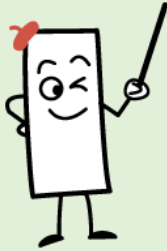


第 11 回 全国高等学校 情報処理選手権 プログラミング部門 問題



各部門の出題の雰囲気の確認や、学習方法アドバイスに書かれている学習方法とあわせて、大会のための学習にご利用ください。

※各大会過去問題の解答解説は、大会エントリー時に送られる確認メールからダウンロードいただけます。

※プログラミング部門では問23のような長文問題を1題出題します。メモ用紙をご準備いただくことを推奨します。

※全国高等学校情報処理選手権の学習以外での目的の無断転載・複製・配布はご遠慮ください。

株式会社アイテック

問1

次の 10 進小数のうち、8 進数に変換したときに有限小数になるものはどれか。

811881

- ア 0.65
- イ 0.7
- ウ 0.75
- エ 0.8

問2

次の論理演算が成立するときに、a に入るビット列はどれか。ここで、 \oplus は排他的論理和を表す。

821774

- ① ② ③ ④

$$1101 \oplus 1011 \oplus \boxed{a} \oplus 1100 = 1111$$

- ア 0011
- イ 0101
- ウ 1001
- エ 1011

問3

桁落ちの説明として、適切なものはどれか。

(H27 春-FE 問 2)

ア 値がほぼ等しい浮動小数点数同士の減算において、有効桁数が大幅に減ってしまうことである。

イ 演算結果が、扱える数値の最大値を超えることによって生じるエラーのことである。

ウ 浮動小数点数の演算結果について、最小の桁よりも小さい部分の四捨五入、切上げ又は切捨てを行うことによって生じる誤差のことである。

エ 浮動小数点数の加算において、一方の数値の下位の桁が結果に反映されないことである。

問4

ある工場では、同じ製品を独立した二つのライン A, B で製造している。ライン A では製品全体の 60%を製造し、ライン B では 40%を製造している。ライン A で製造された製品の 2%が不良品であり、ライン B で製造された製品の 1%が不良品であることが分かっている。いま、この工場で製造された製品の一つを無作為に抽出して調べたところ、それは不良品であった。その製品がライン A で製造された確率は何%か。

(H28 秋-FE 問 2)

ア 40

イ 50

ウ 60

エ 75

問5

四つの整数を引数とする関数 $d(X_1, Y_1, X_2, Y_2)$ を次のとおりに定義する。

$$d(X_1, Y_1, X_2, Y_2) = |X_1 - X_2| + |Y_1 - Y_2|$$

この関数は、2点 (X_1, Y_1) と (X_2, Y_2) との間の2次元正方格子上の最短経路長を求め
るものである。 $d(0, 0, X_2, Y_2) \leq 1$ を満たす整数の組 (X_2, Y_2) は、全部で何組か。

714549

- ア 3
- イ 4
- ウ 5
- エ 6

問6

逆ポーランド表記法(後置表記法)で表現した式、 $AB \div C + DE - \times$ がある。 $A=10, B=2, C=1, D=9, E=7$ のとき、この式の値はどれか。ここで、割り算は小数点以下を切り捨てるものとする。

729449

- ア 8
- イ 12
- ウ 16
- エ 32

問7

次の規則から生成することができる式はどれか。

811543

[規則]

$\langle \text{式} \rangle ::= \langle \text{変数} \rangle \mid (\langle \text{式} \rangle - \langle \text{式} \rangle) \mid \langle \text{式} \rangle \div \langle \text{式} \rangle$

$\langle \text{変数} \rangle ::= A \mid B \mid C \mid D$

- ア $A - (B \div C) - D$
- イ $(A - B) \div (C \div D)$
- ウ $(A \div B) - (C - D)$
- エ $((A - B) - C) \div D$

問8

次のような双方向リストにおいて，“パリ”を“ロンドン”と“ウィーン”の間に挿入することが出来る処理はどれか。

なお，指し示すデータがないとき，ポインタには0が入っている。

713359

アドレス	データ部分	次ポインタ	前ポインタ
100	ウィーン	160	180
120	トウキョウ	180	0
140	パリ	0	0
160	ミラノ	0	100
180	ロンドン	100	120

ア パリの次ポインタを100，ウィーンの前ポインタを140，ウィーンの次ポインタは160のまま，ロンドンの次ポインタを140とする。

イ パリの前ポインタを180，ロンドンの前ポインタは120のまま，ロンドンの次ポインタを140，ウィーンの前ポインタを140とする。

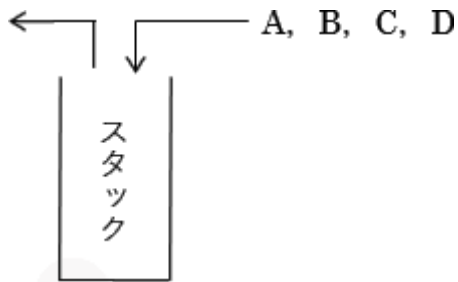
ウ ロンドンの次ポインタを140，パリの前ポインタを180，パリの次ポインタを100，ウィーンの前ポインタを140とする。

エ ロンドンの前ポインタを140，パリの前ポインタを100，パリの次ポインタを180，ウィーンの次ポインタを140とする。

問9

A, B, C, D の順序で入力されるデータがある。各データについてスタックへの挿入と取出しを1回ずつ行うことができる場合、C が先頭にくるデータの出力順序は何通りあるか。

820255

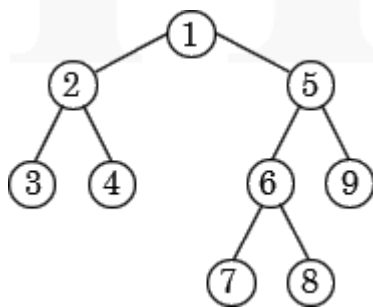


- ア 3
- イ 4
- ウ 5
- エ 6

問10

節に1~9の値がそれぞれ格納されている次の2分木を、通りかけ順(中間順)で探索する。このときの探索順に節の値を並べたものはどれか。

822078



- ア 123456789
- イ 125346978
- ウ 324176859
- エ 342786951

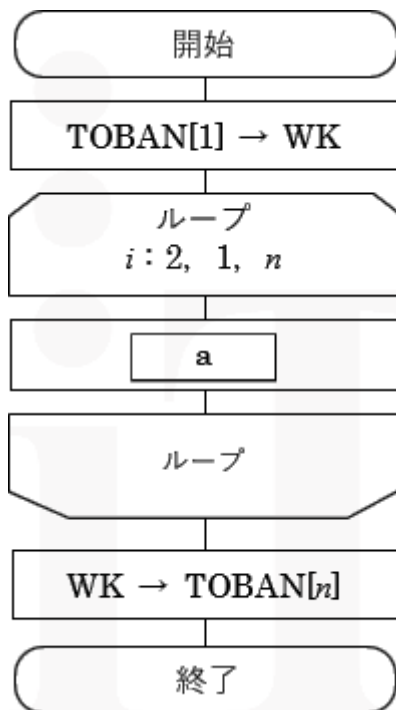
問11

要素番号が1から始まる配列 TOBAN がある。 n 人の名前が TOBAN[1]から TOBAN[n]に入っている。

図は、日直登板が終わった配列1番目の人の名前を TOBAN[n]に移動するために、TOBAN[2]から TOBAN[n]の名前を順に一つずつ前にずらして当番表を再構成する流れ図である。

に入れる処理として、適切なものはどれか。
ここで変数 WK は移動用の作業エリアとする。

811649



(注) ループにおける条件は、
変数名：初期値，増分，
終値を示す。

- ア TOBAN[i] → TOBAN[$i-1$]
- イ TOBAN[$i-1$] → TOBAN[i]
- ウ TOBAN[$i+1$] → TOBAN[$n-i$]
- エ TOBAN[$n-i$] → TOBAN[$n-i-1$]

問12

探索アルゴリズムとその実行時間のオーダーの適切な組合せはどれか。ここで、探索するデータの数を n とし、ハッシュ値が衝突する(同じ値になる)確率は無視できるほど小さいものとする。また、実行時間のオーダーが n^2 であるとは、 n 個のデータを処理する時間が cn^2 (c は定数) で抑えられることをいう。

811647

	線形探索	2分探索	ハッシュ探索
ア	1	n^2	n
イ	n	$\log_2 n$	1
ウ	n	$\log_2 n$	n
エ	n^2	$n \log_2 n$	1

問13

整列アルゴリズムの一つであるクイックソートの記述として、適切なものはどれか。

811648

- ア 整列対象データから最小値を求め、次にそれを除いた部分の中から最小値を求める。この操作を繰り返していき、整列を行う。
- イ 対象集合から要素を順次取り出し、それまでに取り出した要素の集合に順序関係を保つよう挿入して、整列を行う。
- ウ 中間的な基準値を決めて、これよりも大きい要素の集合と小さい要素の集合に分割する。この操作を繰り返すことで、整列を行う。
- エ 隣り合う要素を比較して、大小の順が逆であれば、それらの要素を入れ替えるという操作を繰り返し、整列を行う。

問14

次の関数 $\text{gcm}(m, n)$ の定義に従って $\text{gcm}(140, 54)$ を求めるとき、 $\text{gcm}(m, n)$ は何回呼ばれるか。ここで、最初の $\text{gcm}(140, 54)$ の呼出しも、1 回に数えるものとする。また、 $m, n (m > n \geq 0)$ は整数とし、 $m \bmod n$ は m を n で割った余りを返すものとする。

822327

```
 $\text{gcm}(m, n) = \text{if } n = 0 \text{ then } m$   
 $\qquad \qquad \qquad \text{else } \text{gcm}(n, m \bmod n)$ 
```

- ア 5
- イ 6
- ウ 7
- エ 8

問15

構造化プログラミングを説明したものはどれか。

729961

- ア 入口は一つと決められているが、出口は処理によっては複数になってもよい。
- イ 基本的には、“順次・繰返し・接続”の三つの構造で表現する。
- ウ 構造化プログラミングは、モジュールの物理設計の要点である。
- エ 強い制約条件はあるが、保守しやすいプログラムを作ることができる。

問16

動的テストツールに関する記述のうち、デバッガに関するものはどれか。

812554

- ア テスト対象モジュールに必要なドライバ、又はスタブを生成する。
- イ テストによって実行した経路から網羅度を算出する。
- ウ プログラムの特定の経路をテストするためのデータを生成する。
- エ モジュールの実行を一時停止したり、メモリ内容を書き出したりする。

問17

オブジェクト指向において、スーパークラスで定義されたデータ属性や手続をサブクラスが引き継いで使うことができる特性はどれか。

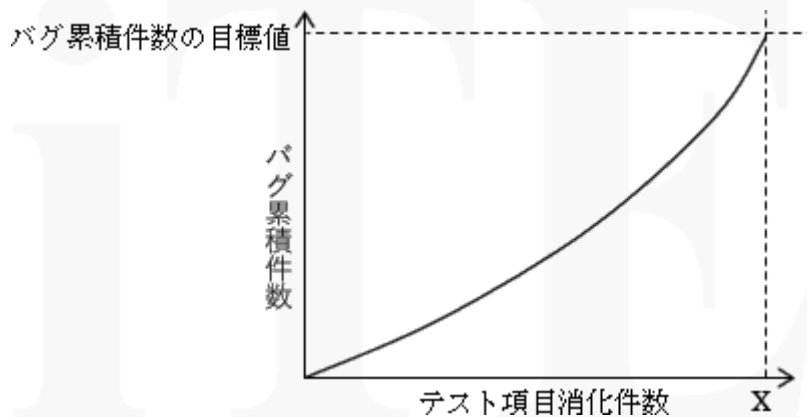
713717

- ア インスタンス
- イ インヘリタンス
- ウ デリゲーション
- エ ポリモーフィズム

問18

図は、テスト項目消化件数 X において、目標値として設定したバグ累積件数に到達したことを示す。この図の状況の説明として、適切なものはどれか。

(H27 春-FE 問 49)

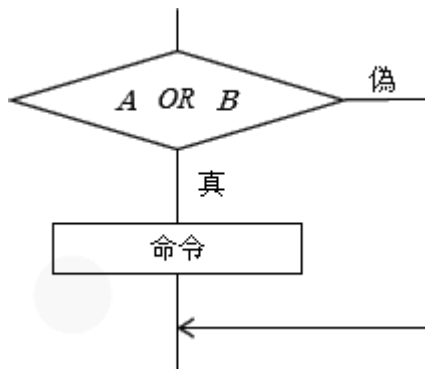


- ア テスト工程が順調に終了したことを示す。
- イ テスト前段階での机上チェックやシミュレーションが十分にされていることを示す。
- ウ まだ多くのバグが内在している可能性があることを示す。
- エ 目標のバグ累積件数が達成されたので、出荷後にバグが発生する確率が低いことを示す。

問19

プログラム中の図の部分を実定条件網羅(分岐網羅)でテストするときのテストケースとして、適切なものはどれか。

(H27 秋-FE 問 47)



ア

A	B
偽	真

イ

A	B
偽	真
真	偽

ウ

A	B
偽	偽
真	真

エ

A	B
偽	真
真	偽
真	真

問20

ホワイトボックステストに関する記述として、最も適切なものはどれか。

714435

- ア 下位のモジュールの代替となるスタブを利用してテストを実施する。
- イ 正常データと誤りデータのクラスを設定して、それぞれのクラスからの代表データをテストデータとする。
- ウ 正しい入力データの範囲の境界値をテストデータとする。
- エ プログラムの判定条件を網羅するテストケースを設定する。

問21

プログラムの著作権に関する記述として、適切なものはどれか。

713770

- ア 請負契約でプログラムを開発した場合、プログラムの著作権は委託側に帰属する。
- イ 購入したプログラムの複製については、いかなる場合でも認められない。
- ウ 社員が会社の業務としてプログラムを開発した場合、プログラムの著作権は開発した社員に帰属する。
- エ プログラムを他の特定のコンピュータで動作させるために移植する場合には、プログラムの改変が認められている。

問22

不正競争防止法によって保護される対象として規定されているものはどれか。

(H28 春-SG 問 35)

- ア 自然法則を利用した技術的思想の創作のうち高度なものであって、プログラム等を含む物と物を生産する方法
- イ 著作物を翻訳し、編曲し、若しくは変形し、又は脚色し、映画化し、その他翻案することによって創作した著作物
- ウ 秘密として管理されている事業活動に有用な技術上又は営業上の情報であって、公然と知られていないもの
- エ 法人等の発意に基づきその法人等の業務に従事する者が職務上作成するプログラム著作物

問23

次のプログラムの説明及びプログラムを読んで、設問 1～3 に答えよ。

811429

プログラム 1 及びプログラム 2 は、いずれも 4 桁の数当てゲームのプログラムである。プログラムが生成した正解となる 4 桁の数字(以下、目標数という)を解答者が推測し、当たるまで推測した数字(以下、推測数という)を入力するゲームである。まず、一つ目のプログラム 1 について考える。

[プログラム 1 のゲーム仕様]

目標数は任意の 4 桁の数字である。解答者が入力した推測数と目標数を各桁ごとに比較して、数字が同じであれば当たりとする。当たりの数が一つ以上あれば、「n 個当たり!」と表示し、一つも当たりにない場合は、「残念 もう一度チャレンジ」と表示する。次の推測数は、はずれた桁の数字だけを入力し、全ての桁が当たるまで続ける。

目標数が“4552”のときの画面表示のおおまかな流れ

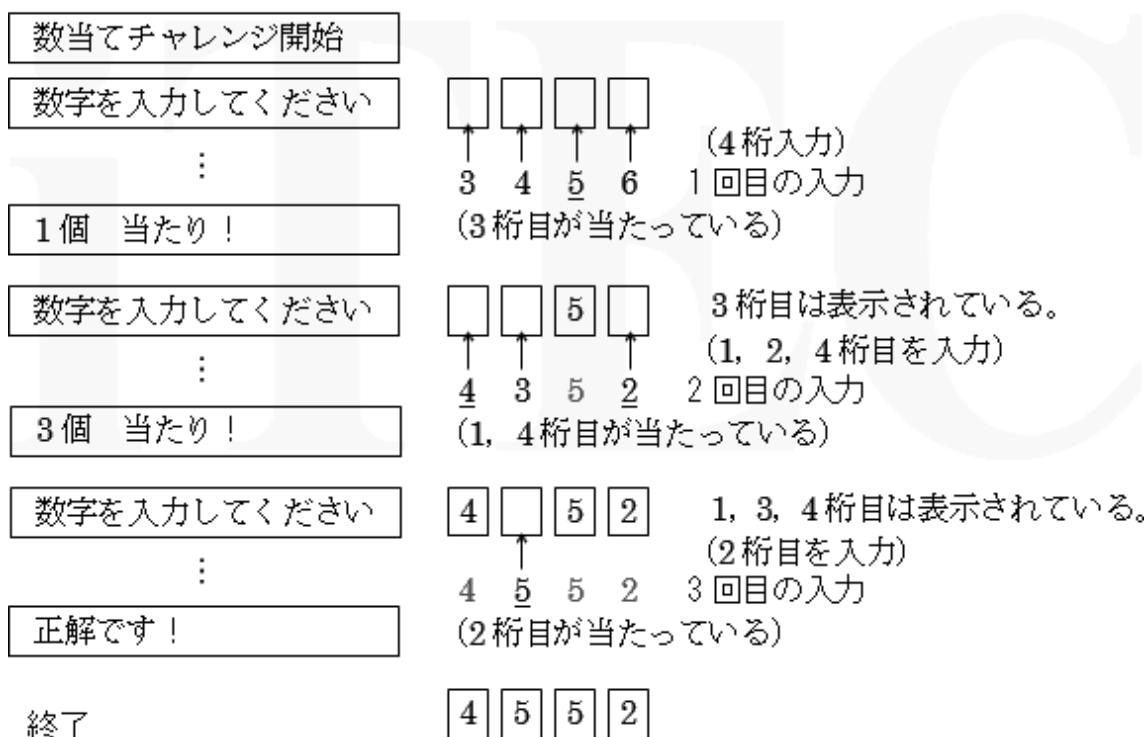


図 1 プログラム 1 のゲーム仕様例

〔プログラム 1 の説明〕

(1) 目標数は関数 `createrandom` で自動的に生成する。生成した数字列は、左の桁から順に文字型配列 `target` の各要素に格納されている。例えば、目標数が“4552”ならば、`target[1] = '4'`、`target[2] = '5'`、`target[3] = '5'`、`target[4] = '2'`と格納される。推測数は解答者が入力し、文字型配列 `num` の指定された要素に格納される。

(2) 目標数と推測数が一致しているかどうかは、`target` と `num` の同じ位置(添字)の内容が一致しているかどうかで判定する。一致していれば、その位置の入力はしなくてよく、一致していない位置について再度入力をうながす。

(3) 次の関数が用意されている。

・関数 `createrandom`(文字型配列)

引数:文字型配列 `target`

機能:4桁の目標数を生成し、文字列として文字型配列 `target` の各要素に格納して返す。

・関数 `inputnum`(文字型変数)

引数:`num` の 1 要素

機能:あらかじめ画面に `num[1]` から `num[4]` を表示する場所が決められており、引数の位置をスペースクリアしてから入力をうながす。数字以外は受け付けない。入力した数字を `num` の指定された要素に設定して返す。

(4) 画面にメッセージを表示する処理は、「」内にメッセージを記述して次のように表示する。

(例)・「正解です！」と表示

画面には、 と表示される。また、メッセージ内にその時点での変数の値を表示する場合は、`[]` でくくった変数名を記述する。

・「`[hitcounter]`個 当たり！」と表示

`hitcounter` が 2 であれば、画面には、 と表示される。

図 2 プログラム 1 の画面メッセージ表示処理の例

[プログラム 1]

```
○プログラム1      /* 数当てゲームのプログラム1 */
○整数型: hitcounter, i, j, comp, p
○整数型: composit[4]
○文字型: target[4], num[4]
/* 初期設定 */
・hitcounter ← 0      /* 当たった数字の個数 */
■ i: 1, i ≤ 4, 1
  ・composit[i] ← i      /* 比較位置の初期値 */
  ■
/* 目標数の生成 */
・createrandom(target)      /* target にランダムな数字を設定 */
/* 推測数入力と判定 */
・「数当てチャレンジ開始」と表示
■ 
  ・comp ← 4 - hitcounter      /* 比較数を設定 */
  ・「数字を入力してください」と表示
  ■ i: 1, , 1
    ・inputnum(num[composit[i]]) /* num の該当位置に数字を入力 */
    ■
/* 目標数と入力した推測数の比較 */
  ・「判定中」と表示
  ・j ← 1
  ■ i: 1, , 1
    ・p ← composit[i]      /* 比較位置を設定 */
    
    ↑
    ・hitcounter ← hitcounter + 1
    ↓
    ・
    ・j ← j + 1
  ■
  ▲ hitcounter = 0
  ・「残念 もう一度チャレンジ」と表示
  ▲ hitcounter ≠ 4
  ▲
  ▲ ・「[hitcounter]個 当たり！」と表示
  ↓
  ■
  ・「正解です！」と表示
  ・return      /* プログラム1の終了 */
```

問23 - 設問 1

プログラム中の に入れる正しい答えを、解答群の中から選べ。

a に関する解答群

- ア hitcounter < 4
- イ hitcounter ≤ 4
- ウ i < 4
- エ i ≤ 4
- オ i < comp
- カ i ≤ comp

b に関する解答群

- ア hitcounter < 4
- イ hitcounter ≤ 4
- ウ i < 4
- エ i ≤ 4
- オ i < comp
- カ i ≤ comp

c に関する解答群

- ア num[i] = target[i]
- イ num[i] = target[j]
- ウ num[i] = target[p]
- エ num[p] = target[p]

d に関する解答群

- ア composit[i] ← p
- イ composit[j] ← p
- ウ composit[p] ← p
- エ p ← composit[i]
- オ p ← composit[j]
- カ p ← composit[p]

次に、もう一つのプログラムであるプログラム 2 について考える。

[プログラム 2 のゲーム仕様]

目標数は各桁の値が全て異なる 4 桁の数字である。解答者が入力する際に、何回目の入力か分かるようにチャレンジ回数を表示する。

入力した推測数が当たりでないときには、目標数に含まれていて桁も一致している数字の個数(以下、ヒット数という)と、目標数に含まれているが桁は一致していない数字の個数(以下、ブロー数という)を表示する。解答者は、この情報を元に、全ての桁の数字が当たるまで推測数の入力続ける。

目標数が“1632”のときの画面表示のおおまかな流れ

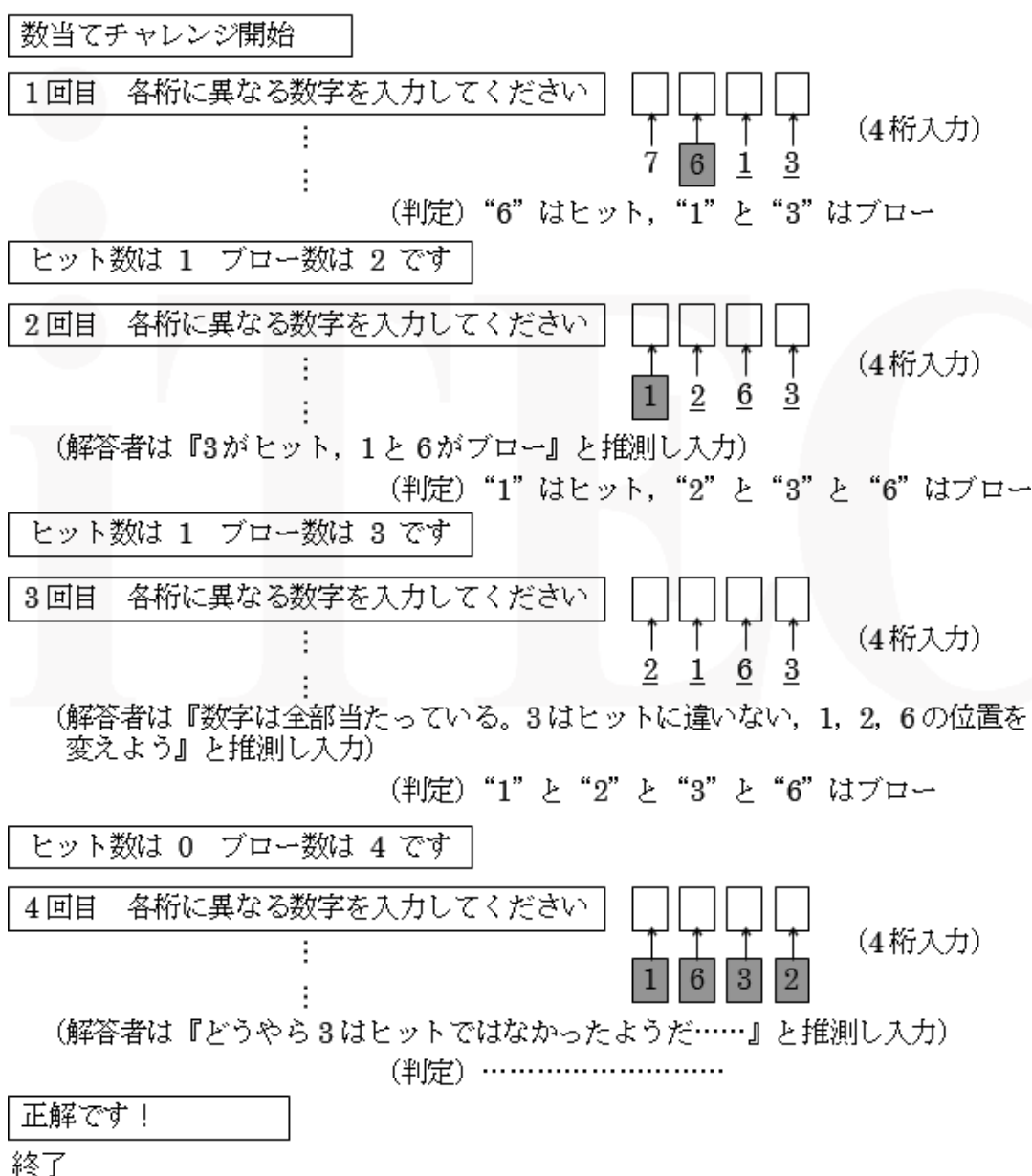


図 3 プログラム 2 のゲーム仕様例

〔プログラム 2 の説明〕

(1) 目標数は関数 createnum で全て異なる 4 桁の数字を自動的に生成する。生成した数字列は、文字列として左の桁から順に文字型配列 target の各要素に格納されている。推測数は解答者が入力し、左の桁から順に文字型配列 num の指定された要素に格納される。

(2) 目標数と推測数が一致しているかどうかの判定は、整数型関数 Match で行う。ヒット数とブロー数を求めるため、target と num の各要素を総当たりで一致しているかどうか比較して調べる。ヒット数が 4 で正解であれば 1 を返す。正解でなければヒット数とブロー数を表示して 0 を返す。

(3) 画面にメッセージを表示する処理と数字入力の関数 inputnum は、プログラム 1 と同じである。ほかには、次の関数が用意されている。

・関数 createnum(文字型配列)

引数:文字型配列 target

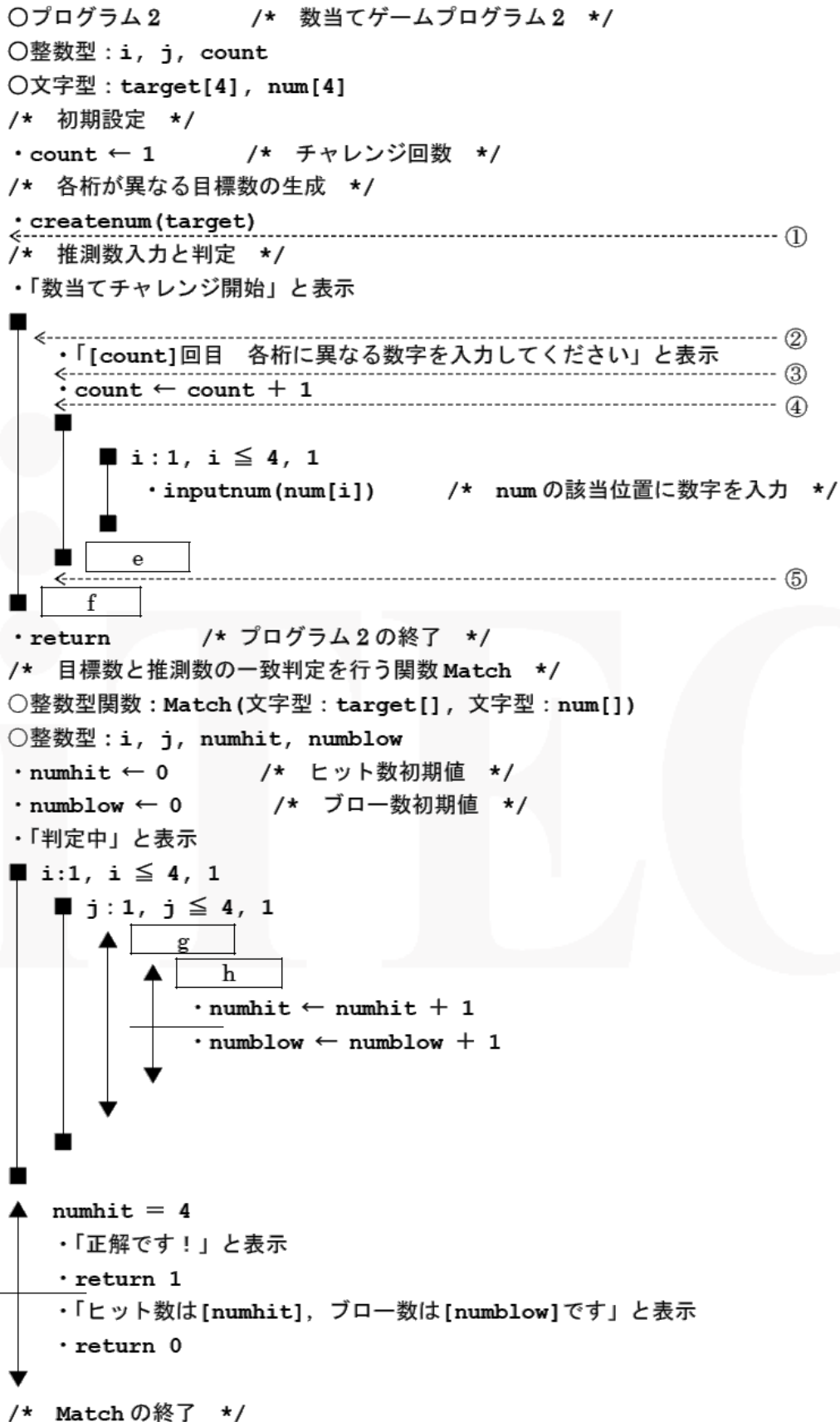
機能:各桁の数字が全て異なる 4 桁の目標数を、文字列として左の桁から順に文字型配列 target の各要素に格納して返す。

・整数型関数 validnum(文字型配列)

引数:文字型配列 num

機能:文字型配列 num の各要素が互いに異なる数字かどうかを判定する。正しくない場合は、「入力が正しくありません。再度入力してください」と表示し 0 を返す。正しければ 1 を返す。

[プログラム 2]



問23 - 設問2

プログラム中の に入れる正しい答えを、解答群の中から選べ。

e に関する解答群

- ア $\text{Match}(\text{target}, \text{num}) = 0$
- イ $\text{Match}(\text{target}, \text{num}) = 1$
- ウ $\text{validnum}(\text{num}) = 0$
- エ $\text{validnum}(\text{num}) = 1$

f に関する解答群

- ア $\text{Match}(\text{target}, \text{num}) = 0$
- イ $\text{Match}(\text{target}, \text{num}) = 1$
- ウ $\text{validnum}(\text{num}) = 0$
- エ $\text{validnum}(\text{num}) = 1$

g に関する解答群

- ア $\text{num}[i] = \text{num}[j]$
- イ $\text{target}[i] = \text{num}[i]$
- ウ $\text{target}[i] = \text{num}[j]$
- エ $\text{target}[i] = \text{target}[j]$
- オ $\text{target}[j] = \text{num}[j]$

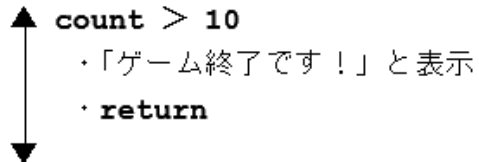
h に関する解答群

- ア $i \neq j$
- イ $i < j$
- ウ $i \leq j$
- エ $i = j$
- オ $i > j$
- カ $i \geq j$

問23 - 設問3

次の記述中の に入れる正しい答えを、解答群の中から選べ。

プログラム 2 において、推測数を 10 回入力しても正解できなかった場合にゲームを終了するよう仕様変更したい。そのためには、次の命令群を i に挿入すればよい。



iに関する解答群

- ア ①
- イ ②
- ウ ③
- エ ④
- オ ⑤