2026 年度

佐賀大学大学院入学試験問題

一般入試

理工学研究科

数理・情報分野

（数学コース・データサイエンスコース・知能情報工学コース）

専門科目-１（数学基礎）

―――――――――――――――― 解答上の注意事項 ――――――――――――――――

1　「解答始め」の合図があるまで問題を見てはならない。

2　「解答始め」の合図があったら，全ての解答紙の所定欄に受験番号を記入すること。

3　問題の解答は，別に指示がある場合を除き，所定の解答欄に記入すること。

4　試験時間中，試験問題の内容について質問がある場合は，手をあげて監督者に申し出ること。

5　各解答紙には，問題番号（1、2）を明記し，1と2は、解答紙を分けること。
解答紙１枚に複数の問題を解答してはならない。

6　解答紙は裏面を用いてもよいが，裏面を用いる場合は表面の右下に「つづく」と明記すること。

7　その他，監督者の指示に従うこと。

（このページを下書きに用いてよい）

科目名	数学基礎（微分積分学・線形代数学）

採点は解答用紙のみで行う．計算/下書き用紙は採点対象外なので，
答えに至るまでの計算過程や説明等はすべて解答用紙に記入すること．

1 以下の問に答えよ．

(1) 広義積分 $\displaystyle\int_0^{\infty} \frac{e^{-x}}{e^{x}+e^{-x}}dx$ の値を求めよ．

(2) 関数 $f(x)=\cos(\pi \sin x)$ の $x=0$ におけるテイラー展開を
$f(x)=a+bx+cx^2+\cdots$ とするとき，定数 a，b，c の値を求めよ．

(3) $z=g(x,y)$ を C^2 級関数とする．$s=e^{\frac{x+y}{2}}$, $t=\dfrac{x-y}{2}$ とするとき，
偏導関数 z_s および z_{st} をそれぞれ z_x, z_y, z_{xx}, z_{xy}, z_{yy}, s, t で表せ．

(4) 次の重積分の値を求めよ．

$$I=\iint_D \log(x^2+y^2)dxdy, \qquad D=\left\{(x,y) \;\middle|\; 1\le x^2+y^2\le 2,\ x\ge 0,\ y\ge 0\right\}.$$

2

行列 $A=\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 4\\ 1 & 0 & 0\end{bmatrix}$ および $B=\begin{bmatrix}2 & 0 & 0\\ 0 & 2 & 0\\ 3 & 1 & 0\end{bmatrix}$ について，以下の問に答えよ．

(1) A のランク（階数）を求め，A が正則か否かを判定せよ．
また，A が正則である場合には，A の逆行列を求めよ．

(2) B の固有値を全て求めよ．
さらに，各固有値に対応する固有ベクトルを1つずつ求めよ．

(3) B が対角化可能か否かを判定せよ．また，B が対角化可能な場合には，
対角化する行列を一つ求めよ．

「解答始め」の合図があるまで問題を見てはならない。

2026 年度

佐賀大学大学院入学試験問題

一般入試・外国人留学生特別入試

理工学研究科

数理・情報分野

（データサイエンスコース・知能情報工学コース）

専門科目-2 （２群）

（情報専門：プログラミング【必須】，①～④より１科目を【選択解答】）

――――――――――――――― 解答上の注意事項 ―――――――――――――――

1　「解答始め」の合図があるまで問題を見てはならない。

2　「解答始め」の合図があったら，全ての解答紙の所定欄に受験番号を記入すること。

3　問題の解答は，別に指示がある場合を除き，所定の解答欄に記入すること。

4　試験問題の内容について質問がある場合は，手をあげて監督者に申し出ること。

5　「プログラミング」は必ず解答すること。①～④の４科目のうち，任意の１科目を選択して解答すること。

6　各解答紙には，解答する科目名を明記し，科目ごとに解答紙を分けること。
解答紙１枚に複数の科目を解答してはならない。

7　解答紙は裏面を用いてもよいが，裏面を用いる場合は表面の右下に「つづく」と明記すること。

8　その他，監督者の指示に従うこと。

（このページを下書きに用いてよい）

科目名	プログラミング

数理・情報　分野

1　(1) C++言語で記述された下記の関数（サブルーチン）があります.

```
int func_P(double P) {
    double R = rand() / (RAND_MAX + 1.0);    // 0 以上，1 未満の一様乱数を R に代入
    if (P > R) {
        return 1;
    }
    else {
        return 0;
    }
}
```

この関数 func_P の引数 P は，0 以上，1 未満の実数です.
「P」,「確率」,「戻り値」「関数」という言葉を使って,
この関数を 40 文字程度の日本語で説明しなさい.

(2) 上記の関数を呼び出す下記の関数（サブルーチン）があります.

```
---------------
int func_N(int N, double P) {
    if (func_P(P) == 1) {
        return N;
    }
    else {
        return func_N(N + 1, P);
    }
}
---------------
```

この関数 func_N の引数 N は自然数で，引数 P はコインを投げて表が出る確率とします.
「(double) func_N(1, 0.4)」の戻り値が何を示すか,
「コイン」,「確率」という言葉を使って，40 文字程度の日本語で説明しなさい.

2 以下のC++言語プログラムの「/*回答部分*/」を，以下の指示に従って作成しなさい．

害虫調査関数：int bug_survey(int y, int x)が，整数型の配列：int bug_data[20][20]に，その区画(x,y)の害虫の有無の調査結果（1:有, 0:無）を代入します．以下の手順に従って，害虫の巣の有無の推定結果を，整数型の配列：int nest_data[20][20]に代入するプログラムを作成しなさい．「/*回答部分*/」以外は，変更できないものとする．

＜巣の有無の推定ルール＞

ある区画の害虫調査結果が「1:有」の場合で，かつ，その区画の上下左右の４つの区画のうち，２つ以上の区画の害虫調査結果が「1:有」の場合，その区画に巣は有る（1:有）と推定する．それ以外の場合は，巣は無い（0:無）と推定する．

ただし，区画の位置がx=0, x=19, y=0, y=19のいずれかである区画は，上下左右の４つの区画のうちいずれかが調査対象外であるため，初期値のままとし，いずれの値も入力しない．

```
-----------------------------------------
#include<iostream>
using namespace std;
int bug_survey(int y, int x);          //害虫調査関数（1:有, 0:無）

int main() {
        int bug_data[20][20] = { 0 };        // 害虫の有無：全ての要素を 0 で初期化
        int nest_data[20][20] = { 0 };       // 巣の有無：全ての要素を 0 で初期化
        for (int y = 0; y < 20; y++) {
                for (int x = 0; x < 20; x++) {
                        // 位置(x,y)の害虫（1:有, 0:無）
                        bug_data[y][x] = bug_survey(y, x);
                }
        }
        /*回答部分*/
}
-----------------------------------------
```

3 「右切り捨て可能素数」は，素数のうち，右から 1 桁ずつ削っていっても，途中のすべての数が素数である数です．例えば，「7393」は，右から 1 桁ずつ削っていっても，途中の「739 → 73 → 7」の全てが素数です．

入力された自然数 n が，「右切り捨て可能素数」である場合は「1 (または true)」を，そうでない場合は「0 (または false)」を返す関数「int is_truncatable_prime(int n) ;」を作成しなさい．

作成する関数内の素数判定は，引数 int n が素数なら 1(true)を，素数ではないなら 0(false)を返す素数判定関数「int is_prime(int n) ;」を呼び出して用いること．

```
#include <iostream>
#include <cmath>
using namespace std;

// 素数判定関数
int is_prime(int n) {
    if (n < 2) return 0;
    if (n == 2) return 1;
    for (int i = 2; i < n; i ++) {
        if (n % i == 0) return false;
    }
    return true;
}

int main() {
    int n;
    cout << "整数を入力してください: ";
    cin >> n;

    if (is_truncatable_prime(n)) {
        cout << n << " は右切り捨て可能素数です。" << endl;
    } else {
        cout << n << " は右切り捨て可能素数ではありません。" << endl;
    }
    return 0;
}
```

---

4 以下のプログラムを実行したときに，表示される画面を書きなさい．

```
#include<iostream>
using namespace std;
int main() {
        int data[4][8] = { 0 };        // 全ての要素を 0 に初期化
        int x = 0;                     // X 座標（横）の位置を初期化
        int y = 0;                     // Y 座標（縦）の位置を初期化
        int direction = 0;             // 方向 0:右下, 1:左下, 2:左上, 3:右上

        for (int i = 1; i < 21; i++) {
                data[y][x] = i;
                switch (direction) {
                case 0:
                        x += 1;
                        y += 1;
                        if (x == 8) {
                                x = 6;
                                direction = 1;
                        }
                        if (y == 4) {
                                y = 2;
                                direction = 3;
                        }
                        break;
                case 1:
                        x -= 1;
                        y += 1;
                        if (x == -1) {
                                x = 1;
                                direction = 0;
                        }
                        if (y == 4) {
                                y = 2;
                                direction = 2;
                        }
                        break;
                case 2:
                        x -= 1;
```

```
                        y -= 1;
                        if (x == -1) {
                                x = 1;
                                direction = 3;
                        }
                        if (y == -1) {
                                y = 1;
                                direction = 1;
                        }
                        break;
                case 3:
                        x += 1;
                        y -= 1;
                        if (x == 8) {
                                x = 6;
                                direction = 2;
                        }
                        if (y == -1) {
                                y = 1;
                                direction = 0;
                        }
                        break;
                }
        }
        for (int y = 0; y < 4; y++) {
                for (int x = 0; x < 8; x++) {
                        cout << data[y][x] << "¥t";
                }
                cout << endl;
        }
        cout << endl;
        return 0;
}
```

---

5 以下のプログラムが 1(true)を返す条件を，「引数 a」と「引数 b」という言葉を使って，以下の/*回答部分*/に 10～30 文字程度を追加して説明しなさい．

『このプログラムは，「　/*回答部分*/　」1(true)を返すプログラムである．』

```
#include <iostream>
using namespace std;

int func_X(int a, int b) {
    while (b != 0) {
        int r = a % b;
        a = b;
        b = r;
    }
    return a;
}

bool func_Y(int n) {
    if (n < 2) return false;
    if (n == 2) return true;
    if (n % 2 == 0) return false;
    for (int i = 3; i < n; i += 2) {
        if (n % i == 0) return false;
    }
    return true;
}

int main() {
    int a, b;
    cout << "2 つの整数を入力してください: ";
    cin >> a >> b;

    int g = func_X(a, b);
    cout << func_Y(g);
    return 0;
}
```

---

6 (6-1) 20×20 マスの迷路：int grid [20][20]があります．スタート位置（int sy, sx）とゴール位置（int gy, gx）を入力すると，スタートからゴールまでをつなぐ経路があるかを探索するプログラムを作成します．移動は上下左右のみとし，斜めの移動はできません．
grid の各マスの初期値は，「8：壁（移動不可）」，「1：通路（移動可）」のいずれかです．
grid の端は，壁で囲まれているものとします．
（grid[0][n] = grid[19][n] = grid[n][0] = grid[n][19] = 8, n∈{0,1,2,…,19} ）
以下に記載された【アルゴリズム】にしたがって，スタートからゴールまでをつなぐ経路があるかを表示するプログラムの回答部分を作成しなさい．

【アルゴリズム】
1．入力された迷路のスタート位置を「2：通路（探索済み）」に変更する．
2．「2：通路（探索済み）」と隣接する「1：通路（移動可）」のマスを，「2：通路（探索済み）」に変更し，変更したマスの数だけ flag をカウントアップする．
3．変更できる通路がなくなる（flag == 0）か，経路がゴールに到達する（grid[gy][gx] == 2）まで，2 を while 文で繰り返す．

```
#include <iostream>
using namespace std;

int grid[20][20] = {   /*略*/   };

int main() {
    int sy, sx, gy, gx;
    cout << "スタート位置 (y x): ";
    cin >> sy >> sx;
    cout << "ゴール位置 (y x): ";
    cin >> gy >> gx;

    if (grid[sy][sx] != 1 || grid[gy][gx] != 1) {
        cout << "スタートまたはゴールが通路でありません。" << endl;
        return 1;
    }

    grid[sy][sx] = 2;  // スタート位置を 2 に

    bool reached = false;
    bool impossible = false;
```

```
    while (!impossible && !reached) {
        int flag = 0;

        /*回答部分*/

        if (flag == 0) impossible = true;
        if (grid[gy][gx] == 2) reached = true;
    }
    if (reached) {
        cout << "ゴールにたどり着けます！" << endl;
    }else {
        cout << "ゴールにたどり着けません。" << endl;
    }
    return 0;
}
```

（6-2）上記のプログラムを以下のように，再帰呼び出しで探索するプログラムに修正しました．関数：bool dfs(int y, int x);を作成しなさい．

```
int gy, gx;
int main() {
    int sy, sx;
    cout << "スタート位置 (y x): ";
    cin >> sy >> sx;
    cout << "ゴール位置 (y x): ";
    cin >> gy >> gx;

    if (grid[sy][sx] != 1 || grid[gy][gx] != 1) {
        cout << "スタートまたはゴールが通路でありません。" << endl;
        return 1;
    }
    bool reached = dfs(sy, sx);
    if (reached) {
        cout << "ゴールにたどり着けます！" << endl;
    } else {
        cout << "ゴールにたどり着けません。" << endl;
    }
    return 0;
}
```

科目名	① 計算機アーキテクチャ	数理・情報　分野

A. 以下は、計算機の基本構成要素についての説明文である。空欄（ ア ）～（ コ ）に当てはまる語句を答えなさい。

1. 計算機の基本構成要素には、入力装置、出力装置、（ ア ）装置、（ イ ）装置、（ ウ ）装置、通信装置がある。
2. （ エ ）は計算機の頭脳に当たる部分で、主に（ ア ）装置と（ イ ）装置で構成されている。
3. （ ア ）装置は、主（ ウ ）からアプリケーションの機械語コードを取得（命令（ オ ））し、その機械語コードを解読（命令（ カ ））し、他の装置への（ ア ）信号を出力する。
4. （ イ ）装置は、整数（ イ ）、論理（ イ ）や文字を取り扱うユニットと、（ キ ）の（ イ ）を取り扱うユニットは完全に分けられている。
5. （ ウ ）装置には、（ ク ）などの磁気デバイス、（ ケ ）などの半導体デバイスおよび（ コ ）などの光デバイスがある。

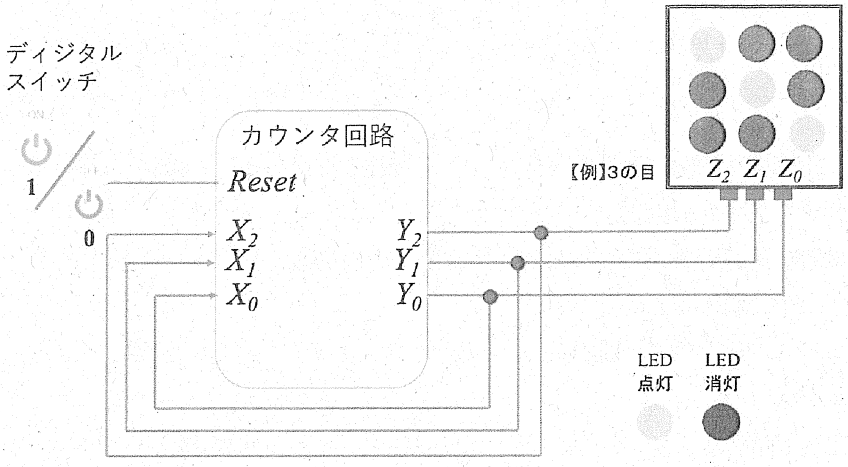
B. 整数値 +2025 および −2025 を、**ビッグエンディアンバイトオーダー**の**2バイト符号つき整数で表現**しなさい。なお、**負整数は「2の補数」で表現**し、解答は2進数 $(******)_2$ または16進数 $(******)_{16}$ で行いなさい。16進数では、0～9, A～Fを用いなさい。

C. 10進実数の「−2025.625」を、**ビッグエンディアンバイトオーダー**のIEEE754単精度浮動小数点形式（符号S＝1ビット、オフセットつき指数部B＝8ビット、仮数部A＝23ビットの合計32ビット）で表しなさい。なお、解答は、2進数 $(******)_2$ または16進数 $(******)_{16}$ で行いなさい。10進実数からIEEE754単精度浮動小数点形式への変換は、次のような手順で行うことができる。

《変換手順》

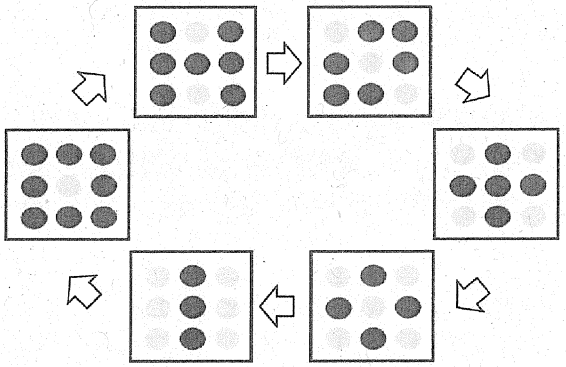
- 変換対象の実数Xが正数またはゼロならS＝0、負数ならS＝1
- 実数Xの絶対値|X|を、$(1.*****)_2 \times 2^k$ の形に正規化し、小数点より後ろの*****の部分を、23ビット（MSB（Most Significant Bit）側に詰めること）で表した数を仮数部Aとする。
- k＋127を8ビット2進数表示した数をオフセットつき指数部Bとする。
  - （例えば、k=0の場合は$(01111111)_2$、k=＋1の場合は$(10000000)_2$、k=−1の場合は$(0111\ 1110)_2$となる）
- **S、B、Aの順に左から右へ**並べる。（順番に注意すること）

D. 次ページのような、ボタンを押すと1、放すと0を出力する1ビットディジタルスイッチ、リセット端子と3ビットフィードバックデータ入力および3ビットデータ出力を有するカウンタ回路、3×3ドットマトリクスLEDを使って、電子サイコロを作成したい。



以下の問に答えなさい。

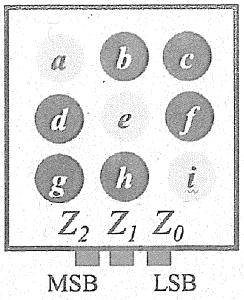
1. 電子サイコロのディジタルスイッチのボタンを押している間（ON）は、現在表示中の目からスタートして、下図のように高速に表示が変化し、ボタンを放すと、現在の目で停止するようにカウンタ回路を設計したい。



以下の真理値表を完成させなさい。

入力				出力		
Reset (ボタン)	X_2(MSB)	X_1	X_0(LSB)	Y_2(MSB)	Y_1	Y_0(LSB)
0	0	0	0	−	−	−
0	0	0	1	0	0	1
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1	−	−	−
1	0	0	0	−	−	−
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1	−	−	−

2. 次に、3×3 ドットマトリクス LED を用いたサイコロ表示回路を考える。LED 点灯を 1、消灯を 0 とした時、以下の真理値表を完成させなさい。なお、LED には以下のような記号をつけておく。



入力			出力								
Z_2(MSB)	Z_1	Z_0(LSB)	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1									
0	1	0									
0	1	1									
1	0	0									
1	0	1									
1	1	0									
1	1	1	1	1	1	1	1	1	1	1	1

3. 1 のカウンタ回路または 2 のサイコロ表示回路の中から一つ選び、真理値表から各出力端子または各 LED のカルノー図を作成しなさい。カルノー図は、原則として以下のフォーマットとする。

《カウンタ回路の場合》

出力 Y_k :

R X_2 ＼ X_1 X_0	0 0	0 1	1 1	1 0
0 0				
0 1				
1 1				
1 0				

《サイコロ表示回路の場合》

LED *p* :

Z_2 ＼ Z_1 Z_0	0 0	0 1	1 1	1 0
0				
1				

4. 3 のカルノー図をもとに、各出力端子または各 LED の論理式を構成しなさい。論理式には、AND は「∧」を、OR は「∨」を、NOT は「¬」を用いること。また、論理式は 3 のカルノー図に基づいて導出し、それ以上の簡単化はしないこと。

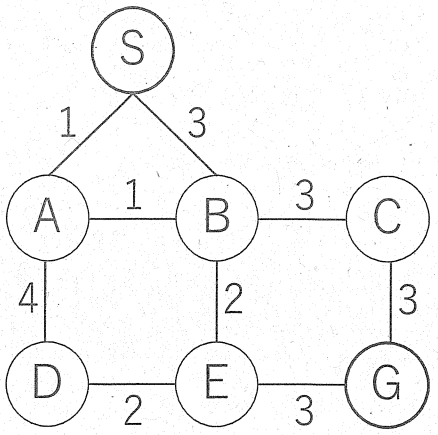
<以上。以後空白>

（このページを下書きに用いてよい）

科目名	②人工知能

数理・情報　分野

1. 図に示す無向グラフを、出発節点をS、目標節点をGとして、探索を行うことを考える。グラフの枝(エッジ)の数字は、その枝を移動するコストを表している。また、節点（ノード）の展開を行った際に、OPENリストに並べる子節点（ノード）の順序が一意に定まらない場合には、アルファベット順に並べるものとする。この時、以下に示すそれぞれのアルゴリズムでこのグラフの探索を行い、探索終了時の探索木、OPENリスト、CLOSEDリスト、および、探索の結果得られた経路とその経路のコストを示せ。



(1) 縦型探索（深さ優先探索）
(2) 横型探索（幅優先探索）
(3) ダイクストラ法（出発点からの距離が最も小さいノードを展開していく探索法）

2. 人間と対話できる、外見では区別できない二種類のロボットを開発している工場があるとする。二種類のロボットのうち、一つは常に正直である正直ロボット、もう一つは常に嘘をつく嘘つきロボットである。ある時、その中で2体のロボットaとbが、正直ロボットと嘘つきロボットのどちらであるか、わからなくなってしまった。この2体を判別するために問いかけたところ、ロボットaが、『私は嘘つきだが、ロボットbは正直である』と述べた。以下の問いに答えよ。

(1) ロボットaは正直であることを命題P、ロボットbは正直であることを命題Qとして、ロボットaの言明『私は嘘つきだが、ロボットbは正直である』を命題論理式で表現せよ。

(2) 上記(1)の結果の命題論理式に対して、真理表を作成し、その真偽を明らかにせよ。

(3) 上記(2)で求めた真理表に基づいて、二つのロボットaとbがそれぞれどちらのロボットであるかを判定するとともに、その根拠を述べよ。

（このページを下書きに用いてよい）

科目名	③ソフトウェア工学

数理・情報　　分野

注意事項：説明を求める問題に対しては，個別の指示がない限り，各設問について 100～200 字程度で具体的かつ明確に説明すること．

1　2 種類以上の利用者がいる情報システムを 1 つ挙げ，それについて以下の問いに答えよ．ただし，利用者の種類によって使用できる機能が異なる情報システムでなければならない．（8 点/問．計 32 点）

1.1　その情報システムの名称，目的および利用者の種類を説明せよ．

1.2　その情報システムが管理する必要のあるデータを ER 図で表現せよ．

1.3　その情報システムの機能を 4 つ以上挙げ，利用者の種類と機能の関係をユースケース図で示せ．

1.4　その情報システムが管理するデータと機能の関係を CRUD 図で示せ．

2　ソフトウェア工学の観点から見て良いプログラミング言語が満たすべき条件を 3 項目以上挙げ，それぞれを 50 字程度で説明せよ．（5 点/項目．計 15 点）

3　ソフトウェアの開発段階（仕様設計からテストまでの工程）で最適化を図るのは通常望ましくなく，保守段階で行うのが望ましい．その理由を説明せよ．（10 点）

4　以下のソフトウェア工学技術を活用して高品質のソフトウェアを開発するためには，どのようにする必要があるか説明せよ．（7 点/問．計 28 点）

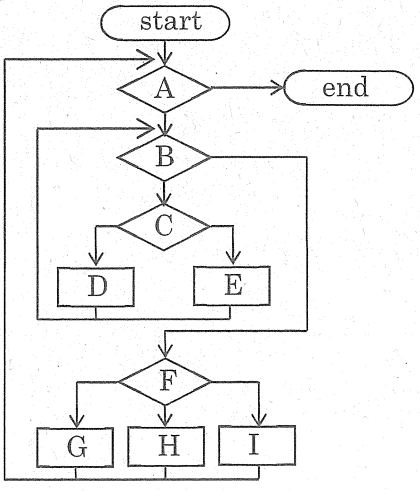
4.1　Function Point 法

4.2　プロセスモデル

4.3　デザインパターン

4.4　継続的インテグレーション

5　右のアルゴリズムについて，命令網羅，判定条件網羅，複数条件網羅（パス網羅）となるようなステップの実行系列を示せ．ただし，それぞれの場合について，実行系列の数が最小になるようにすること．（5 点/問．計 15 点）



（このページを下書きに用いてよい）

科目名	④ 確率・統計データ解析	数理・情報	分野

以下の問に答えよ．ただし，標準正規分布の上側α点を$z(\alpha)$と表したとき，$z(0.005)=2.58$, $z(0.025)=1.96$, $z(0.05)=1.65$とする．また，標準正規分布の分布関数を

$$\Phi(z)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{-\frac{1}{2}x^2}\,dx$$

と表したとき，$\Phi(2)=0.98$, $\Phi(-2)=0.02$, $\Phi(1)=0.84$, $\Phi(-1)=0.16$とする．

1. ある製品の重量は平均100g，標準偏差20gの正規分布に従うとされる．この製品を無作為に16個抽出したとき，その標本平均$\bar{X}$が95g以上105g以下となる確率を求めよ．

2. 1から10までの数字が書かれたルーレットを12回まわすとき，偶数が出る回数の期待値，分散および標準偏差を求めよ．

3. ある電子部品は工場A, B, Cの3工場で生産され，全体に占める割合はそれぞれ0.3, 0.5, 0.2である．また，各工場での不良率はそれぞれ0.01, 0.02, 0.05である．

   (a) 全体の不良品割合を求めよ．

   (b) 不良品が見つかったとき，それが工場Bで生産されたものである確率を求めよ．

4. 確率変数Xの確率密度関数を

$$f(x)=\begin{cases} k\,x(2-x) & (0\le x\le 2), \\ 0 & (\text{その他}) \end{cases}$$

   とする．

   (a) 定数kを求めよ．

   (b) 期待値$E(X)$と分散$V(X)$を求めよ．

5. ある大学で900人を無作為抽出して，オンライン授業を「好ましい」と回答した学生が450人であった．この大学全体で「好ましい」と考える学生の割合pの95%信頼区間を求めよ．さらに，信頼区間の幅を0.1未満にするために必要な最小標本サイズを求めよ．

6. あるメーカーの飲料水用充填機の説明書には，平均充填量が500mlと記載されている．無作為に36本を抽出したところ，標本平均は$\bar{x}=504$mlであった．このとき，標本平均が500mlと異なるといえるか，有意水準5%で検定せよ．ただし，充填機の母標準偏差は既知で$\sigma=12$mlとする．

「解答始め」の合図があるまで問題を見てはならない。