



受験番号	
名前	

データ構造とアルゴリズム

うさぎ模試 (C 言語)

(90 分・100 点)

★注意事項 解く前に絶対見てね!★

1. 問題用紙はこの冊子を含め全部で 11 ページである。
2. 解答はすべて問題用紙とは別の解答用紙に解答すること。なお、解答用紙はマークシート用紙と記述用紙の 2 枚がある。
3. 本冊子に誤字・脱字・意味不明な日本語及び問題用紙の汚れ等があった場合、手を挙げてどうしてもできないので適当に想像しながら読むこと。
4. 問題は第 1 問～第 4 問があり、すべての問題が必答問題である。
5. マークシートの解答番号は ～ である。マークシートの採点には Google フォームを使用する (QR コードが Google フォームである)。
6. 記述用の問題の解答には黒色鉛筆・シャープペンシル・ボールペン・万年筆・iPad・PC などなにを使っても構わないが、**本番に筆記用具を忘れると悲惨な目に合うので試験当日は筆記用具を忘れないこと。**
7. このテストで用いられている言語は日本語と C 言語である。そのため、C 言語をある程度 (構造体、ポインタ、スタック、キュー、2 分探索木などを完全に理解したレベル) 勉強した人向けの試験となっています。
8. 解説はサイト「工業大学生ももやまのうさぎ塾」で公開しています。記述部分の採点はこちらのサイトにて自己採点をお願いします。マーク部分は Google フォームで自動的に採点されます。 Google フォーム : <http://bit.ly/da-usagi-ans>
解説記事 (予定) : <https://www.momoyama-usagi.com/entry/info/algo/moshi>

1. [小問集合] (配点 30) マーク番号: ~

次の(1)~(8)の問いに答えなさい。

(1) つぎのCプログラムの実行結果として正しいものを①~⑧の中から選びなさい。

マーク番号:

```
int f1(int a, int *b){
    a += 10;
    *b += 10;
    return a;
}

int main(void) {
    int a = 10, b = 10, c = 10;
    c = f1(a,&b);
    printf("%d,%d,%d\n",a,b,c);

    return 0;
}
```

- の選択肢
- ① 10,10,10 ② 10,10,20 ③ 10,20,10 ④ 10,20,20
⑤ 20,10,10 ⑥ 20,10,20 ⑦ 20,20,10 ⑧ 20,20,20

(2) 関数 f2 が下のように定義されている。

このとき、f2(180,54) の返り値として正しいものを選びなさい。マーク番号:

```
int f2(int a, int b) {
    if(b == 0) {
        return a;
    }
    else {
        return f2(b,a % b);
    }
}
```

- の選択肢
- ① 1 ② 2 ③ 3 ④ 9 ⑤ 18 ⑥ 36 ⑦ 54 ⑧ 180

(3) うさぎ塾の生徒 1,000 人の模試成績が学生番号順に格納されているデータから、データを探し出したい。

[i] 線形探索、2分探索それぞれでデータを探索する際の最大探索回数（目的のデータかどうかをチェックするために必要な最大の回数）として正しいものをそれぞれ下から選びなさい。

線形探索： 2分探索：

・ の選択肢
① 1 ② 2 ③ 8 ④ 9 ⑤ 10 ⑥ 100 ⑦ 250 ⑧ 1,000

[ii] 4ヶ月後、うさぎ塾は大繁盛し、生徒は4倍の4,000人に増えた。データ数が4倍になったとき、線形探索、2分探索の探索回数はどうなるか。最も正しいものを選びなさい。

線形探索： 2分探索：

・ の選択肢
① 変わらない ② 2倍になる ③ 4倍になる ④ 16倍になる
⑤ 1回増える ⑥ 2回増える ⑦ 3回増える ⑧ 4回増える

(4)

関数 matMul は、 n 次行列の積を計算する関数である。

```
void matMul(int a[n][n],int b[n][n], int c[n][n]) {
    for(int i = 0; i < n; i += 1) {
        for(int j = 0; j < n; j += 1) {
            for(int k = 0; k < n; k += 1) {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }
}
```

この関数の n に対する計算量（オーダー）として、最も適切なものを選びなさい。

マーク番号：

の選択肢
① $O(1)$ ② $O(n)$ ③ $O(n^2)$ ④ $O(n^3)$ ⑤ $O(2^n)$ ⑥ $O(3^n)$

(5) 関数 bubble_sort は昇順のバブルソートを行うプログラムである。

```
void bubble_sort(int data[],int n) {
    int tmp;
    for(int i = 0; i < n - 1; i += 1) {
        for(int j = n - 1; j >= i + 1; j -= 1) {
            if(data[j-1] > data[j]) {
                // 入れ替え始まり (ここが実行される回数は?)
                tmp = data[j-1];
                data[j-1] = data[j];
                data[j] = tmp;
                // 入れ替え終わり
            }
        }
    }
}
```

main 関数が下のように与えられているとき、配列 a のデータが入れ替わる回数は何回になるか。その回数をマーク番号 にマークしなさい。

```
int main(void) {
    int a[4] = {2,4,3,1};
    bubble_sort(a,4); // [2,4,3,1] を [1,2,3,4] になるようにバブルソート
    return 0;
}
```

(6) 関数 findpivot1, findpivot2 はクイックソートのピボット (基準値) を決める関数である。

```
int find_pivot1(int data[], int left, int right) {
    int mid = (left + right) / 2;
    return data[mid];
}

int find_pivot2(int data[], int left, int right) {
    int mid = (left + right) / 2;
    int pivot_kouho[3] = {data[left], data[mid], data[right]};
    bubble_sort(pivot_kouho,3); // 昇順ソート
    return pivot_kouho[1];
}
```

data[10] = {6, 3, 1, 4, 0, 8, 2, 7, 5, 9} とするとき、find_pivot1(data,0,9) および find_pivot2(data,0,9) の返回值として正しいものをマークしなさい。

(例: 3 が返回值だと思ったら 3 にマーク)

find_pivot1 : find_pivot2 :

(7) つぎの探索アルゴリズムの中で、番兵が有効なものはどれか。マーク番号：

の選択肢

- ① 深さ優先探索 ② 線形探索 ③ 2分探索
④ 幅優先探索 ⑤ ハッシュ探索

(8) データ構造に関する記述のうち、適切なものはどれか。

[基本情報技術者 平成 17 年度 問題 13]

マーク番号：

の選択肢

- ① 2分木は、データ間の関係を階層的に表現する木構造の一種であり、すべての節が二つの子をもつデータ構造である。
- ② スタックは、最初に格納したデータを最初に取り出す先入れ先出しのデータ構造である。
- ③ 線形リストは、データ部と次のデータの格納先を指すポインタ部から構成されるデータ構造である。
- ④ 配列は、ポインタの付替えだけでデータの挿入・削除ができるデータ構造である。

2. [スタック・キュー] (配点 20) マーク番号: ~

次の(1)~(3)の問いに答えなさい。

ただし、それぞれの問いにある図のスタック・キューは下から順番にデータが入る。

(1) 空の状態のスタックとキューから、次の手続を順に実行していく。このとき、図の

~ に入っているデータの値を解答欄にマークしなさい。ただし、空欄の(何も入っていない)場合は0をマークすること。

ここで、手続きに引用している関数は、次のとおりとする。

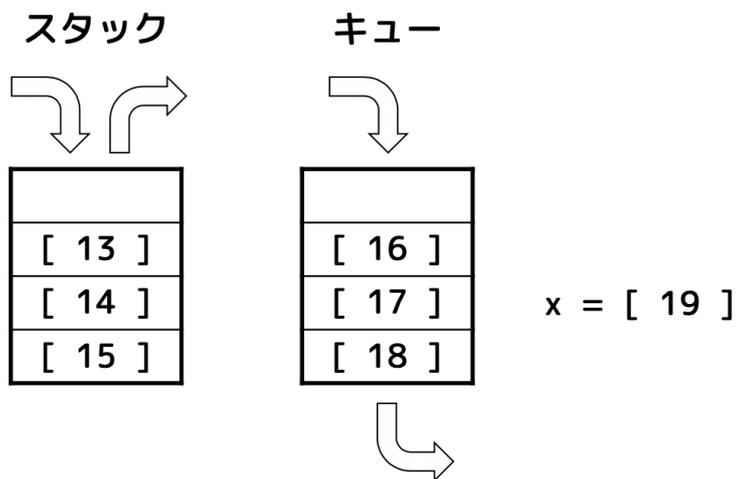
※ (2)以降も同じ関数を用いる

★関数の定義★

push(x) : データ x をスタックに積む。
pop() : データをスタックから取り出して、その値を返す。
enqueue(x) : データ x をキューに挿入する。
dequeue() : データをキューから取り出して、その値を返す。

★手続き★

```
enqueue(1)
push(2)
enqueue(3)
push(dequeue())
enqueue(4)
enqueue(pop())
push(5)
enqueue(dequeue())
enqueue(pop())
push(6)
x = dequeue()
```



(2) ある状態（空ではない）から次の手続きを順に実行すると、スタック・キューの中身は下の図のようになった。

このとき、X, Y に当てはまるデータの値をマークしなさい。ただし、データの値が確定しないときは0をマークすること。

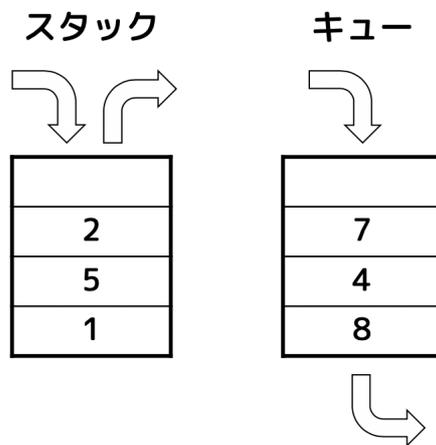
ただし、*, X, Y には 1~9 の値が入り、それぞれの * は違う数字となる。

X の値 : Y の値 :

★手続き★

```

push(X)
enqueue(*)
push(*)
enqueue(pop())
enqueue(Y)
push(dequeue())
a = pop()
a = pop()
a = dequeue()
    
```

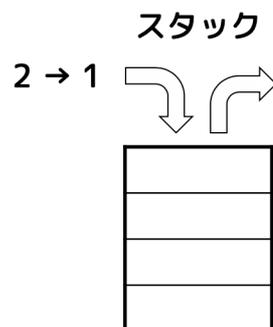


(3) つぎの図で表される 2 つの順序(i), (ii)で入力されるデータがある。各データについて 1 回ずつスタックの挿入 (push) と取り出し (pop) を行うことができる場合、データの出力順序は何通りになるかをそれぞれマークしなさい。

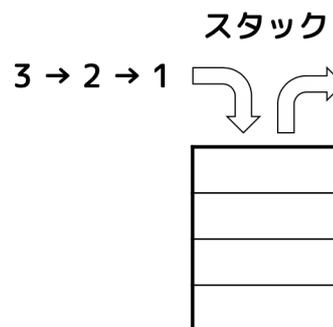
(i) 1→2 : 通り

(ii) 1→2→3 : 通り

(i)



(ii)



3. [2分探索木] (配点 20) マーク番号: ~

次の(1)~(2)の問いに答えなさい。ただし、(1), (2)ともに**同じデータ値のノードは考えない**ものとする。

(1) 下図のような2分木の走査順を考える。

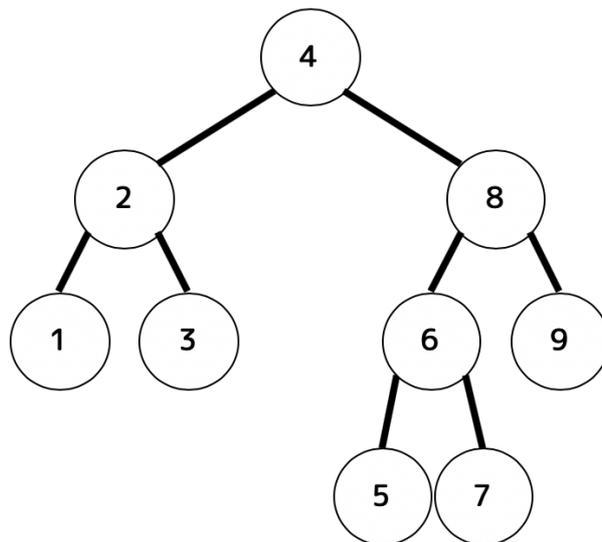
このとき、つぎの[i]~[iv]の走査方法で、4番目と7番目に走査されるデータの値をそれぞれマークしなさい。

[i] 幅優先探索 → 4番目: 7番目:

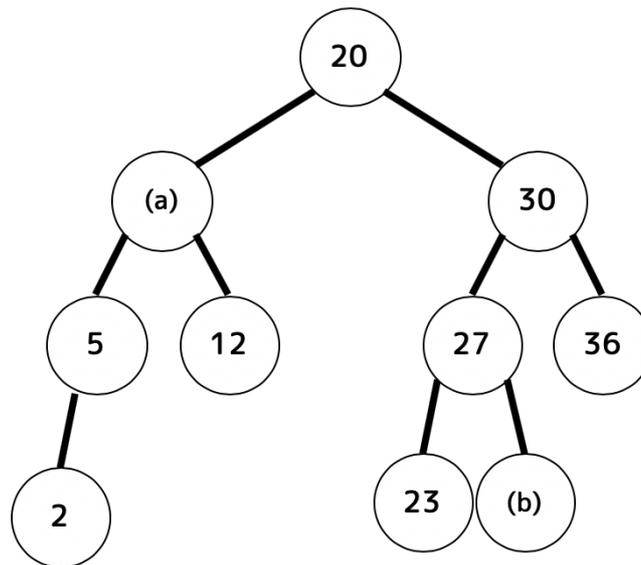
[ii] 行きがけ順 (先行順) → 4番目: 7番目:

[iii] 通りがけ順 (中間順) → 4番目: 7番目:

[iv] 帰りがけ順 (後行順) → 4番目: 7番目:



(2) (1)とは別の下図のような2分探索木(順序木となる2分木)を考える。
 このとき、[i], [ii]の問いに答えなさい。



[i] (a)と(b)に入れられる自然数(1以上の整数)は全部で何通りあるかをマークしなさい。ただし、無限に存在する場合は0をマークすること。(0通りでも0をマーク)

[注意: 2分探索木内に同じ値のノードは2つ以上含まれない]

(a): 通り (b): 通り

[ii] 子ノードを持つ親ノードを削除することを考える。このとき、自身以下の部分木から新しく親となるノードを移動させることで順序木を崩さずに2分探索木を構成することができる。

ここで、次の(ア),(イ)で表されるノードを削除するとき、それぞれどのノードを新しく親にすれば(削除した場所にノードを移動させれば)順序木を崩さずに2分探索木を構成することができるかを下の選択肢の中から**すべて**選び、該当する番号をマークしなさい。

(ア) 5を削除したとき:

(イ) 20を削除したとき:

<input type="text" value="34"/>	・	<input type="text" value="35"/>	の選択肢			
① 2	② 5	③ 12	④ 23	⑤ 27	⑥ 30	⑦ 36
⑧ [i]の(a)		⑨ [i]の(b)				

マークシートに答える問題は以上です。

4. [総合問題] (配点 30) 解答は記述用解答用紙に記入しなさい。

某 Y 音楽能力検定の指導グレードの成績を処理するプログラムを作りたい。

(1)~(3)の問いに答えなさい。

ここで、指導グレードの成績はソルフェージュ(solf)・鍵盤実技(key)・調音(choon)・楽典(gakuten)・コード(chord)の5つで評価される(成績は0点~100点の1点刻み)。下の表は、指導グレードの成績の例である。

受験番号	名前	ソルフェ	鍵盤実技	調音	楽典	コード
1	Tonkatsu	63	72	81	100	92
2	Shibainu	71	79	86	88	88
4	DAMIDAMI	54	75	90	89	81
6	S.Ringo	70	66	85	77	70
7	Tako	75	74	76	78	80
5	n	77	61	79	99	90
3	Usakko	60	63	83	94	91

(1) 受験番号を id、名前を name、英語の得点を eng、ソルフェージュの成績を solf、鍵盤実技の成績を key、調音の成績を choon、楽典の成績を gakuten、コードの成績を chord を管理する構造体 RESULT を解答欄に記入しなさい。ただし、次の条件に注意しながらデータの型を決めること。

★条件★

- ・ 名前は 20 桁以内の英数字で管理される。
- ・ 受験番号は必ず数字となり、小数、英字は入らない。
- ・ 各 5 科目はそれぞれ 100 点満点で、0~100 点の 1 点刻みで採点される。

(2) 次の[i], [ii]の処理を行う関数を作りなさい。

[i] 構造体 RESULT の変数である r を引数としてとり、r に格納されている 5 科目の合計点数を出力する関数 sum_score。

(ある受験者 r の 5 科目の合計点を出力する関数を作ればよい。)

[ii] 構造体 RESULT の変数である r を引数としてとり、受験者 r が合格なら 1、不合格なら 0 を返す関数 isPass。(必要ならば関数 sum_score を用いてよい)

ただし、合格条件は下の 2 つの条件を両方満たした場合である。

(ある受験者 r の合格を判定する関数を作ればよい。)

★合格条件★

- ・それぞれの項目の平均点が 75 点以上
- ・5 つの項目のうち、60 点未満の項目が 1 つも存在しないこと

前のページにある表の受験者のうち、合格しているのは受験番号が 1 番、2 番、3 番、5 番、7 番の人である。(4 番は 60 点未満の科目あり、6 番は平均点が 75 点未満のため不合格)

(3) 指導グレードの成績が構造体 RESULT の配列で格納されている。ただし、データは全くのランダムな順に並んでいる。(受験番号順でもなく、成績順でもない。)

ここで、ある受験番号 id が与えられたときに、与えられた id の受験者が合格しているかどうかを調べたい。

[i] 構造体 RESULT の配列から受験番号 id を探索したい場合、線形探索・二分探索のどちらで行うのが効率的か。理由とともに答えなさい。

[ii] 構造体 RESULT の配列 r、配列 r の要素数 n (受験人数)、受験番号 myid を引数としてとり、受験番号 myid の人が合格していれば 1、不合格なら 0、該当する受験番号の人がいなければ -1 を返す関数 check_res を作りなさい。ただし、必要ならば関数 sum_score、関数 isPass を用いても構わない。

(※ ただし[i]で答えた探索アルゴリズム以外を用いても構わない。)

問題は以上です。お疲れさまでした。