メモ、電子辞書、翻訳機能付き電卓、音声応答のある電卓、電卓付き腕時計、時計型ウェアラブル端末等
5. その他試験監督者が不適切と認めるもの

一般財団法人 職業教育・キャリア教育財団

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

＜受験上の注意＞

1. この試験問題は35ページあります。ページ数を確認してください。
乱丁等がある場合は、手をあげて試験監督者に合図してください。
※問題を読みやすくするために空白ページを設けている場合があります。
2. 解答用紙（マークシート）に、受験者氏名・受験番号を記入し、受験番号下欄の数字をぬりつぶしてください。正しく記入されていない場合は、採点されませんので十分注意してください。
3. 試験問題についての質問には、一切答えられません。自分で判断して解答してください。
4. 試験中の筆記用具の貸し借りは一切禁止します。筆記用具が破損等により使用不能となった場合は、手をあげて試験監督者に合図してください。
5. 試験を開始してから30分以内は途中退出できません。30分経過後退出する場合は、もう一度、受験番号・マーク・氏名が記載されているか確認して退出してください。なお、試験終了5分前の合図以降は退出できません。試験問題は各自お持ち帰りください。
6. 試験後の合否結果（合否通知）、および合格者への「合格証・認定証」はすべて、Web認証で行います。
 - ①情報検定（J検）Webサイト合否結果検索ページ及びモバイル合否検索サイト上で、デジタル「合否通知」、デジタル「合格証・認定証」が交付されます。
 - ②団体宛には合否結果一覧ほか、試験結果資料一式を送付します。
 - ③合否等の結果についての電話・手紙等でのお問い合わせには、一切応じられませんので、ご了承ください。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

<問題の構成>

必須問題 全員解答

問題 1 ～ 問題 4	2 ページ～17 ページ
-------------------------	--------------

選択問題 次の問題から 1 問選択し解答せよ。

(選択した問題は解答用紙「選択欄」に必ずマークすること)

※選択欄にマークがなく、解答のみマークした場合は採点を行いません。

・ C 言語の問題	19 ページ～24 ページ
・ 表計算の問題	25 ページ～32 ページ
・ アセンブラの問題	33 ページ～35 ページ

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

必須問題

問題1 次のリスト構造に関する記述を読み、各設問に答えよ。

リスト構造とはデータ構造の一つであり、データとデータの格納場所を示すポインタから構成されている。ポインタを利用することでデータの追加、削除などを簡単に行えることが特徴である。

<設問1> 次のリスト構造のデータ探索に関する記述中の に入れるべき適切な字句を解答群から選べ。

[リスト構造]

リスト構造が図1の状態である場合、次の手順でデータ探索するプログラムを考える。

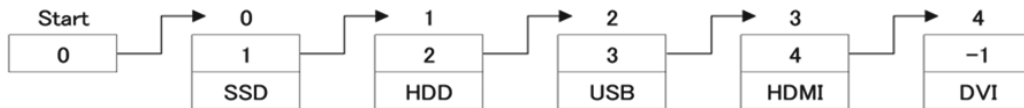


図1 リスト構造

Start に先頭ポインタの値として0が格納されている。アドレス0のデータを参照すると1とSSDが格納されている。ここで、1は次へのポインタの値であり、SSDはアドレス0に格納されているデータである。データを探索する場合は、格納されているデータと探索したいデータを照合し、一致していた場合は格納場所を返すものとする。一致していない場合は、ポインタが指し示す次のデータを参照する。これをデータが一致する、またはポインタの値が-1になるまで繰り返す。

ここでHDMIを探索したいデータとした場合、返却される値は (1) である。

(1) の解答群

- | | | |
|-------|------|------|
| ア. -1 | イ. 0 | ウ. 1 |
| エ. 2 | オ. 3 | カ. 4 |

<設問2> 次のリスト構造のデータ参照順序に関する記述中の に入れるべき適切な字句を解答群から選べ。

探索対象データが見つかった場合に、リストの先頭に一致したデータを再配置する場合を考える。例えばリストが図1の状態からHDDを探索した場合、探索したいデータであるHDDはリスト内に存在するため、ポインタを操作することでHDDをリストの先頭に移動することができる。移動した結果が図2の状態である。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

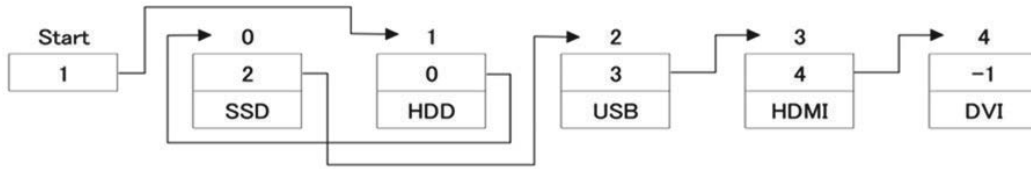


図2 再配置後のリスト構造

再配置が完了した状態で、Start が指し示すポインタの値から順番にポインタが指し示す格納場所をたどっていくと HDD → SSD → USB → HDMI → DVI の順番にアクセスできる。

次に、図1の状態から USB を探索した場合、Start の値を に変更し、リストの USB のポインタの次のポインタを に変更する。最後に、アドレス の次のポインタを 3 に変更する必要がある。すべての変更を終え、Start が指し示すポインタの値から順番にポインタにアクセスしていくと の順番にアクセスできる。

また、図1の状態から HDMI を探索して並べ替え、並べ替えデータをもとに HDD を探索して並べ替えた場合を考える。すべての変更を終え、Start が指し示すポインタの値から順番にポインタをたどっていくと の順番にアクセスできる。

(2) ~ (4) の解答群

- | | | |
|-------|------|------|
| ア. -1 | イ. 0 | ウ. 1 |
| エ. 2 | オ. 3 | カ. 4 |

(5) の解答群

- ア. HDD → SSD → USB → HDMI → DVI
イ. HDMI → USB → HDD → SSD → DVI
ウ. USB → HDD → SSD → DVI → HDMI
エ. USB → SSD → HDD → HDMI → DVI

(6) の解答群

- ア. HDD → HDMI → SSD → USB → DVI
イ. HDD → USB → HDMI → SSD → DVI
ウ. HDMI → HDD → SSD → USB → DVI
エ. HDMI → USB → HDD → SSD → DVI

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

問題2 次の挿入法によるデータの整列に関する説明を読み、各設問に答えよ。

1次元配列 $h[0] \sim h[4]$ までデータが格納済みである。このデータに対して挿入法により降順に整列する流れ図である。次に示す「挿入法の手順」を m が 1~4 まで繰り返すことにより整列が完了する。

[挿入法の手順]

$h[0] \sim h[m-1]$ まで降順に整列されているとき、 $h[m]$ を格納すべき位置を見つけて挿入する手順は、次のようになる。

- ① $h[m]$ の内容を w に退避する。
- ② k を $m-1$ とする。
- ③ $h[k]$ の内容が w 以上であれば、格納すべき位置を見つけたことになるので④に進む。そうでなければ、 $h[k]$ の内容を $h[k+1]$ へ格納して k から 1 を引き、③へ戻る。
- ④ w を $h[k+1]$ へ格納する。

<設問1> 挿入法による処理が進行中の図1の状態です。空欄に入れるべき適切な字句を解答群から選べ。

配列 h の初期値

添字→	0	1	2	3	4
配列 h	18	28	32	21	25

$m=1$ のとき

添字→	0	1	2	3	4
配列 h	28	18	32	21	25

h[1]を挿入

整列済み部分 未整列部分

$m=2$ のとき

添字→	0	1	2	3	4
配列 h			(1)		

h[2]を挿入

$m=3$ のとき

添字→	0	1	2	3	4
配列 h			(2)		

h[3]を挿入

$m=4$ のとき

添字→	0	1	2	3	4
配列 h	32	28	25	21	18

h[4]を挿入

図1 挿入法による処理が進んでいく過程

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

(1) , (2) の解答群

ア.

32	21	18	25	28
----	----	----	----	----

イ.

32	25	21	18	28
----	----	----	----	----

ウ.

32	25	18	21	28
----	----	----	----	----

エ.

32	25	18	28	21
----	----	----	----	----

オ.

32	28	18	21	25
----	----	----	----	----

カ.

32	28	21	18	25
----	----	----	----	----

<設問2> 図2の挿入法を用いた流れ図中の に入れるべき適切な字句を解答群から選べ。

(3) の解答群

ア. $h[k-1] : h[k]$

イ. $h[k] : w$

ウ. $h[m] : w$

(4) の解答群

ア. $k \leftarrow 1$

イ. $k \leftarrow k-1$

ウ. $k \leftarrow k+1$

(5) の解答群

ア. $h[k] \leftarrow w$

イ. $h[k+1] \leftarrow w$

ウ. $h[m] \leftarrow w$

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

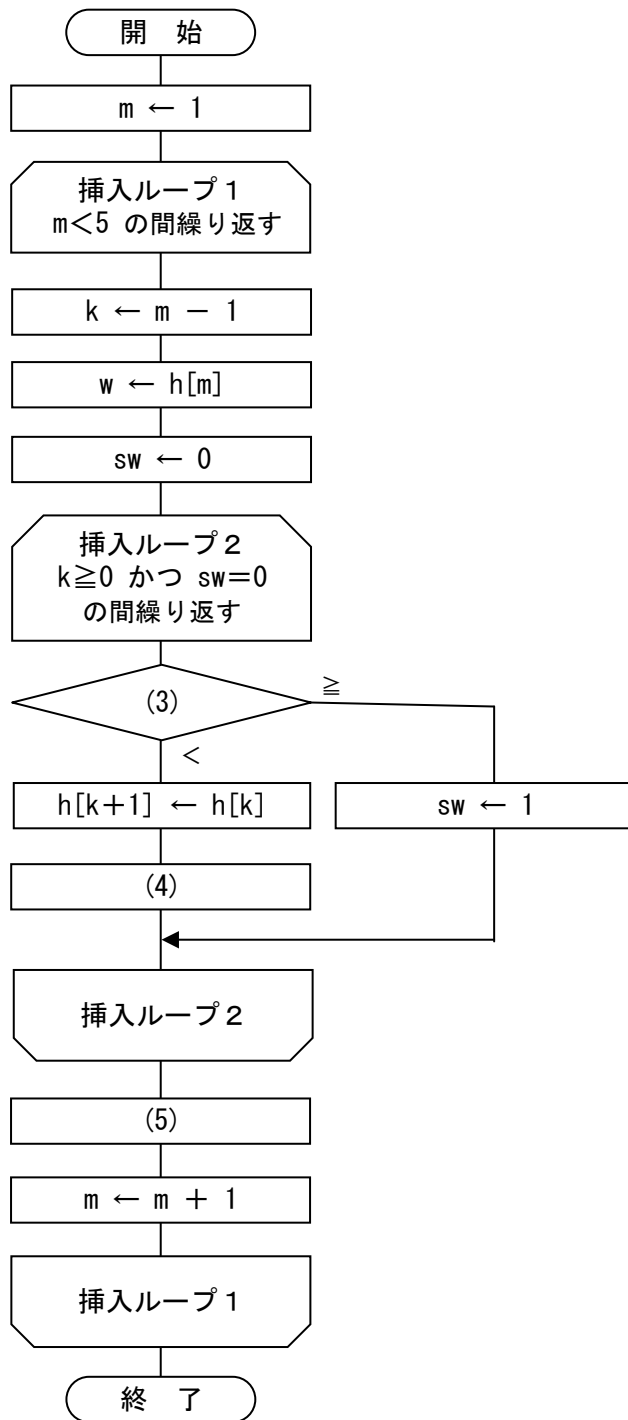


図2 挿入法を用いた流れ図

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

問題3 次の文字列の操作に関する記述を読み、各設問に答えよ。

配列 `str` に格納された文字列(以下, `str`)から、配列 `del` に格納された文字列(以下, `del`)を削除する処理を考える。なお、文字列は配列の1要素に1文字ずつ格納されており、配列の要素位置は0から始まる。

削除する処理は、`str` から `del` を検索し、検索できた位置より後にある文字を前方へ移動することで行う。この時、空いた位置には空白文字(以下, □で表す)を設定するものとする。

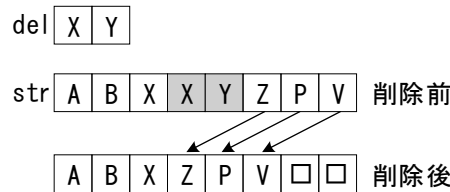


図1 文字列を削除する

[文字列の検索について]

`str` の各位置において、`del` が出現しているかどうかを調べる。

`str` と `del` を要素位置の小さい方から1つずつ順番に比較し、`del` の最後の文字まで一致していれば検索できたことになる。途中で文字の不一致があれば、`str` の検索を開始した位置から1つ後に移動して比較を再開する。

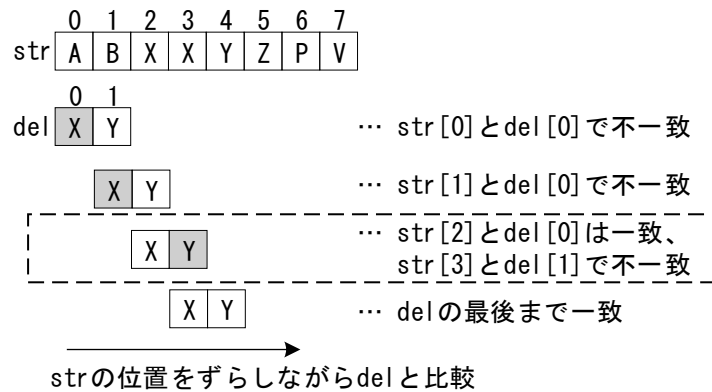


図2 文字列を検索する流れ

<設問1> 次の文字列検索に関する記述中の□に入れるべき適切な字句を解答群から選べ。

図2の破線で囲まれた部分では、`str`[2]と`del`[0]は一致しているため`str`[3]と`del`[1]の比較を行っている。`str`[3]と`del`[1]は不一致のため、次の検索は検索を開始した位置から1つ移動した位置からとなる。つまり、□(1)の比較から検索を再開する。再開時を考慮すると、`str`の添字は検索開始位置をどのように保持するかを考えなければならない。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

(1) の解答群

ア. `str[3]`と`del[0]`

イ. `str[3]`と`del[1]`

ウ. `str[4]`と`del[0]`

エ. `str[4]`と`del[1]`

<設問 2> 次の文字列検索の流れ図に関する記述を読み、流れ図中の に入るべき適切な字句を解答群から選べ。

[文字列検索の流れ図 (find) に関する説明]

`str` に格納された文字列から `del` に格納された文字列を検索し、検索ができた場合は `str` の開始位置を、検索できなかった場合は `-1` を返す。また、検索文字列が `str` 内に複数現れる場合は最初の文字列の情報を返却する。なお、`str` の文字数は `sLen`、`del` の文字数は `dLen` に設定されている。図 3 の実行例では、`1` を返却する。

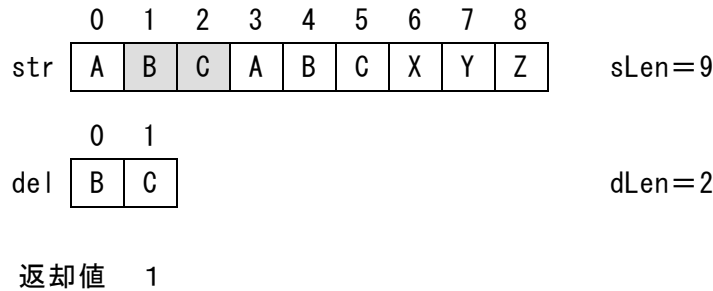


図 3 文字列検索の実行例

(2) の解答群

ア. `p = -1`

イ. `p ≠ -1`

ウ. `p > 0`

エ. `p ≥ 0`

(3) の解答群

ア. `k > 0`

イ. `k > 1`

ウ. `k < dLen`

エ. `k < sLen`

(4) の解答群

ア. `i ← i + 1`

イ. `i ← i + k`

ウ. `k ← k + 1`

エ. `k ← k + i`

(5) の解答群

ア. `sw : 0`

イ. `sw : 1`

ウ. `k : sLen`

エ. `k : sLen - 1`

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

(6) の解答群

ア. $i \leftarrow i + 1$

イ. $i \leftarrow i + p$

ウ. $p \leftarrow i$

エ. $p \leftarrow i + 1$

[流れ図]

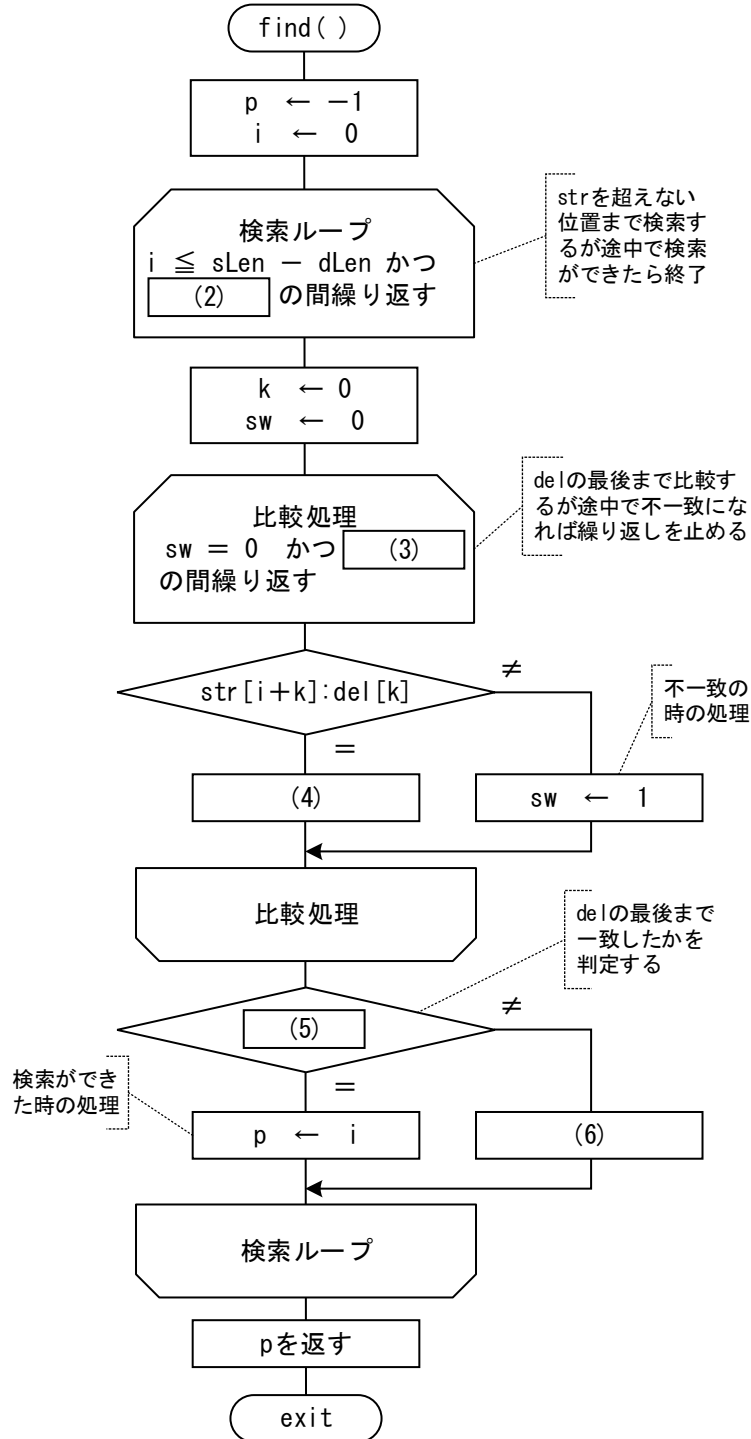


図4 findの流れ図

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

<設問 3 > 次の文字列削除に関する記述中の に入れるべき適切な字句を解答群から選べ。なお、str の文字数は sLen、del の文字数は dLen に設定されている。

図 5 の実行例では、str 中に del が検索できた位置は 1 である。

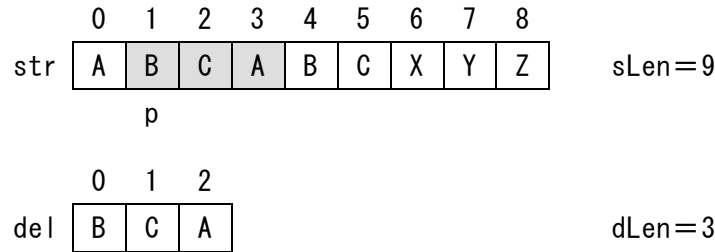


図 5 find の実行例

検索できたならば、削除処理を行うために配列内の文字を移動する。図 5 の実行例では、str[1]に (7) が格納され、str[2]へ (8) が格納されるように移動する。この移動は、str[5]への格納まで続け、str[6]～str[8]へは空白文字を格納する。

(7) , (8) の解答群

ア. str[3] イ. str[4] ウ. str[5] エ. str[6]

<設問 4 > 次の文字列削除の流れ図に関する記述を読み、流れ図中の に入れるべき適切な字句を解答群から選べ。

[文字列削除の流れ図に関する説明]

str に格納された文字列から del に格納された文字列を削除する。

str から del を検索する処理は、図 4 の find を使用する。また、str に del が複数存在する場合は、全て削除する。

なお、str の文字数は sLen、del の文字数は dLen に設定されており、実行後の sLen は空白文字を含めない削除後の文字数にする。

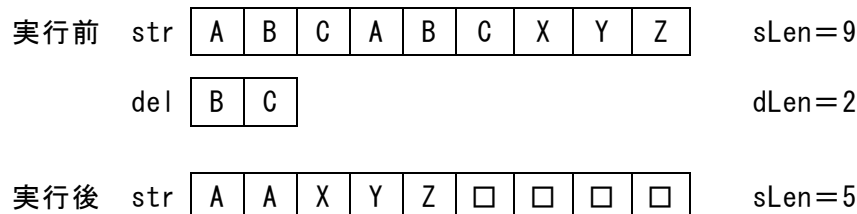


図 6 文字列削除の実行例

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

[流れ図]

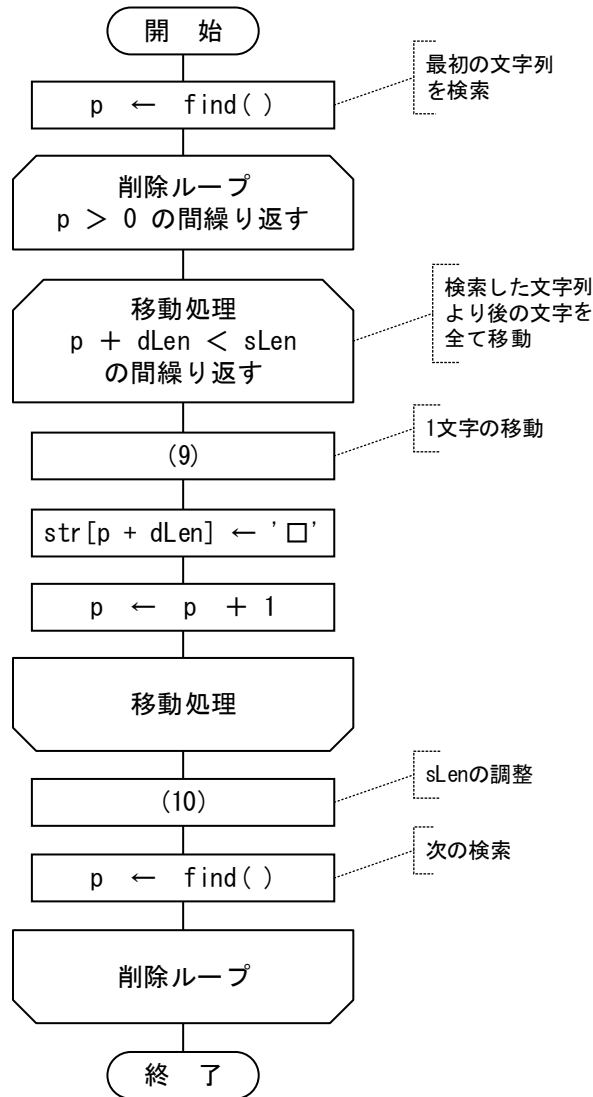


図7 文字列削除の流れ図

(9) の解答群

ア. $\text{str}[p] \leftarrow \text{str}[dLen]$

ウ. $\text{str}[p] \leftarrow \text{str}[p+1]$

イ. $\text{str}[p] \leftarrow \text{str}[p-1]$

エ. $\text{str}[p] \leftarrow \text{str}[p+dLen]$

(10) の解答群

ア. $sLen \leftarrow sLen - 1$

ウ. $sLen \leftarrow sLen - p$

イ. $sLen \leftarrow sLen + 1$

エ. $sLen \leftarrow sLen - dLen$

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

問題4 次のハフマン符号の説明を読み、プログラム中の に入れるべき適切な字句を解答群から選べ。

[ハフマン符号の説明]

ハフマン符号は、文字列の可逆圧縮などに用いられるものである。文字の出現頻度に応じて、頻度の高い文字には短い符号を、頻度の低い文字には長い符号を割り当てることによって文字列全体を圧縮する方法である。

[プログラムの説明]

ここでは、ハフマン符号の生成を関数 Huffman で、次の手順により行う。

I. 文字列(in_data[])中の文字の出現回数を求める。

文字列が、“ABBBCDAABBBBBBBBCDD”文字列の長さ(in_len)が20の場合、出現文字は4種類(m_len)、表1のような出現回数となる。

表1 各文字の出現回数

文字	A	B	C	D
出現回数	3	12	2	3

II. 出現回数の低い方から文字を二つ選び、それらを葉とする2分木(ハフマン木)を作成し、親となる節には葉の出現回数の合計を記述する。出現回数の低い方を左側の葉とし、節の値が同じ時の順序は、並べ替える直前の順序に従う。

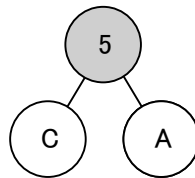


図1 最初の2分木

次に、残った文字と追加した節の出現回数の中から同様に出現回数の低い二つを選び2分木を作成し、その出現回数の合計を新しい親に記述する。

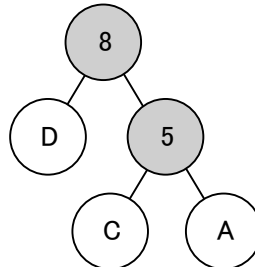


図2 次の2分木

これらの操作を、すべての文字が一つの2分木に組み込まれるまで繰り返す。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

Ⅲ. ここでは、2分木の左の枝に“0”右の枝に“1”を割り当てる。文字が格納された葉にたどりつくまで繰り返し、枝の数字を並べたもの(ビット列)が、その文字に割り当てられたハフマン符号となる。

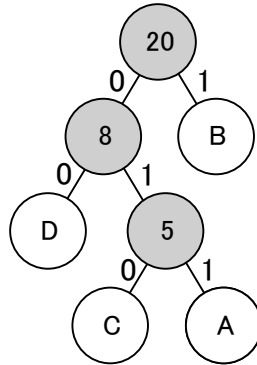


図3 完成した2分木

表2 データの文字とハフマン符号

文字	ハフマン符号
A	011
B	1
C	010
D	00

なお、図3の2分木の要素を配列で表現すると、次のようになる。配列の初期値は、文字の場合は空白、数字の場合は-1とし、各配列の添字は0から始まる。

要素番号 →	0	1	2	3	4	5	6
文字 : moji []	A	B	C	D			

出現回数 : m_cnt []

3	12	2	3	5	8	20
---	----	---	---	---	---	----

※ 要素番号4からは、親の値となる

節の親の要素番号 : parent []

4	6	4	5	5	6	-1
---	---	---	---	---	---	----

節の左側の子の要素番号 : left []

-1	-1	-1	-1	2	3	5
----	----	----	----	---	---	---

節の右側の子の要素番号 : right []

-1	-1	-1	-1	0	4	1
----	----	----	----	---	---	---

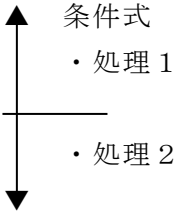
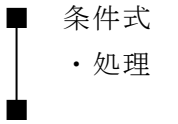
図4 図3の2分木の配列

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

[擬似言語の記述形式の説明]

記述形式	説明
○	手続き, 変数などの名前, 型などを宣言する。
・変数 ← 式	変数に式の値を代入する。
/* 文 */	注釈を記述する。
 条件式 ・処理 1 ・処理 2	選択処理を示す。 条件式が真の時は処理 1 を実行し, 偽の時は処理 2 を実行する。
 条件式 ・処理	前判定繰り返し処理を示す。 条件式が真の間, 処理を実行する。

[演算子と優先順位]

演算の種類	演算子	優先順位
単項演算	+, -, not	高 ↑ ↓ 低
乗除演算	*, /, %	
加減演算	+, -	
関係演算	>, <, ≥, ≤, =, ≠	
論理積	and	
論理和	or	

注記 整数同士の除算では, 整数の商を結果として返す。%演算子は剰余算を表す。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

[プログラム]

```
○ Huffman( 文字型配列 : in_data[], moji[],
            整数型配列 : m_cnt[], node[], parent[], left[], right[],
            整数型 : in_len )
○ 整数型 : k, m, m_len, cnt, kosu, first, second
/* node は親が生成されていない節の要素番号を格納する */
/* 出現文字ごとの回数を求める */
・ k ← 0
・ m_len ← 0
■ k < in_len
  ・ m ← 0
  ■ m < m_len
    ↑ moji[m] ≠ in_data[k] /* 以前に出現している文字か? */
    ↓
    ・ m ← m + 1
    ↓
    ・ break /* 繰り返しを抜ける */
  ■
  ↑ m = m_len
  ・ moji[m] ← in_data[k] /* まだ出現していない文字なら, 格納する */
  ・ (1)
  ・ m_len ← m_len + 1
  ・ m_cnt[m] ← m_cnt[m] + 1
  ↓
  ・ (2)
■

/* ハフマン木を作成する */
・ cnt ← m_len
・ kosu ← tree( m_len, parent, node, m_cnt )
■ kosu ≥ 2 /* 親が作成されていない文字が二つ以上ある */
  ・ first ← (3) /* 一番目に少ない値をもつ要素番号 */
  ・ second ← (4) /* 二番目に少ない値をもつ要素番号 */
  ・ m_cnt[m_len] ← m_cnt[first] + m_cnt[second] /* 子の値の合計 */
  ・ left[m_len] ← first
  ・ right[m_len] ← second
  ・ parent[first] ← m_len /* 子に親の節の要素番号を格納 */
  ・ parent[second] ← m_len /* 子に親の節の要素番号を格納 */
  ・ m_len ← m_len + 1
  ・ kosu ← tree( m_len, parent, node, m_cnt )
■
```

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

```
/* ハフマン符号を作成 */
```

```
• k ← 0
```

```
■ k < cnt
```

```
• moji[k]を出力する
```

```
• out_bit( k, parent, left )      /* 各文字のビットを出力する */
```

```
• k ← k + 1
```

```
■
```

```
/* 親が抽出されていない節の抽出 */
```

```
○tree( 整数型 : m_len, 整数型配列 : parent[], node[], m_cnt[] )
```

```
○整数型 : i, kosu
```

```
• kosu ← 0
```

```
• i ← 0
```

```
■ i < m_len
```

```
▲ parent[i] < 0      /* 親が作成されていないか? */
```

```
• node[kosu] ← i
```

```
• kosu ← kosu + 1
```

```
▼
```

```
• i ← i + 1
```

```
■
```

```
• n_sort( kosu, node, m_cnt )
```

```
• return kosu
```

```
/* 親が抽出されていない節に対する出現回数を昇順に整列 */
```

```
○n_sort( 整数型 : kosu, 整数型配列 : node[], m_cnt[] )
```

```
○整数型 : j, k, work
```

```
• k ← 0
```

```
■ k < kosu - 1
```

```
• j ← k + 1
```

```
■ j < kosu
```

```
▲ m_cnt[node[k]] > m_cnt[node[j]]
```

```
• work ← node[k]
```

```
■ (5)
```

```
▼
```

```
• j ← j + 1
```

```
■
```

```
• k ← k + 1
```

```
■
```

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

/* ビットの出力をする */

○out_bit(整数型 x, 整数型配列: parent[], left[])

↑ parent[x] ≥ 0

• out_bit(parent[x], parent, left)

↑ left[parent[x]] = x

• "0" を出力する

• "1" を出力する



(1) の解答群

ア. m_cnt[m] ← 0

イ. m_cnt[m] ← 1

ウ. m_cnt[m] ← m_cnt[m] - 1

エ. m_cnt[m] ← m_cnt[m] + 1

(2) の解答群

ア. k ← k - 1

イ. k ← k + 1

ウ. k ← k + m

エ. k ← m

(3) , (4) の解答群

ア. node[0]

イ. node[1]

ウ. node[kosu]

エ. node[m_len]

(5) の解答群

ア. • node[j] ← node[k]

イ. • node[k] ← node[j]

• node[k] ← work

• node[j] ← work

ウ. • node[k] ← work

エ. • node[j] ← work

• node[j] ← node[k]

• node[k] ← node[j]

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

< 選 択 問 題 >

選択問題は問題から1つ選択し解答せよ。

選択した問題は必ず、解答用紙「選択欄」にマークすること。

※選択欄にマークがなく、解答のみの場合は採点を行いません。

各構成は以下のとおり。

選択問題

- | | |
|------------|---------------|
| ・ C言語の問題 | 19 ページ～24 ページ |
| ・ 表計算の問題 | 25 ページ～32 ページ |
| ・ アセンブラの問題 | 33 ページ～35 ページ |

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

選択問題 C言語の問題

次のクイックソートに関する記述を読み、各設問に答えよ。

[クイックソートについて]

ソートアルゴリズムの中でも高速な処理が期待できるものであり、最善計算量および平均計算量は他のソートアルゴリズムと比べて最も高速である。ただし、データの並びによっては高速に並べ替えができない場合があり、その時の計算量は選択法ソートと同程度になる。なお、この問題で扱うデータは整数値であり、昇順に並べ替えるものとする。

[クイックソートの考え方]

クイックソートでは、配列内から基準値(以下、ピボット)を選択し、ピボットより小さい値と大きい値に分割する作業を、分割したそれぞれの配列領域で繰り返す。すべての分割が終了すれば各配列領域の要素数が1になり、並べ替えは完了する。

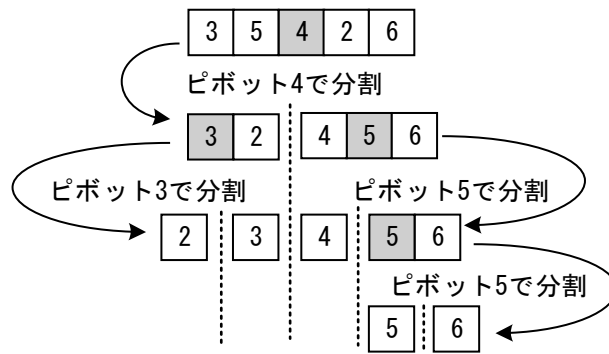


図1 クイックソートの流れ

[クイックソートの効率化]

クイックソートでは、分割した結果で左側または右側にだけ要素が片寄ってしまう場合に効率が悪いことがわかっている。最も効率が良いのは、データを順番に並べた時の中央にある値(以下、メジアン)をピボットにすることであるが、ここでは分割する配列領域の先頭、中央、末尾の3要素を順番に並べたときのメジアンをピボットにする。例えば、配列領域の先頭の値が3、中央の値が1、末尾の値が8であれば、ピボットは3である。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

<設問 1 > 次のプログラムの説明を読み、プログラム中の に入れるべき適切な字句を解答群から選べ。

[プログラムの説明]

引数で受け取った3つの整数値からメジアンを返却する関数 median である。

[関数の説明]

median 関数

引 数 : x, y, z (整数値)

機 能 : x, y, z の中からメジアンを返却する

戻り値 : メジアン

[プログラム]

```
int median(int x, int y, int z) {
    int max, min, res;
    /* x と y の大きい方を max, 小さい方を min に設定 */
    if (x > y) {
        max = x;
        min = y;
    } else {
        max = y;
        min = x;
    }
    /* z と max, min を比較してメジアンを res に設定する */
    if (z > max) {
         (1);
    } else if (z > min) {
         (2);
    } else {
         (3);
    }
    return res;
}
```

(1) ~ (3) の解答群

ア. res = max

イ. res = min

ウ. res = x

エ. res = y

オ. res = z

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

[クイックソートの手順]

1. 配列内で分割を行う領域の小さい要素位置を low, 大きい要素位置を high とする。
ただし, $low \geq high$ であれば戻る。
2. left に low, right に high を代入し, 分割する領域の先頭, 中央 (low と high を加えて 2 で割った位置), 末尾の位置にある値からメジアンを求めてピボットとする。
3. $left < right$ の間, 以下の操作を繰り返す。なお, $left \geq right$ となる場合は, 分割が完了した時である。

- ① left を 1 ずつ増やししながら, ピボット以上の値を検索する。

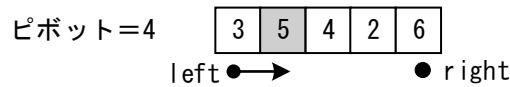


図 2 ピボット以上の値を検索

- ② right を 1 ずつ減らしながら, ピボット以下の値を検索する。

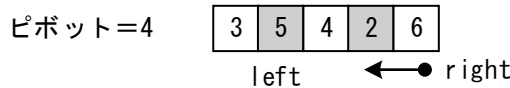


図 3 ピボット以下の値を検索

- ③ ①と②の結果, $left < right$ であれば配列の left と right の位置にある値を交換する。

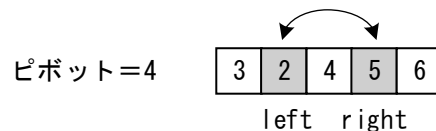


図 4 値の交換

4. 分割した左側の配列領域の要素数が 1 より大きければ, さらに分割するため再帰呼び出しを行う。この時の左端は low, 右端は left-1 である。
5. 分割した右側の配列領域の要素数が 1 より大きければ, さらに分割するため再帰呼び出しを行う。この時の左端は right+1, 右端は high である。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

<設問2> 次の図5の最初の配列から、クイックソートの手順で示した1~3を1回実行した後の配列の状態を解答群から選び(4)に答えよ。なお、ピボットは2とする。

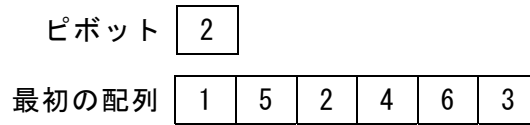


図5 クイックソート前の状態

(4)の解答群

ア.

1	2	3	4	6	5
---	---	---	---	---	---

イ.

1	2	5	4	6	3
---	---	---	---	---	---

ウ.

1	3	2	4	6	5
---	---	---	---	---	---

エ.

1	3	2	5	6	4
---	---	---	---	---	---

<設問3> 次のプログラムの説明を読み、プログラム中の

--

 に入れるべき適切な字句を解答群から選べ。

[プログラムの説明]

引数で受け取る配列をクイックソートにより並べ替える関数 `qsort` である。このプログラムでは、関数 `median` と配列のデータを交換するための関数 `swap` を利用する。関数 `median` は、設問1のプログラムである。

[関数の説明]

`qsort` 関数

引 数：`*ary`(データを格納した配列：整数型)，`low`(領域の左端：整数型)，`high`(領域の右端：整数型)

機 能：配列内をクイックソートにより並べ替える

戻り値：なし

`swap` 関数

引 数：`*a`(整数値)，`*b`(整数値)

機 能：`a` と `b` の内容を交換する

戻り値：なし

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

[プログラム]

```
void qsort(int *ary, int low, int high) {
    int i, m, left, right, pivot;
    if (low >= high) {
        return;
    } else {
        left = low;          /* 配列領域の左端を設定 */
        right = high;       /* 配列領域の右端を設定 */
        m = (low + high) / 2; /* 配列領域の中央を設定 */
        pivot = median(ary[left], ary[m], ary[right]);
        while(left < right) {
            /* ピボットより大きい値を左側から探す */
            while(ary[left] < pivot) {
                (5);
            }
            /* ピボットより小さい値を右側から探す */
            while(ary[right] > pivot) {
                (6);
            }
            if (left < right) {
                swap(&ary[left], &ary[right]); /* データを交換 */
            }
        }
        /* ピボットより小さい配列領域の並べ替え */
        if (low < left - 1) {
            qsort(ary, low, left - 1);
        }
        /* ピボットより大きい配列領域の並べ替え */
        if (high > right + 1) {
            qsort(ary, right + 1, high);
        }
    }
}

void swap(int *a, int *b) {
    int w;
    (7);
}
```

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

(5) , (6) の解答群

ア. `left--`

イ. `left++`

ウ. `right--`

エ. `right++`

(7) の解答群

ア. `w = a;`

イ. `w = &a;`

`a = b;`

`&a = &b;`

`b = w;`

`&b = w;`

ウ. `w = *a;`

エ. `*w = *a;`

`*a = *b;`

`*a = *b;`

`*b = w;`

`*b = *w;`

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

選択問題 表計算の問題

この問題で使用する表計算ソフトの仕様は下記のとおりである。

AND 関数

論理式のすべての評価が真であれば真、1つでも偽であれば偽を返す。

書式：AND(論理式 1, 論理式 2, …)

HLOOKUP 関数

検索値を検索範囲の上端列の中から検索し、見つかった列の指定した行位置の値を返す。行位置は1から始まる相対的な値であり、検索範囲中に見つけた列の中で、上から何番目の行かを示す。検索の型に0を指定すると検索値と完全に一致する値を検索し、1を指定すると検索値と一致する値がない場合に、検索値未満で一番大きい値を検索する。

書式：HLOOKUP(検索値, 範囲, 行位置, 検索の型)

IF 関数

条件が真のときに真の場合、偽のときに偽の場合の計算結果や値を返す。

書式：IF(条件, 真の場合, 偽の場合)

OR 関数

論理式の評価が1つでも真であれば真、すべて偽であれば偽を返す。

書式：OR(論理式 1, 論理式 2, …)

SUM 関数

範囲内の数値を合計した値を返す。

書式：SUM(範囲)

VLOOKUP 関数

検索値を検索範囲の左端列の中から検索し、見つかった行の指定した列位置の値を返す。列位置は1から始まる相対的な値であり、検索範囲中に見つけた行の中で、左から何番目の列かを示す。検索の型に0を指定すると検索値と完全に一致する値を検索し、1を指定すると検索値と一致する値がない場合に、検索値未満で一番大きい値を検索する。

書式：VLOOKUP(検索値, 検索範囲, 位置, 検索の型)

式

=に続いて計算式や関数などを入力する。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

セル番地の絶対参照

セル番地に \$ を付けることで、絶対番地(絶対参照)を表す。

別シートの参照

ワークシート名に「!」を付けてセル位置を指定することにより別シートを参照できる。

例：シート名「集計」のセル A1 を参照する場合は、「集計!A1」と記述する。

この問題で使用するマクロの仕様は下記のとおりである。

[マクロの記述]

マクロは一意のマクロ名を付けて宣言する。ワークシートには複数のマクロを格納できる。なお、マクロではワークシートと同様の関数を式内で利用することができる。

変数の型には、数値型、文字列型及び論理型があり、宣言することで使用できる。また変数名に続けて添字を「[」と「]」で囲むことで配列変数として宣言できる。添字が複数ある場合は「,」で区切る。添字は 0 から始まる。

[マクロの記述形式]

○マクロ：マクロ名	マクロを宣言する。
○変数型：変数名	変数を宣言する。
・変数 ← 式	変数に式の値を代入する。
/* 文 */	注釈を記述する。
▲ 条件式 ↑ ・処理 1 — ↓ ・処理 2 ▼	選択処理 条件式が真の時は処理 1 を実行し、偽の時は処理 2 を実行する。
▲ 条件式 ↑ ・処理 ↓ ▼	選択処理 条件式が真の時は処理を実行する。
■ 条件式 ・処理 ■	前判定繰り返し処理を示す。 条件式が真の間、処理を実行する。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

[式と演算子の優先順位]

演算の種類	演算子	優先順位
単項演算	+, -, not	高 ↑ ↓ 低
乗除演算	*, /, %	
加減演算	+, -	
関係演算	>, <, ≥, ≤, =, ≠	
論理積	and	
論理和	or	

[セルの参照]

マクロ中からセル番地を指定して、セルの値を参照することができる。セル番地を指定した絶対参照と、基点となるセル番地とそこからの相対的な位置を指定した相対参照がある。相対参照では、セル番地、行方向の相対位置、列方向の相対位置を「,」で区切って順に指定する。

セルの絶対参照と相対参照の例

- wk_a ← 絶対 (A1) /* A1 番地 */
- wk_b ← 絶対 (Sheet1!A1) /* Sheet1 の A1 番地 */
- wk_c ← 相対 (A1, 1, 2) /* C2 番地 */

(注) セル A1 から 1 行, 2 列移動したセル C2 が参照される。

J社では、表計算ソフトを使用して給与計算時の資格手当の計算をしている。社員数は30名で、「社員表」ワークシートの2行目からA列に社員番号、B列に氏名、C列に性別、D列に生年月日、E列に入社年月日、F列に所属ID、G列に役職名が31行まで入力されている。役職についていない社員の役職は空白になっている。日付を表しているセルにはシリアル値が格納されており、YYYY/M/Dの形式で表示している。

	A	B	C	D	E	F	G
1	社員番号	氏名	性別	生年月日	入社年月日	所属ID	役職名
2	10203	森脇 雅彦	男	1969/6/3	1991/4/1	E01	部長
3	11387	布施 早紀	女	1971/1/28	1993/10/1	J01	課長
4	12886	佐伯 三平	男	1973/7/17	1995/7/1	S01	
5	13128	矢吹 隆明	男	1975/9/24	1997/10/1	E01	課長
6	13438	高岡 千咲	女	1985/1/24	2007/4/1	S01	係長
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
30	21448	亀田 浩秋	男	1982/1/3	2004/4/1	E01	
31	22382	毛利 文隆	男	1968/8/31	1990/9/1	E01	

図1 「社員表」ワークシート

「資格表」ワークシートは、2行目からA列に資格名、B列に資格グレードが28行まで入力されている。また、E列～H列の1行目に資格グレード、2行目に1ヶ月ごとの資

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

格手当金額が入力されている。資格グレードはA～Dの4段階に区切られており、グレードにより1ヶ月の資格手当金額が設定されている。

	A	B	C	D	E	F	G	H
1	資格名	資格グレード		資格グレード	A	B	C	D
2	ITパスポート試験	D		資格手当金額	20,000	15,000	10,000	5,000
3	基本情報技術者試験	C						
4	応用情報技術者試験	B						
5	ネットワークスペシャリスト	A						
6	データベーススペシャリスト	A						
7	情報処理安全確保支援士	A						
⋮	⋮	⋮						
27	日商簿記検定(2級)	D						
28	社会保険労務士	A						

図2 「資格表」ワークシート

資格取得状況は図3のようになっている。「資格取得一覧」ワークシートは、2行目からA列に社員番号、B列に資格名、C列に資格取得日が90行まで入力されている。資格手当の計算では、複数の資格を取得している場合、資格手当が高額な方から2種類の合計を資格手当金とする。なお、処理対象は、各月の給与計算時まで申請登録済みの資格に限る。

	A	B	C
1	社員番号	資格名	資格取得日
2	11387	社会保険労務士	2013/8/14
3	12886	基本情報技術者試験	2014/10/17
4	15651	消防設備士(甲種:4類)	2015/5/13
5	16231	消防設備士(甲種:4類)	2015/5/13
6	16049	建築設備士	2015/5/14
⋮	⋮	⋮	⋮
89	21398	建築CAD検定(准1級)	2021/4/13
90	21448	データベーススペシャリスト	2021/4/19

図3 「資格取得一覧」ワークシート

<設問1> 次の「資格取得一覧」ワークシートの拡張に関する記述中の に入るべき適切な字句を解答群から選べ。

「資格取得一覧」ワークシートに、各資格に対する1ヶ月ごとの資格手当金額を追加する。なお、資格手当金額は、「資格表」ワークシートを検索して表示する。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

	A	B	C	D
1	社員番号	資格名	資格取得日	資格手当金額
2	11387	社会保険労務士	2013/8/14	20,000
3	12886	基本情報技術者試験	2014/10/17	10,000
4	15651	消防設備士(甲種:4類)	2015/5/13	10,000
5	16231	消防設備士(甲種:4類)	2015/5/13	10,000
6	16049	建築設備士	2015/5/14	10,000
⋮	⋮	⋮	⋮	⋮
89	21398	建築CAD検定(准1級)	2021/4/13	15,000
90	21448	データベーススペシャリスト	2021/4/19	20,000

図4 拡張した「資格取得一覧」ワークシート

資格手当金額を表示するため、セル D2 に次の式を入力し、セル D3~D90 まで複写した。

= (, 資格表!E\$1:H\$2, 2, 0)

(1), (2) の解答群

- ア. HLOOKUP
- イ. HLOOKUP(B2, 資格表!\$A2:\$B28, 2, 0)
- ウ. HLOOKUP(B2, 資格表!A\$2:B\$28, 2, 0)
- エ. VLOOKUP
- オ. VLOOKUP(B2, 資格表!\$A2:\$B28, 2, 0)
- カ. VLOOKUP(B2, 資格表!A\$2:B\$28, 2, 0)

<設問2> 次の「手当集計表」ワークシートの作成に関する記述中の に入れるべき適切な字句を解答群から選べ。

	A	B	C	D
1	社員番号	所属ID	役職名	資格手当合計
2	10203	E01	部長	10,000
3	11387	J01	課長	40,000
4	12886	S01		30,000
5	13128	E01	課長	30,000
6	13438	S01	係長	35,000
⋮	⋮	⋮	⋮	⋮
30	21448	E01		35,000
31	22382	E01		35,000

図5 「手当集計表」ワークシート

- ・ A~C 列の社員番号, 所属 ID, 役職名は, 「社員表」ワークシートから複写した。
- ・ D 列の資格手当金額はマクロ Salary_cal を作成して格納した。
マクロ Salary_cal は, 処理 I~IV を実行する。

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

[マクロ Salary_cal の説明]

資格手当の計算では、複数の資格を取得している場合、資格手当が高額な方から2種類を優先する。

I : セル A2 を最初の対象セルとして、以降、セル A3, A4, …と順次、対象セルを下に移しながら、対象セルの値が空値になるまで、II と III の処理を繰り返す。

II : 資格手当が高額なものを保存する二つの変数 shikaku_1st, shikaku_2nd に 0 を格納する。

III : 対象セルに入力されている社員番号と同じ値が入力されているかどうかを、「資格取得一覧」ワークシートのセル A2 から下の行方向に検索する。同じ社員番号が見つかったとき、当該行の列 D の資格手当金額と shikaku_1st, shikaku_2nd の中で、1 番高額のものを shikaku_1st に、2 番目に高額のものを shikaku_2nd に格納する。

IV : 当該行の列 D に shikaku_1st と shikaku_2nd の合計を格納する。

[マクロ : Salary_cal]

○マクロ : Salary_cal

○整数型 : i, j, shikaku_1st, shikaku_2nd

・ i ← 1

■ 相対(手当集計表!A1, i, 0) ≠ ""

・ shikaku_1st ← 0

・ shikaku_2nd ← 0

・ j ← 1

■ 相対(資格取得一覧!A1, j, 0) ≠ ""

(3)

相対(資格取得一覧!A1, j, 3) > (4)

・ shikaku_2nd ← shikaku_1st

・ shikaku_1st ← 相対(資格取得一覧!A1, j, 3)

相対(資格取得一覧!A1, j, 3) > (5)

・ shikaku_2nd ← 相対(資格取得一覧!A1, j, 3)

・ j ← j + 1

(6)

・ i ← i + 1

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

(3) の解答群

- ア. 相対(手当集計表!A1, i, 0) = 相対(資格取得一覧!A1, 1, 0)
- イ. 相対(手当集計表!A1, i, 0) = 相対(資格取得一覧!A1, i, 0)
- ウ. 相対(手当集計表!A1, i, 0) = 相対(資格取得一覧!A1, j, 0)
- エ. 相対(手当集計表!A1, j, 0) ≠ 相対(資格取得一覧!A1, j, 0)

(4) , (5) の解答群

- ア. shikaku_1st
- イ. shikaku_2nd
- ウ. 相対(資格取得一覧!A1, i, 0)
- エ. 相対(資格取得一覧!A1, j, 0)
- オ. 相対(手当集計表!A1, i, 3)
- カ. 相対(手当集計表!A1, j, 3)

(6) の解答群

- ア. 相対(手当集計表!A1, i, 3) ← shikaku_1st
- イ. 相対(手当集計表!A1, i, 3) ← shikaku_1st + shikaku_2nd
- ウ. 相対(手当集計表!A1, i, 3) ← shikaku_2nd
- エ. 相対(手当集計表!A1, i, j) ← shikaku_1st + shikaku_2nd

＜設問3＞ 次の「手当集計表」ワークシートの拡張に関する記述中の に入れるべき適切な字句を解答群から選べ。

「手当集計表」ワークシートに、役職報酬と手当合計を追加する。役職報酬は、「役職報酬表」ワークシートを検索して表示する。なお、課長以上は資格手当がなくなり役職報酬のみとなる。

	A	B	C	D	E	F
1	社員番号	所属ID	役職名	資格手当合計	役職報酬	手当合計
2	10203	E01	部長	10,000	80,000	80,000
3	11387	J01	課長	40,000	60,000	60,000
4	12886	S01		30,000		30,000
5	13128	E01	課長	30,000	60,000	60,000
6	13438	S01	係長	35,000	20,000	55,000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
30	21448	E01		35,000		35,000
31	22382	E01		35,000		35,000

図6 拡張した「手当集計表」ワークシート

	A	B
1	役職名	役職報酬
2	部長	80,000
3	課長	60,000
4	係長	20,000
5	主任	10,000

図7 「役職報酬表」ワークシート

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

・役職報酬を表示するため、セル E2 に次の式を入力し、セル E3～E31 まで複写した。
役職名が空白であれば、役職報酬も空白とする。 =

・手当合計を表示するため、セル F2 に次の式を入力し、セル F3～F31 まで複写した。
=

(7) の解答群

- ア. HLOOKUP(C2, 役職報酬表!A\$2:B\$5, 2, 0)
- イ. VLOOKUP(C2, 役職報酬表!A\$2:B\$5, 2, 0)
- ウ. IF(C2="", "", HLOOKUP(C2, 役職報酬表!A\$2:B\$5, 2, 0))
- エ. IF(C2="", "", VLOOKUP(C2, 役職報酬表!A\$2:B\$5, 2, 0))

(8) の解答群

- ア. IF(AND(C2 = "部長", C2 = "課長"), E2, SUM(D\$2:E\$2))
- イ. IF(C2 >= "課長", E2, SUM(D2:E2))
- ウ. IF(OR(C2 = "部長", C2 = "課長"), E2, SUM(D\$2:E\$2))
- エ. IF(OR(C2 = "部長", C2 = "課長"), E2, SUM(D2:E2))

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

選択問題 アセンブラの問題

次のアセンブラ言語CASL II プログラムの説明を読み、各設問に答えよ。

[プログラムの説明]

TBL 番地から始まる4語の連続した領域に、8ビットのデータが1語に2個ずつまとめて格納されている。この8個のデータに対してそれぞれパリティビットを求め、PARITY 番地の下位8ビットに格納するプログラムである

パリティビットは、対象データ中の1のビットの数が偶数個または奇数個になるように調整するためのビットである。パリティチェックは対象データとパリティビットを含めたビット列で、1のビット数の奇偶検査を行い誤りの有無を検査する方式である。偶数個で調整する場合を偶数パリティ方式、奇数個で調整する場合を奇数パリティ方式と呼ぶ。プログラムでは奇数パリティ方式を採用する。

また、プログラム中の内部サブルーチン CHECK は、GR0 の下位8ビットで受け取った1個のデータに対してパリティチェックを行い、パリティビットを GR3 の最下位ビットに設定する。

図1に TBL 番地以降のデータと PARITY 番地の関係を示す。なおパリティビットはすべてPで表現している。

TBL+0 番地	データ 1	データ 2
+1 番地	データ 3	データ 4
+2 番地	データ 5	データ 6
+3 番地	データ 7	データ 8

PARITY 番地	00000000	PPPPPPPP
-----------	----------	----------

↳ 左端がデータ1のパリティビット
右端がデータ8のパリティビット

図1 TBL 番地以降と PARITY 番地の関係

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

<設問 1 > 次のプログラム中の に入れるべき適切な字句を解答群から選べ。

[プログラム]

行番号	ラベル	命令	オペランド	コメント
100	MAIN	START		
110		LD	GR1,=0	
120		LD	GR6,=0	; PARITY 番地用の初期値
130	LOOP1	LD	GR0,TBL,GR1	
140		<input type="text"/>	(1)	; 上位 8 ビットの取り出し
150		CALL	CHECK	
160		SLL	GR6,1	
170		<input type="text"/>	(2)	; パリティビットの合成
180		LD	GR0,TBL,GR1	
190		<input type="text"/>	(3)	; 下位 8 ビットの取り出し
200		CALL	CHECK	
210		SLL	GR6,1	
220		<input type="text"/>	(2)	
230		LAD	GR1,1,GR1	
240		CPA	GR1,=4	
250		JMI	LOOP1	
260		ST	GR6,PARITY	
270		RET		
280	CHECK	LAD	GR3,0	
290		LAD	GR2,8	; 繰り返し回数の初期値
300	LOOP2	<input type="text"/>	(4)	; ビットを検査
310		JOV	BIT1	
320		JUMP	BIT0	
330	BIT1	LAD	GR3,1,GR3	; 1 のビットをカウント
340	BIT0	<input type="text"/>	(5)	
350		JNZ	LOOP2	
360		AND	GR3,=#0001	
370		XOR	GR3,=#0001	
380		RET		
390	TBL	DC	#0102	
400		DC	#0304	
410		DC	#0506	
420		DC	#0708	
430	PARITY	DS	1	
440		END		

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

(1) の解答群

ア. SLL GR0,1
ウ. SRL GR0,1

イ. SLL GR0,8
エ. SRL GR0,8

(2) の解答群

ア. OR GR3,GR1
ウ. OR GR6,GR1

イ. OR GR3,GR6
エ. OR GR6,GR3

(3) の解答群

ア. AND GR0,#00FF
ウ. AND GR6,#00FF

イ. AND GR0,#FF00
エ. AND GR6,#FF00

(4) の解答群

ア. AND GR0,#0001
ウ. SLL GR0,1

イ. OR GR0,#0001
エ. SRL GR0,1

(5) の解答群

ア. ADDA GR1, =1
ウ. SUBA GR2, =1

イ. ADDA GR2, =1
エ. SUBA GR1, =1

<設問2> TBL番地以降に図2の値を設定したとき、PARITY番地の下位8ビットの値を解答群から選べ。

TBL+0番地	00000001	00000010
+1番地	00000011	00000100
+2番地	00000101	00000110
+3番地	00000111	00001000

図2 TBL番地以降の値

(6) の解答群

ア. 00010010
ウ. 11000010

イ. 00101100
エ. 11110000

<設問3> 1命令を削除することで奇数パリティ方式を偶数パリティ方式に変更できる。削除する命令の行番号を解答群から選べ。

(7) の解答群

ア. 300

イ. 330

ウ. 360

エ. 370

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

<メモ欄>

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf

<メモ欄>

令和6年(2024年)4月から選択問題(問題5)に改訂があります。

改訂の詳細は以下URLをご覧ください。

https://jken.sgec.or.jp/common/pdf/information/jken_r6_kaitei_jsystem_programming.pdf