2021 年度 春・(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	JT-2, 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	4301 コンピュータ室	採点	
	可		金曜 3,4 限				

注意事項：答案は本紙解答欄に書け．

問 1 （各 2 点，計 10 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．

- プログラムを書くということと，プログラミング言語を使うということは [1-a] 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-b] と [1-c] のトレードオフという．
- [1-d] は [1-e] に比べて大きなバラつきがある．

1-a 選択肢：ア. 同じ技術である，**イ.** 別の技術である，**1-b,c 選択肢: ウ.** アクセス速度，**エ.** 空間，**オ.** コスト，**カ.** 時間 **1-d,e 選択肢: キ.** アルゴリズムの性能，**ク.** マシンの性能

解答	解答	解答	解答	解答
1-a:	1-b:	1-c	1-d	1-e

問 2 （各 2 点，計 6 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) マシンやコンパイラのベンチマークテストではハードウェアやコンパイラの性能に影響されない表現が用いられる．

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値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 5 点，計 15 点）

以下の問に答え，解答欄に記入せよ．$O()$ 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 $10\log n + \frac{1}{2}n$ 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) 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
3-a:	3-b:	3-c:

問 4（各 5 点，計 15 点）

要素が int 型の待ち行列に先頭から順に 1,2,3,4 が格納されているとする．enqueue は待ち行列に値を入れる操作，dequeue は待ち行列から値を取り出す操作であり，a, b, c は int 型の変数とする．以下の操作系列 A を 繰り返す時，以下の問に答えよ．但し，待ち行列の長さは十分にあるとする．

操作系列 A: a = dequeue(), b = dequeue(), c = dequeue(), enqueue(c),e nqueue(b), をこの順に行う．ただし，空の待ち行列に対して dequeue() を**行った**時点で dequeue は値を返さずプログラムは終了する．

4-a) 最初の操作系列 A を行った直後の待ち行列末尾の要素の値を答えよ．

4-b) プログラムが終了するまでに何回操作系列 A が**開始**されるか答えよ．

4-c) プログラム終了前の最後に値を返した dequeue() が返す値を答えよ．

解答	解答	解答
4-a:	4-b:	4-c:

問 5 (10 点)

スタックが教科書と同様に実装されているとする（要素のデータの型のみ int に変 更してある）．すなわち，スタックの本体をグローバル変数である int 型配列 stack とし，stack[0] がスタックの底であるとする．また，スタックポインタとして，スタックに入っている要素の個数を整数型のグローバル変数 n で記憶する．このとき，スタックに引数で与えられたデータ x を積む関数 push 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：教科書からスタックのオーバーフロー防止の処理を省略している)

解答
5:

```
void push(int x)
{
    stack[  5  ] = x;
    n++;
}
```

図 1. 関数 push

問 6（各 10 点，計 20 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個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．header はダミーで，header の next メンバが 1 個目のセルへのポインタであるとする（図 2 の header の next メンバ参照）．

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

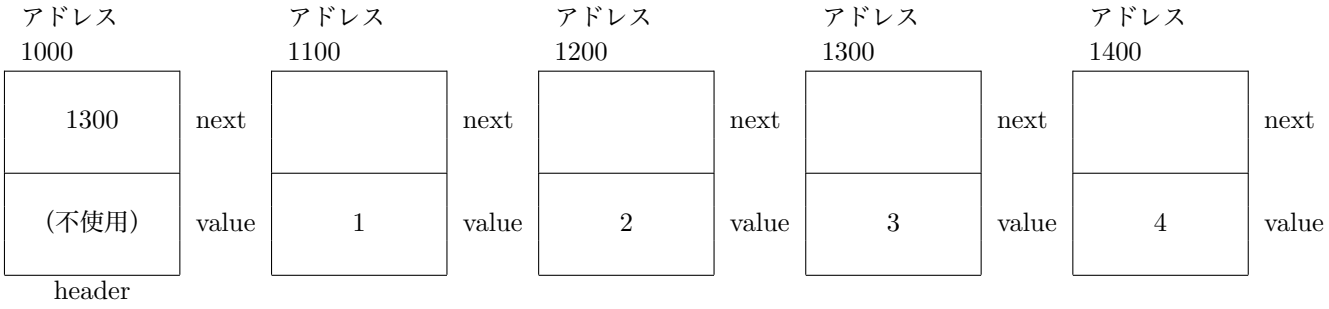

図 2 連結リストのメモリ上のイメージ

6-b) 6-a 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書け．

解答 6-b:

問 7 (12 点)

連結リストが教科書と同様に実装されているとする．すなわち，セルは struct CELL 構造体で，その next メンバは次のセルへのポインタ，value メンバはそのセルに対応するデータの値を格納するものとする．struct CELL 型のグローバル変数 header が宣言されており，header.next には連結リストの先頭要素を指すポインタが格納されているとする．

図 3. は引数 p, v をとり，p で指されたセルの直後に値が v であるセルを挿入する関数 insert_position である．空欄にあてはまる文を解答欄に書け．（エラーチェックは省略している）

```
insert_position(struct CELL *p, int v)
{
  struct CELL *xp;
  [ 7-a ] = malloc(sizeof(struct CELL));
  xp->value = v;
  [ 7-b ] = p->next;
  p->next = xp;
}
```

図 3. 関数 insert_position

解答 7-a:	解答 7-b:

問 8（12 点）

双方向リストが教科書と同様に実装されているとする．すなわち，セルは struct CELL 構造体（上の問とは異なることに注意）で，その prev メンバは前のセルへのポインタ，next メンバは次のセルへのポインタ，value メンバはそのセルに対応するデータの値を格納するものとする．

図 4. は struct CELL 型へのポインタ p と要素の値 v を引数とし，p が指す要素の直後に値が v の要素を挿入する関数 insert である．図中の空欄にあてはまる文を解答欄に書け．（エラーチェックは省略している）

```
void insert(struct CELL *p, int v)
{
  struct CELL *xp;
  xp = malloc(sizeof(struct CELL));
  xp->value = v;
  xp->prev = p;
  xp->next = p->next;
  p->next->prev = [ 8-a ];
  [ 8-b ] = xp;
}
```

図 4. 関数 insert

解答 8-a:	解答 8-b: