

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-1, 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	4303 コンピュータ室	採点
	可		火曜 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)

以下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** であるチューリングマシンは記憶装置として無限長の **1-b** を持っている。チューリングマシンはアルゴリズムの **1-c** を議論するために考案されたが後に **1-d** を求めるために使われるようになる。
- アルゴリズムとデータ構造は **1-e** ことが望ましい。

1-a 選択肢：ア. コンピュータの数学的モデル, **イ.** 最初に動作したコンピュータ, **1-b 選択肢：ウ.** テープ, **エ.** 半導体メモリ, **オ.** 表示装置, **1-c 選択肢：カ.** 高速性, **キ.** 実用性, **ク.** 停止性, **1-d 選択肢：ケ.** アルゴリズムの複雑さ, **コ.** 実際の計算時間 **1-e 選択肢：サ.** 同時に学ぶ, **シ.** 独立に学ぶ

解答 1-a:	解答 1-b:	解答 1-c:	解答 1-d:	解答 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)** アルゴリズムの性能を計るためには実際にそのアルゴリズムを実現する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のが良い。
- 2-b)** 実在のマシンを使ってアルゴリズムの性能を評価するというのは客観性に難がある。
- 2-c)** 計算量を求めるときに最速となる入力データを想定したものを最大計算量という。

解答 2-a:	解答 2-b:	解答 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 $2 \log n + 3n$ だった場合の計算量を O (オーダー) 表記で書け。
- 3-b)** O(n) の計算量をもつループを O(n) 回繰り返す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 個の要素の登録に必要な計算量を O (オーダー) 表記で書け。

解答 3-a:	解答 3-b:	解答 3-c:
------------	------------	------------

問 4 (各 5 点, 計 15 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), a = pop(), push(4), push(5), push(a), b = pop(), push(6), push(b), c = pop(), d = pop(), e = pop(), f = pop(), g = pop()
 操作が終了した後に 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 4-a: 変数 e	解答 4-b: 変数 f	解答 4-c: 変数 g
--------------------	--------------------	--------------------

問 5 (10 点)

待ち行列が教科書と同様に実装されているとする。すなわち, グローバル変数の queue が待ち行列本体を表現する配列で, グローバル変数の front が待ち行列の先頭を, rear が末尾の次の要素を指すものとする。また, 教科書と同様にリング状に要素を格納する。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。待ち行列からデータを取り出す関数 dequeue 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いる)

解答 5:

```
int dequeue()
{
    int x;
    if(front == rear)
        exit(1);
    x = queue[ 5 ];
    front = next(front);
    return(x);
}
```

図 1. 関数 dequeue

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 (1, 2, 3, 4) 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。

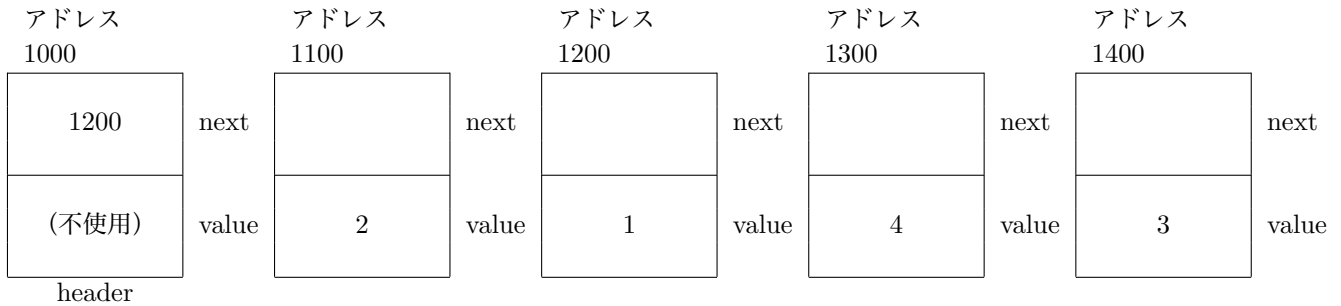


図 2 連結リストのメモリ上のイメージ

6-b) 6-a の状態のリストに対し、`header.next->next = header.next->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. は連結リストの要素全てを空白をはさんで順に出力する関数 print_all である。空欄にあてはまる文を回答欄に書け。(エラーチェックは省略している)

```
print_all()
{
    struct CELL *p;
    p = header.next;
    while (  ) {
        printf("%d ", p->value);
        ;
    }
}
```

図 3. 関数 print_all

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 を引数とし、p が指す要素を削除する関数 delete である。空欄にあてはまる文を解答欄に書け。(前後には十分セルが続いているとする。消すセルが存在しない場合のエラーチェックは省略している。また消される値を返す処理は無い。)

```
void delete(struct CELL *p)
{
    p->prev->next = p-> ;
    p->next->prev = p-> ;
    free(p);
}
```

図 4. 関数 delete

解答
8-a:

解答
8-b: