情報工学

予想問題　第4回　解答

（理論・変換系の重量級年）

記述式の小問については解答例を示す。同じ内容が伝われば表現は問わない。

作成：Claude（Anthropic）

1 【必須問題】アルゴリズムとプログラミング

(1-1) (a) `C[v] += C[v-1];` (b) `C[digit(A[i], p)]` (c) `p *= 10;`

(a) について：8 行目終了時点で `C[v]` は「着目桁の値がちょうど v である要素の個数」である。これを累積和に変換することで，`C[v]` は「着目桁の値が v 以下である要素の個数」となる。

(b) について：11 行目で `C[...]` を 1 減じた後の値が，その要素を格納すべき `B` の添字となる。

(1-2) 各反復終了時点の配列 `A` の内容は以下のとおり。

r	着目桁	反復終了時の `A`
0	1 の位	720, 355, 436, 457, 657, 329, 839
1	10 の位	720, 329, 436, 839, 355, 457, 657
2	100 の位	329, 355, 436, 457, 657, 720, 839

出力は以下のとおり。

```
329 355 436 457 657 720 839
```

(2-1) $r = 0$（1 の位）のとき，各要素の 1 の位は $329 \to 9$，$457 \to 7$，$657 \to 7$，$839 \to 9$，$436 \to 6$，$720 \to 0$，$355 \to 5$ である。

8 行目のループ終了時（各値の出現回数）：

$$C = [1, 0, 0, 0, 0, 1, 1, 2, 0, 2]$$

（添字 0 から 9 の順。値 0 が 1 個，5 が 1 個，6 が 1 個，7 が 2 個，9 が 2 個。）

9 行目のループ終了時（累積和）：

$$C = [1, 1, 1, 1, 1, 2, 3, 5, 5, 7]$$

(2-2) 累積和をとった後の `C[d]` は「着目桁の値が d 以下である要素の個数」，すなわち桁の値が d である要素群が `B` 中で占める区間の**末尾の次の位置**を指す。11 行目で `C[d]` を 1 減じてから格納するので，同じ桁の値をもつ要素は後ろの位置から順に埋められていく。

したがって入力配列を後ろから走査すれば，入力で後方にあった要素ほど `B` の後方に置かれ，同じ桁の値をもつ要素どうしの相対的な順序が保存される。これが安定性である。逆に前から走査すると同じ桁の値をもつ要素の順序が反転してしまい，安定でなくなる。

(2-3) 例として 2 要素 $\{21, 25\}$ を考える。

1 の位で整列すると $21, 25$ の順になる（$1 < 5$）。次に 10 の位で整列するが，両者とも 10 の位は 2 で等しい。ここで安定でないソートを用いると，順序が入れ替わって $25, 21$ となる可能性がある。この時点で処理は終了するので，最終結果が $25, 21$ という誤った順序になる。

基数ソートは「下位の桁での整列結果を，上位の桁が等しい要素の間で保存する」ことによって正しさが保証されている。安定性が失われるとこの前提が崩れ，アルゴリズム全体が正しく動作しなくなる。

(3-1) 時間計算量は $\mathbf{O(d(n+k))}$，空間計算量は $\mathbf{O(n+k)}$。

時間計算量の根拠：1 回の `counting_sort` の呼び出しでは，計数のループが $O(n)$，累積和のループが $O(k)$，配置のループが $O(n)$ であり，合わせて $O(n+k)$ である。21 行目の書き戻しも $O(n)$ である。これを桁数 d 回繰り返すので，全体で $O(d(n+k))$ となる。

空間計算量の根拠：補助的に使用するのは出力用の配列 `B`（大きさ n）と計数用の配列 `C`（大きさ k）のみであり，桁ごとの繰り返しで使い回されるので $O(n+k)$ である。

(3-2) $\Omega(n \log n)$ の下界は，「要素どうしの**比較**のみによって順序を決定する」という計算モデルに対して導かれたものである。このモデルでは，$n!$ 通りの順列を区別するために決定木の葉が $n!$ 個必要であり，木の高さが $\log_2(n!) = \Omega(n \log n)$ となることから下界が従う。

一方，基数ソートは要素どうしの比較を一切行わず，桁の値そのものを配列 `C` や `B` の添字として直接利用している。すなわち比較モデルの外側にあるアルゴリズムであるため，この下界の制約を受けず，d と k が小さければ $O(n \log n)$ より高速になり得る。

(3-3) 各要素を基数 n の数として表現する。$0 \le x \le n^3 - 1$ であるから，基数 n での桁数は

$$d = \lceil \log_n(n^3) \rceil = \mathbf{3}$$

であり，n に依存しない定数となる。各桁がとり得る値の種類数は $k = n$ である。

したがって全体の時間計算量は

$$O\big(d(n+k)\big) = O\big(3(n+n)\big) = \mathbf{O(n)}$$

となり，線形時間で整列できる。（これは比較ソートの下界 $\Omega(n \log n)$ を下回るが，小問 (6) のとおり比較モデルの外にあるため矛盾しない。）

2 【必須問題】計算機システムとシステムプログラム

(1-1) -6.5 を 2 進数で表すと -110.1_2 である。正規化すると

$$-110.1_2 = -1.101_2 \times 2^2.$$

したがって符号部 $s = 1$，指数部は $e - 127 = 2$ より $e = 129 = 10000001_2$，仮数部は小数部 101 に 0 を補って 10100000000000000000000 となる。

並べると

$$\underbrace{1}_{s}\ \underbrace{10000001}_{e}\ \underbrace{10100000000000000000000}_{f}$$

であり，4 ビットずつ区切って

$$1100\ 0000\ 1101\ 0000\ 0000\ 0000\ 0000\ 0000_2 = \texttt{0xC0D00000}.$$

(1-2) 最小の正規化数は $e = 1$（すなわち指数 $1 - 127 = -126$），仮数部 $f = 0$ のときであるから

$$1.0 \times 2^{-126} = \mathbf{2^{-126}} \approx 1.18 \times 10^{-38}.$$

最大の数は $e = 254$（指数 $254 - 127 = 127$），仮数部がすべて 1 のときであるから

$$\left(2 - 2^{-23}\right) \times 2^{127} = (\mathbf{2} - \mathbf{2^{-23}}) \cdot \mathbf{2^{127}} \approx 3.40 \times 10^{38}.$$

（$e = 0$ と $e = 255$ はそれぞれ非正規化数・無限大や NaN のために予約されている。）

(1-3) 1.0 は指数部 $e = 127$，仮数部 $f = 0$ で表される。仮数部は 23 ビットであるから，1.0 の次に大きい表現可能な数は $1 + 2^{-23}$ である。最近接偶数丸め (round to nearest even) のもとでは，1.0 と $1 + 2^{-23}$ のちょうど中間である $1 + 2^{-24}$ は仮数部が偶数となる 1.0 に丸められる。したがって求める最大の数は

$$\mathbf{2^{-24}} \approx 5.96 \times 10^{-8}.$$

引き起こされる問題：絶対値の大きく異なる二つの数を加算すると，小さい方の数の下位の桁が丸めによって失われる。これを情報落ちという。例えば大きな数に微小な数を繰り返し加算していく計算では，各回の加算が結果に反映されず，本来得られるべき値から大きくずれてしまう。総和を求める際には絶対値の小さいものから順に加算するなどの配慮が必要となる。

(1-4) 理由：2 の補数表現では 32 ビットすべてを数値の大きさに使うため表現できる範囲は -2^{31} 以上 $2^{31} - 1$ 以下に限られる。一方，浮動小数点数は 8 ビットを指数部に割り当てており，指数によって値のスケールを 2^{-126} から 2^{127} まで変化させられるため，はるかに広い範囲を表せる。

失われるもの：仮数部が 23 ビットに限られるため**有効数字（精度）**が失われる。整数表現では範囲内のすべての整数を誤差なく表せるのに対し，浮動小数点数では 2^{24} を超える整数は正確に表現できず，隣り合う表現可能な数の間隔が値の大きさに応じて広がる。

(2-1) ページサイズが $4\,\mathrm{KiB} = 2^{12}$ バイトであるからページ内オフセットは **12 ビット**。仮想アドレスは 32 ビットであるから仮想ページ番号は $32 - 12 =$ **20 ビット**。物理メモリは $1\,\mathrm{GiB} = 2^{30}$ バイトであり物理アドレスは 30 ビットであるから，物理フレーム番号は $30 - 12 =$ **18 ビット**。

(2-2) エントリ数は仮想ページ番号のとり得る値の個数に等しく 2^{20} である。1 エントリ 4 バイトであるから

$$2^{20} \times 4\ \text{バイト} = \mathbf{4\ MiB}.$$

次に 2 段ページテーブルについて。 外側ページテーブルのエントリ数は，仮想ページ番号の上位 10 ビットのとり得る値の個数であるから $2^{10} = \mathbf{1024}$ 個。

内側ページテーブル 1 個あたりの大きさは，エントリ数 $2^{10} = 1024$，1 エントリ 4 バイトであるから

$$1024 \times 4 = \mathbf{4\ KiB}\ (= \text{ちょうど 1 ページ分}).$$

節約できる理由：2 段方式では外側ページテーブル（4 KiB）は常に必要だが，内側ページテーブルは実際に使用している仮想アドレス領域に対応するものだけを割り当てればよい。未使用領域については外側エントリの有効ビットを 0 としておけば内側テーブルを用意する必要がない。プロセスが空間のごく一部しか使わない場合，必要な内側テーブルは数個で済むため，1 段方式で常に 4 MiB を確保するのに比べて大幅に節約できる。

(2-3) `0x00403008` を 2 進数で表すと

$$0000\ 0000\ 0100\ 0000\ 0011\ 0000\ 0000\ 1000$$

である。上位 10 ビット，次の 10 ビット，下位 12 ビットに分割すると

	2 進数	16 進数
外側インデックス（上位 10 ビット）	0000000001	`0x001`
内側インデックス（次の 10 ビット）	0000000011	`0x003`
オフセット（下位 12 ビット）	000000001000	`0x008`

表 1 より外側インデックス 1 は有効（$V = 1$）で内側テーブル PT-B を指す。表 2 より PT-B の索引 3 は有効でフレーム番号 `0x003A0` である。

物理アドレスはフレーム番号を 12 ビット左シフトしてオフセットを加えたものであるから

$$\texttt{0x003A0} \times \texttt{0x1000} + \texttt{0x008} = \texttt{0x3A0000} + \texttt{0x008} = \texttt{0x003A0008}.$$

(2-4) TLB ヒット時は $20 + 100 = 120\,\text{ns}$，TLB ミス時は $20 + 100 + 100 = 220\,\text{ns}$ を要する。よって

$$\text{EAT} = 0.98 \times 120 + 0.02 \times 220 = 117.6 + 4.4 = \mathbf{122}\ [\mathbf{ns}].$$

2 段ページテーブルの場合： 2 段ページテーブルでは TLB ミス時にページテーブルの参照が 2 回必要であるから，ミス時は $20 + 100 + 100 + 100 = 320\,\text{ns}$ を要する。よって

$$\text{EAT} = 0.98 \times 120 + 0.02 \times 320 = 117.6 + 6.4 = \mathbf{124}\ [\mathbf{ns}].$$

トレードオフ：2 段方式はページテーブルが占める記憶領域を大幅に削減できる一方で，TLB ミス時のアドレス変換に要するメモリ参照回数が増えるため，実効アクセス時間が増加する（本問では $122 \to 124\,\text{ns}$）。すなわち**記憶領域の節約と変換速度のトレードオフ**である。TLB ヒット率が高いほどこの増加分は小さく抑えられる。

【補足】ページフォルト率との関係: ページフォルトが起きない場合は 100 ns，起きる場合は $10\,\text{ms} = 10^7\,\text{ns}$ の処理に加えてメモリアクセスを要する。ページフォルト率を p とすると

$$\text{EAT} = (1 - p) \times 100 + p \times \left(10^7 + 100\right) = 100 + p \times 10^7\ \ [\text{ns}].$$

これが 200 ns 以下となる条件は

$$100 + 10^7 p \le 200 \quad \Longleftrightarrow \quad p \le \frac{100}{10^7} = \mathbf{1.0 \times 10^{-5}}.$$

スラッシングが致命的である理由：上の式が示すとおり，ページフォルト 1 回の処理時間はメモリアクセスの約 10 万倍であり，実効アクセス時間はページフォルト率に極めて敏感である。多重プログラミング度を上げすぎると各プロセスに割り当てられるページ枠が不足し，必要なページを追い出しては即座に読み込み直す状態に陥ってページフォルト率が急上昇する。その結果 CPU はほとんどの時間を入出力の待ちに費やすことになり，実行できる処理量が激減する。

4 【選択問題】計算理論

(1) $\Gamma = \{Z, A, B\}$ とし，入力の a に対して A を，b に対して B をスタックに積む。区切り記号 c を読んだ時点で照合の状態 q_1 へ移り，以降は入力とスタックトップを照合しながら取り去る。スタックが開始記号 Z のみになったら受理状態 q_2 へ移る。必要な遷移は以下のとおり。

q_0 **における自己ループ（積み込み）：**	$(a,Z)/AZ$, $(a,A)/AA$, $(a,B)/AB$ $(b,Z)/BZ$, $(b,A)/BA$, $(b,B)/BB$
$q_0 \to q_1$ **（区切り記号を読む）：**	$(c,Z)/Z$, $(c,A)/A$, $(c,B)/B$
q_1 **における自己ループ（照合）：**	$(a,A)/\varepsilon$, $(b,B)/\varepsilon$
$q_1 \to q_2$ **（受理）：**	$(\varepsilon,Z)/Z$

各状態・各入力記号・各スタックトップの組合せに対して適用可能な遷移は高々一つであり，ε 遷移をもつのは q_1 でスタックトップが Z のときのみで，このとき入力を読む遷移（トップが A または B の場合）とは競合しない。したがってこのプッシュダウンオートマトンは決定性である。

(2) L_1 では区切り記号 c が入力中に明示されているため，c を読んだ瞬間に「積み込みから照合への切り替え点」が一意に定まる。したがって機械は決定的に動作できる。

一方 L_2 には切り替え点，すなわち w と w^R の境界（文字列の中央）を示す情報が入力中に存在しない。機械は入力を読み進めながら「ここが中央である」と推測しなければならないが，決定性プッシュダウンオートマトンは推測（複数の可能性の並行探索）ができない。中央の位置を誤ると照合が破綻し，やり直すこともできないため，本質的に非決定性が必要となる。

(3) M' を以下のように構成する。

- 状態集合：$Q \cup \{q_s, q_f\}$
- スタックアルファベット：$\Gamma \cup \{Z_0\}$（スタックの開始記号は Z_0）
- 最終状態の集合：$\{q_f\}$
- 開始状態：q_s
- 追加する遷移：
  - $\delta'(q_s, \varepsilon, Z_0) = (q_0,\ ZZ_0)$ …… M の開始記号 Z を Z_0 の上に積んでから M を起動する。
  - すべての $q \in Q$ に対して $\delta'(q, \varepsilon, Z_0) = (q_f,\ Z_0)$ …… Z_0 が露出したら受理状態へ移る。

同じ言語を受理する理由：M' は最初に Z_0 の上に M の初期スタック内容を積むので，M の動作中は Z_0 がスタックの底に隠れており，M の動作に影響を与えない。M がスタックを空にする（M の記号をすべて取り去る）のは，M がその入力を受理するとき，かつそのときに限る。このとき M' のスタックトップには Z_0 が現れるので，追加した ε 遷移により q_f へ移り，最終状態による受理が成立する。逆に M がスタックを空にしなければ Z_0 は露出せず q_f に到達できない。よって両者が受理する言語は一致する。

(4-1) $L = \{a^n b^n \mid n \geq 1\}$ が語頭独立でないと仮定する。すなわち，$u = a^i b^i \in L$ が $w = a^j b^j \in L$ の真の接頭辞であるような $i, j \geq 1$ が存在するとする。

真の接頭辞であるから $|u| < |w|$，すなわち $2i < 2j$ より $j > i$ である。$i \geq 1$ であるから $j \geq i+1$ が成り立つ。

ここで両者の $i+1$ 文字目を比較する。

- $u = a^i b^i$ の $i+1$ 文字目：先頭から i 文字が a，続く i 文字が b であり，$i \geq 1$ より $i+1 \leq 2i$ であるから，$i+1$ 文字目は **b** である。
- $w = a^j b^j$ の $i+1$ 文字目：先頭から j 文字が a であり，$j \geq i+1$ より $i+1 \leq j$ であるから，$i+1$ 文字目は **a** である。

u は w の接頭辞であるから，両者の $i+1$ 文字目は一致しなければならない。しかし一方は b，他方は a であり矛盾する。

したがって仮定が誤りであり，$L = \{a^n b^n \mid n \geq 1\}$ は語頭独立である。☒

(4-2) 空欄の答えは以下のとおり。

空欄	答え
(ア)	入力 u を読む計算における動作
(イ)	空
(ウ)	いかなる遷移も実行することができない

補足：(ア) について，M が決定性であることから各時点で適用可能な遷移は高々一つに定まる。uv の先頭 $|u|$ 記号は u そのものであるから，そこまでの動作（状態の列とスタックの内容の変化）は u を単独で読む場合と完全に一致する。

(ウ) について，遷移関数はスタックの先頭にある記号 $s \in \Gamma$ を取り去ることを要求するため，スタックが空の状態では入力を読む遷移も ε 遷移も適用できず，計算はそこで停止する。

(5-1) 状態を一つ（q とする）だけ持ち，スタックの開始記号を文法の開始記号 S とする。必要な遷移は以下の 2 種類である。

- **変数の展開**：各生成規則 $X \to \alpha$ に対して

$$(\varepsilon,\ X)/\alpha$$

すなわち入力を読まずに，スタックトップの変数 X を取り去り，規則の右辺 α を（右端の記号から順に）積む。

- **終端記号の照合**：各終端記号 $a \in T$ に対して

$$(a,\ a)/\varepsilon$$

すなわち入力から a を読み，スタックトップの a を取り去る（何も積まない）。

スタックには「まだ生成されていない部分」が保持され，変数の展開は常に最左の変数に対して行われるので，この動作は最左導出を模倣している。導出が完了して入力を読み終えたときスタックは空になるので，空スタックによる受理で $L(G)$ が認識される。

(5-2) 主張は**偽**である。

理由：小問 (4) より，空スタックによる受理を行う決定性プッシュダウンオートマトンで認識される言語は必ず語頭独立である。ところが $\{a^n b^n \mid n \geq 1\}$ は文脈自由言語であるが，たとえば ab と $aabb$ を含み，ab は $aabb$ の真の接頭辞ではないので……と一見すると語頭独立に見える。しかしより単純な例として，文脈自由言語 $\{a^n \mid n \geq 1\}$（これは正則言語でもあり文脈自由言語でもある）は，小問 (4-1) で示したとおり語頭

独立でない。したがってこの言語は空スタック受理の決定性プッシュダウンオートマトンでは認識できない。よって「任意の文脈自由言語が認識できる」という主張は成り立たない。

6 【選択問題】電子回路と論理設計

(1-1) pop は SP を減じるだけでメモリの内容を消さないことに注意する。

操作	SP	Mem[0]	Mem[1]	Mem[2]
リセット直後	0000	—	—	—
push 3	0001	3	—	—
push 7	0010	3	7	—
push 2	0011	3	7	2
pop	0010	3	7	2
pop	0001	3	7	2
push 5	0010	3	**5**	2

最後の `push 5` では，そのときの SP の値 0001 が指す 1 番地に 5 が書き込まれ（7 が上書きされ），SP は 0010 となる。直後に `pop` を実行すると SP は 0001 となり，次のサイクルで `r_dat` には 1 番地の内容すなわち **5** が現れる。

(1-2) スタックが空となるのは SP が 0 のとき，満杯となるのは SP が $8 = 1000_2$ のときである。よって

$$E = \overline{s_3}\,\overline{s_2}\,\overline{s_1}\,\overline{s_0}, \qquad F = s_3\,\overline{s_2}\,\overline{s_1}\,\overline{s_0}.$$

(1-3) 8 語すべてが使用中の状態を表すには SP が値 8 をとる必要がある。SP が 3 ビットであれば値の範囲は 0〜7 であり，8 は 000 に折り返してしまう。すると「空（SP = 0）」と「満杯（SP = 8）」が同じ符号となり，E と F を区別できなくなる。0 から 8 までの 9 通りを表現するには 4 ビットが必要である。

(1-4) 本問のデータパスが実現しているのは**スタック**（LIFO：後入れ先出し）である。

2026 年度の「カウンタ 2 個とメモリ」からなる構成が実現するのは**キュー**（FIFO：先入れ先出し）であり，格納した順にデータが取り出される。これに対しスタックは最後に格納したデータから先に取り出される点が異なる。

(2-1) 状態遷移図は以下のとおり（出力は各状態に付随する Moore 型）。

現状態	遷移条件 → 次状態	出力
S_{IDLE}	`req_push`= 1 かつ $F = 0 \to S_{\mathrm{PUSH}}$ `req_push`= 1 かつ $F = 1 \to S_{\mathrm{ERR}}$ `req_push`= 0，`req_pop`= 1，$E = 0 \to S_{\mathrm{POP}}$ `req_push`= 0，`req_pop`= 1，$E = 1 \to S_{\mathrm{ERR}}$ 上記以外 → S_{IDLE}	すべて 0
S_{PUSH}	無条件 → S_{IDLE}	`w_en`= 1, `up`= 1
S_{POP}	無条件 → S_{IDLE}	`down`= 1
S_{ERR}	無条件 → S_{IDLE}	`err`= 1

(2-2) 状態割り当ては $S_{\mathrm{IDLE}} = (0,0)$，$S_{\mathrm{PUSH}} = (0,1)$，$S_{\mathrm{POP}} = (1,0)$，$S_{\mathrm{ERR}} = (1,1)$ である。S_{IDLE} 以外の三つの状態からは無条件に $S_{\mathrm{IDLE}} = (0,0)$ へ戻るので，$D_1 = D_0 = 0$ となる。したがって次状態が非零となるのは現状態が S_{IDLE}，すなわち $\overline{y_1}\,\overline{y_0}$ のときに限られる。

S_{IDLE} における遷移先を整理すると，

- $S_{\mathrm{PUSH}} = (0,1)$ へ行くのは $\texttt{req\_push} \cdot \overline{F}$ のとき
- $S_{\mathrm{POP}} = (1,0)$ へ行くのは $\overline{\texttt{req\_push}} \cdot \texttt{req\_pop} \cdot \overline{E}$ のとき
- $S_{\mathrm{ERR}} = (1,1)$ へ行くのは $\texttt{req\_push} \cdot F$ または $\overline{\texttt{req\_push}} \cdot \texttt{req\_pop} \cdot E$ のとき

D_1 は次状態の上位ビットが 1 となる場合（S_{POP} または S_{ERR}），D_0 は下位ビットが 1 となる場合（S_{PUSH} または S_{ERR}）であるから

$$D_1 = \overline{y_1}\,\overline{y_0}\Big(\texttt{req\_push} \cdot F + \overline{\texttt{req\_push}} \cdot \texttt{req\_pop}\Big),$$

$$D_0 = \overline{y_1}\,\overline{y_0}\Big(\texttt{req\_push} + \overline{\texttt{req\_push}} \cdot \texttt{req\_pop} \cdot E\Big),$$

$$\texttt{err} = y_1 y_0.$$

（D_1 について：S_{POP} の条件 $\overline{\texttt{req\_push}}\,\texttt{req\_pop}\,\overline{E}$ と S_{ERR} の条件の一部 $\overline{\texttt{req\_push}}\,\texttt{req\_pop}\,E$ は E について相補的なので，まとめて $\overline{\texttt{req\_push}} \cdot \texttt{req\_pop}$ となる。D_0 についても同様に，S_{PUSH} の条件 $\texttt{req\_push}\,\overline{F}$ と S_{ERR} の条件の一部 $\texttt{req\_push}\,F$ がまとまって $\texttt{req\_push}$ となる。）

(3-1)　S_{PUSH}，S_{POP}，S_{ERR} から無条件に S_{IDLE} へ戻る代わりに，これらの状態においても $\texttt{req\_push}$ と $\texttt{req\_pop}$ を評価し，S_{IDLE} と同じ遷移条件で直接次の操作状態へ遷移できるようにすればよい。すなわち S_{IDLE} における遷移論理を S_{PUSH}，S_{POP}，S_{ERR} の各状態にも適用し，要求がない場合にのみ S_{IDLE} へ戻る構成とする。これにより連続する要求に対して 1 クロックサイクルあたり 1 操作を処理できる。

（なお，この改造では次状態の論理式から $\overline{y_1}\,\overline{y_0}$ の因子が外れ，D_1, D_0 は現状態によらず要求信号と E, F のみで決まる形になる。）

(3-2)　破壊されるのは **0 番地**の内容である。

理由：$F = 1$ すなわち SP の値が $8 = 1000_2$ のときに push 操作を行うと，メモリのアドレス入力 $\texttt{adr}$ には SP の**下位 3 ビット**が与えられる。1000_2 の下位 3 ビットは $000_2 = 0$ であるから，スタックの底である 0 番地が指定され，そこに書き込みが行われて既存の内容が上書きされてしまう。同時に SP は $9 = 1001_2$ となり，スタックの整合性が失われる。

7　【選択問題】数学解析と信号処理

(1-1)　$z^2 + 1 = (z - i)(z + i)$ であるから，特異点は $z = i$ と $z = -i$ であり，分母の零点はいずれも単純であるから**ともに 1 位の極**である。留数は

$$\mathrm{Res}_{z=i} f_1(z) = \lim_{z \to i} (z - i) \cdot \frac{1}{(z - i)(z + i)} = \frac{1}{2i}, \qquad \mathrm{Res}_{z=-i} f_1(z) = \frac{1}{-2i}.$$

(1-2)　$f_2(z) = \dfrac{z}{(z-1)^2(z+2)}$ の特異点は $z = 1$ と $z = -2$ であり，$z = 1$ は **2 位の極**，$z = -2$ は **1 位の極**である。

$z = -2$（1 位）：

$$\mathrm{Res}_{z=-2} f_2(z) = \lim_{z \to -2} (z + 2) \cdot \frac{z}{(z-1)^2(z+2)} = \frac{-2}{(-3)^2} = -\frac{2}{9}.$$

$z = 1$（2 位）：公式に $m = 2$ を適用して

$$\mathrm{Res}_{z=1} f_2(z) = \frac{1}{1!} \lim_{z \to 1} \frac{d}{dz} \left[(z-1)^2 \cdot \frac{z}{(z-1)^2(z+2)} \right] = \lim_{z \to 1} \frac{d}{dz} \left[\frac{z}{z+2} \right].$$

ここで

$$\frac{d}{dz} \left[\frac{z}{z+2} \right] = \frac{(z+2) \cdot 1 - z \cdot 1}{(z+2)^2} = \frac{2}{(z+2)^2}$$

であるから，$z=1$ を代入して

$$\mathrm{Res}_{z=1} f_2(z) = \frac{2}{9}.$$

(2-1) 積分路 $|z-i|=1$ は中心 i，半径 1 の円である。$z=-i$ との距離は $|-i-i|=2>1$ であるから，内部の特異点は $z=i$ のみである。留数定理より

$$\oint_{|z-i|=1} \frac{dz}{z^2+1} = 2\pi i \cdot \frac{1}{2i} = \boldsymbol{\pi}.$$

(1-3) $|z|=2$ の内部には $z=i$ と $z=-i$ の両方が含まれるから

$$\oint_{|z|=2} \frac{dz}{z^2+1} = 2\pi i \left(\frac{1}{2i} + \frac{1}{-2i} \right) = \mathbf{0}.$$

(1-4) $|z|=3$ の内部には $z=1$ と $z=-2$ の両方が含まれるから，小問 (1-2) の結果を用いて

$$\oint_{|z|=3} \frac{z\,dz}{(z-1)^2(z+2)} = 2\pi i \left(\frac{2}{9} - \frac{2}{9} \right) = \mathbf{0}.$$

(1-5) C_R 上では $|z|=R$ である。三角不等式より

$$|z^2+1| \geq |z|^2 - 1 = R^2 - 1 > 0 \qquad (R>1)$$

であるから，被積分関数の絶対値は

$$\left| \frac{1}{z^2+1} \right| \leq \frac{1}{R^2-1}$$

で上から抑えられる。半円 C_R の長さは πR であるから

$$\left| \int_{C_R} \frac{dz}{z^2+1} \right| \leq \frac{1}{R^2-1} \cdot \pi R = \frac{\pi R}{R^2-1}.$$

右辺は分子が R の 1 次，分母が 2 次であるから $R \to \infty$ で 0 に収束する。したがって $\displaystyle\int_{C_R} \frac{dz}{z^2+1} \to 0$ である。

(1-6) 閉曲線 C（実軸上の $[-R,R]$ と C_R）について，$R>1$ のとき内部の特異点は $z=i$ のみであるから留数定理より

$$\oint_C \frac{dz}{z^2+1} = 2\pi i \cdot \frac{1}{2i} = \pi.$$

一方

$$\oint_C \frac{dz}{z^2+1} = \int_{-R}^{R} \frac{dx}{x^2+1} + \int_{C_R} \frac{dz}{z^2+1}$$

であり，小問 (3-1) より第 2 項は $R \to \infty$ で 0 に収束する。よって

$$\int_{-\infty}^{\infty} \frac{dx}{x^2+1} = \boldsymbol{\pi}.$$

原始関数による検算：$\displaystyle\int \frac{dx}{x^2+1} = \arctan x$ であるから

$$\int_{-\infty}^{\infty} \frac{dx}{x^2+1} = [\arctan x]_{-\infty}^{\infty} = \frac{\pi}{2} - \left(-\frac{\pi}{2}\right) = \pi$$

となり一致する。

(1-7) 被積分関数を複素関数として $g(z) = \dfrac{1}{(z^2+1)^2} = \dfrac{1}{(z-i)^2(z+i)^2}$ とみると，上半平面にある極は $z=i$ であり，分母に $(z-i)^2$ の因子をもつので **2 位の極**である。

小問 (1-2) と同様に $m=2$ の公式を適用すると

$$\mathrm{Res}_{z=i} g(z) = \frac{1}{1!}\lim_{z\to i}\frac{d}{dz}\left[(z-i)^2\cdot\frac{1}{(z-i)^2(z+i)^2}\right] = \lim_{z\to i}\frac{d}{dz}\left[\frac{1}{(z+i)^2}\right].$$

ここで

$$\frac{d}{dz}\left[(z+i)^{-2}\right] = -2(z+i)^{-3}$$

であるから，$z=i$ を代入して

$$\mathrm{Res}_{z=i} g(z) = \frac{-2}{(2i)^3} = \frac{-2}{-8i} = \frac{1}{4i}.$$

半円上の積分の評価：C_R 上では $|z|=R$ であり

$$|(z^2+1)^2| = |z^2+1|^2 \geq \left(R^2-1\right)^2$$

であるから

$$\left|\int_{C_R}\frac{dz}{(z^2+1)^2}\right| \leq \frac{\pi R}{(R^2-1)^2} \xrightarrow{R\to\infty} 0.$$

（分子が R の 1 次，分母が 4 次であるため。）

したがって留数定理より

$$\int_{-\infty}^{\infty}\frac{dx}{(x^2+1)^2} = 2\pi i\cdot\frac{1}{4i} = \boldsymbol{\frac{\pi}{2}}.$$

(2-1) 部分分数分解を

$$\frac{1}{(s+1)(s^2+4)} = \frac{A}{s+1} + \frac{Bs+C}{s^2+4}$$

とおく。両辺に $(s+1)(s^2+4)$ を掛けて

$$1 = A(s^2+4) + (Bs+C)(s+1).$$

$s=-1$ を代入して $1=5A$ より $A=\frac{1}{5}$。s^2 の係数比較より $0=A+B$ すなわち $B=-\frac{1}{5}$。定数項の比較より $1=4A+C$ すなわち $C=\frac{1}{5}$。

よって

$$F(s) = \frac{1}{5}\cdot\frac{1}{s+1} - \frac{1}{5}\cdot\frac{s}{s^2+4} + \frac{1}{10}\cdot\frac{2}{s^2+4}$$

であり，逆ラプラス変換して

$$f(t) = \frac{1}{5}e^{-t} - \frac{1}{5}\cos 2t + \frac{1}{10}\sin 2t \qquad (t\geq 0).$$

(2-2) 最終値の定理を形式的に適用すると

$$\lim_{s\to 0} sF(s) = \lim_{s\to 0}\frac{s}{(s+1)(s^2+4)} = \frac{0}{1\cdot 4} = \mathbf{0}.$$

一方，小問 (2-1) で求めた $f(t)$ において $t\to\infty$ とすると，第 1 項 $\frac{1}{5}e^{-t}$ は 0 に収束するが，第 2 項と第 3 項は $\cos 2t$，$\sin 2t$ を含むため**振動し続けて収束しない**。

したがって両者は**一致しない**。理由は，最終値の定理が適用できるのは $sF(s)$ のすべての極が複素平面の左半平面にある場合に限られるからである。本問の $F(s)$ は $s=\pm 2i$ という虚軸上の極をもつためこの条件を満たさず，定理を適用してはならない。形式的に計算して得られる値 0 には意味がない。

(2-3) 差分方程式 $y[n] = x[n] + a\,y[n-1]$ の両辺を z 変換すると，時間シフトの性質より $\mathcal{Z}[y[n-1]] = z^{-1}Y(z)$ であるから

$$Y(z) = X(z) + az^{-1}Y(z) \quad\Longrightarrow\quad \left(1-az^{-1}\right)Y(z) = X(z).$$

よって

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}.$$

インパルス応答は $|z| > |a|$ で $\dfrac{1}{1 - az^{-1}} = \sum_{n=0}^{\infty} a^n z^{-n}$ と展開できるので

$$h[n] = a^n u[n] \qquad (u[n] \text{ は単位ステップ列}).$$

(2-4) $H(z) = \dfrac{z}{z - a}$ の極は $z = a$ である。因果的なシステムが BIBO 安定であるための条件は，すべての極が単位円の内側にあること，すなわち

$$|a| < 1.$$

このとき

$$\sum_{n=0}^{\infty} |h[n]| = \sum_{n=0}^{\infty} |a|^n = \frac{1}{1 - |a|} < \infty$$

となり，インパルス応答が絶対総和可能であることが確認できる。（絶対総和可能性は BIBO 安定性と同値である。）

(2-5) $\dfrac{z}{z - a}$ を a で偏微分すると

$$\frac{\partial}{\partial a}\left(\frac{z}{z - a}\right) = \frac{z}{(z - a)^2}$$

となり，これは与えられた $H(z)$ に一致する。一方，小問 (2-3) より $\dfrac{z}{z - a}$ の逆 z 変換は $a^n u[n]$ であるから，同じ操作を時間領域で行うと

$$\frac{\partial}{\partial a}\left(a^n\right) = n\, a^{n-1}.$$

z 変換は a に関する微分と交換できるので

$$h[n] = n\, a^{n-1} u[n].$$

（検算：$n = 0$ で $h[0] = 0$，$n = 1$ で $h[1] = 1$ となる。実際 $\frac{z}{(z-a)^2} = z^{-1} + 2az^{-2} + 3a^2 z^{-3} + \cdots$ と展開され一致する。なお $z = a$ は 2 位の極であり，小問 (1-2) で扱った 2 位の極をもつ複素関数と同じ構造である。）