

2023年度 秋学期中間試験				問題枚数	1/1	
科目名	出題者氏名	受験クラス	学生証番号	氏名		
データ構造とアルゴリズム	山本 宙 佐藤 未来子 星野 祐子	JE-1, 2, 3, その他				
持込	不可	◇可の場合は、記入	開講曜日・時限	現在使用している授業教室	12-305 コンピュータ室 12-307A, B コンピュータ室 12-309 コンピュータ室	採点
	可		月曜 3, 4 限			

注意事項：答案は本紙解答欄に書け。

問 1 (各 3 点, 計 15 点)

次の文の空欄に最も当てはまる語句を選択肢より選び、記号で答えよ。

- 「 $1-a + 1-b = 1-c$ 」はWirthの有名なキャッチフレーズであり、Wirthの著書であるプログラミングの教科書の題名にもなっている名文句である。
- 遅いがメモリを食わないアルゴリズムと、速いがメモリを大食いするアルゴリズムの選択を迫られるシチュエーションを **1-d** と **1-e** のトレードオフという。

1-a,b,c 選択肢: ア・アルゴリズム, イ・オペレーティングシステム, ウ・自然言語, エ・データ構造, オ・プログラミング言語, カ・プログラム, **1-d,e 選択肢:** キ・アクセス速度, ク・空間, ケ・コスト, コ・時間

解答	解答	解答	解答	解答
1-a: ア	1-b: エ	1-c: カ	1-d: ク	1-e: コ

問 2 (各 3 点, 計 9 点)

以下の問題文が正しい場合には解答欄に○を、誤りの場合は×を記入せよ。

- 2-a)** 実際に動かして実行時間を測定する方法はアルゴリズムの評価に適している。
2-b) アルゴリズムの性能を計るためにプログラムを書いてコンピュータで実行して時間を測定する方法は、プログラムを書く際のコーディングの技量に左右される。
2-c) 計算量は入力データのサイズ n の関数として表現する。

解答	解答	解答
2-a: ×	2-b: ○	2-c: ○

問 3 (各 4 点, 計 16 点)

以下の問に答え、解答欄に記入せよ。オーダー表記 $O()$ の括弧の中はオーダーが変わらない範囲で最も簡単な式を書け。

- 3-a)** 実際の計算時間が $3 \log n + 10n$ だった場合の計算量を $O()$ (オーダー) 表記で書け。
3-b) $O(n^2)$ の計算量の計算の後に $O(n \log n)$ の計算を行った場合の全体の計算量を $O()$ (オーダー) 表記で書け。
3-c) 線形探索法での 1 個の要素の探索に必要な計算量を $O()$ (オーダー) 表記で書け。
3-d) 二分探索法での 1 個の要素の探索に必要な計算量を $O()$ (オーダー) 表記で書け。

解答	解答	解答	解答
3-a: $O(n)$	3-b: $O(n^2)$	3-c: $O(n)$	3-d: $O(\log n)$

問 4 (各 5 点, 計 20 点)

現在スタックに格納されている要素は 1 つで、その値は 1 とする。スタックに以下の操作を順に行なう。push はスタックへ値を積む操作、pop はスタックから値を降ろす操作である。ここで、 a から g は int 型の変数とする。

操作: push(2), push(3), push(4), $a = \text{pop}()$, push(5), push(6), $b = \text{pop}()$, push(7), $c = \text{pop}()$, push(8), $d = \text{pop}()$, $e = \text{pop}()$, $f = \text{pop}()$, $g = \text{pop}()$
 操作が終了した後に d, e, f, g に格納されている値を順に以下の解答欄に答えよ。但し、スタックの長さは十分にあるとする。

解答 (4a) 変数 d 8	解答 (4b) 変数 e 5	解答 (4c) 変数 f 3	解答 (4d) 変数 g 2
---------------------	---------------------	---------------------	---------------------

問 5 (5 点)

待ち行列の実装は教科書と同様、すなわち、待ち行列本体の配列 queue、先頭の要素の添字を記憶する front と、末尾の要素の次の要素の添字を記憶する rear というグローバル変数で実現されており、要素はリング状に格納され、引数で与えられた添字の要素の次の要素の添字を返す関数(配列の最後の添字の次は最初の添字を返す) next が実装されているとする。待ち行列にデータを入れる関数 enqueue を完成させるために図 1 の空欄を埋める式を解答欄に書け。(注:教科書から要素の型を整数型に変更しており、待ち行列がフルのときのエラーチェックも省略している)

```
enqueue(int x)
{
    queue[ 5 ] = x;
    rear = next(rear);
}
```

図 1. 関数 enqueue

解答
5: rear

問 6 (各 5 点, 計 25 点)

次の図2はリストのメモリ内部での連結リストによる実現状況である。図2で縦に繋がった2つの四角を合わせたものが一つのセルを表すものとする。セルの上が next メンバ、下が value メンバである。next メンバは次のセルへのポインタ(次のセルのアドレス)を、value メンバはそのセルに格納するデータの値をもつとする。最後のセルの next には NULL を記入するものとする。

各セルには「アドレス」が記されており、下方にはセルの名称として「header」または「#」を付けた識別名が書かれている。header はダミーで、header の next メンバには 1 個目のセルのアドレス「1400」が格納されている状態である(図2の header の next メンバ参照)。

6-a,b) header.next で指されている1個目のセルの識別名(#名称)と格納されているデータの値を解答欄に答えよ。

6-c) 連結リストが実現するリストが (1,2,3,4) となるように図2のセルの next メンバ部分にアドレスを記入せよ。

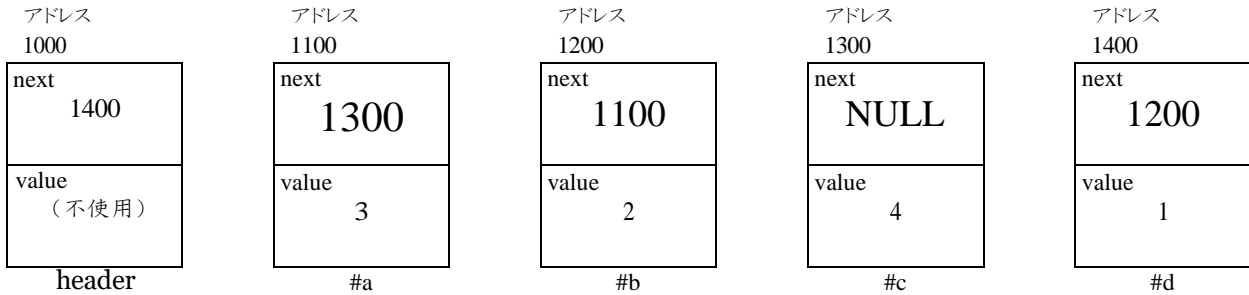


図2 連結リストのメモリ上のイメージ

6-d) 6-c の状態のリストに対し、`header.next = header.next->next;` を実行した後の連結リストが実現するリストを(1, 2, 3, 4) のようなカンマと括弧の記法で解答欄へ書け。

6-e) 6-c の状態のリストの最後のセルの next メンバを変更して、各要素が環状に連結された循環リストとする方法を考える。6-c の状態のリストにおいて、NULL を設定した最後のセルの next メンバへ設定するべきアドレスを解答欄へ書け(リストの頭を用いた方法、用いない方法のいずれでも構わない)。

解答(セルの識別名)	解答(データの値)	解答	解答
6-a: #d	6-b: 1	6-d: (2, 3, 4)	6-e: 1400または1000

問 7 (各 5 点, 計 10 点)

次の図3は教科書5.3節と同様の双方向リストの概念図であり、図4は図3をメモリ内部での双方向リストにより実現した状況を示している。図4で縦に繋がった3つの四角を合わせたものが一つのセルを表すものとする。セルの上が next メンバ、中が prev メンバ、下が value メンバである。next メンバは次のセルへのポインタ(次のセルのアドレス)を、prev メンバは前のセルへのポインタを、value メンバはそのセルに格納するデータの値をもつとする。

各セルには「アドレス」が記されており、下方にはセルの名称として「head」または「#」を付けた識別名が書かれている。head はダミーで、head の next メンバには 1 個目のセルのアドレス「1200」が格納されており、head の prev メンバには最後のセルのアドレス「1300」が格納されている状態である(図4の head の next メンバおよび prev メンバ参照)。

7-a) 双方向リストが実現するリストが図3のとおり (1,2,3,4) となるように図4のセルの next メンバ部分にアドレスを記入せよ。

7-b) 双方向リストが実現するリストが図3のとおり逆順で (4,3,2,1) となるように図4のセルの prev メンバ部分にアドレスを記入せよ。

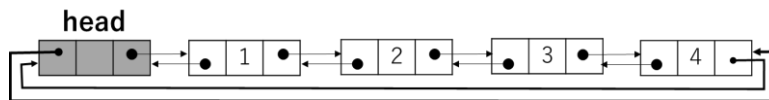


図3 双方向リストの概念図

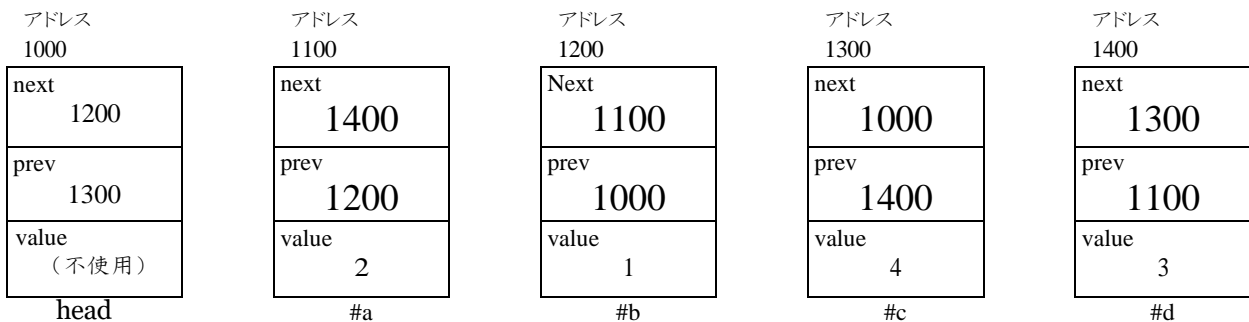


図4 双方向リストのメモリ上のイメージ