筑波大学　情報学群　情報科学類・情報メディア創成学類

令和３年度　学群編入学試験

学力試験問題（専門科目）

[注意事項]

1．　試験開始の合図があるまで、問題の中を見てはいけません。

2．　解答用紙と下書き用紙の定められた欄に、氏名、受験番号を記入すること。

3．　この問題冊子は全部で11ページ（表紙、白紙を除く）です。

4．　専門科目の選択について、

（ア）情報科学類と情報メディア創成学類を併願する者は、問題 1 から問題 6（数学、情報基礎、物理学）の計 6 問から 4 問を選択して答えなさい。ただし、情報メディア創成学類の合否判定においては、数学と情報基礎の解答のみを評価します。

（イ）情報科学類を単願する者は、問題１から問題６（数学、情報基礎、物理学）の計６問から４問を選択して答えなさい。

（ウ）情報メディア創成学類を単願する者は、問題１から問題４（数学、情報基礎）の計４問をすべて答えなさい。

5．　解答用紙は、専門科目で選択した４問に対して、各問１枚の合計４枚を用いること。

6．　解答用紙上部の［　　］欄に解答する問題番号を記入すること。

7．解答用紙の裏面を使用する場合には、その旨を解答用紙の表面に示してください。

問題1　数学1

(1) 次の2重積分を求めなさい.

$$\iint_D \sqrt{y^2 - x^2}\ dxdy, \qquad D = \{(x, y) \mid 0 \leqq x \leqq y \leqq 1\}$$

(2) $x + 2y + z + e^{2z} - 1 = 0$ から定まる陰関数 $z = f(x, y)$ について，以下の問いに答えなさい.

(2-1) 偏導関数 $\dfrac{\partial z}{\partial x}$, $\dfrac{\partial z}{\partial y}$, $\dfrac{\partial^2 z}{\partial x^2}$, $\dfrac{\partial^2 z}{\partial y \partial x}$, $\dfrac{\partial^2 z}{\partial y^2}$ を求めなさい.

(2-2) $z = f(x, y)$ が表す曲面上の点 $(x, y, z) = (-2, 1, 0)$ における接平面の方程式を求めなさい.

(2-3) $f(x, y)$ の原点 $(x, y) = (0, 0)$ における2変数のテイラー展開を2次の項まで求めなさい.

問題2　数学2

未知数 a, b を含む次の行列 A に関して設問 (1)-(3) に答えなさい.

$$A = \begin{pmatrix} a & 1 \\ 1 & b \end{pmatrix}$$

(1) 行列 A による一次変換で直線 $2x + 3y = 1$ が直線 $x + 4y = 3$ に写るとき，a, b の値を求めなさい.

(2) (1) の条件を満たす行列 A のすべての固有値と，各固有値に対応する長さが 1 の固有ベクトルを 1 つ求めなさい.

(3) (1) の条件を満たす行列 A による一次変換で円 $x^2 + y^2 = 1$ を写した図形の方程式を求めなさい.

問題 3　情報基礎 1

重みなし有向グラフに関する C 言語プログラム（リスト 1）について，以下の問いに答えなさい．

(1) 配列 G は二次元配列であり，頂点 i から j に辺が存在する場合は (i, j) 要素 G[i][j] に 1，それ以外の場合は 0 が格納される．この配列で表されるグラフを図示しなさい．

(2) 関数 enqueue は，キュー Q に対して要素を追加する関数であり，また dequeue は，Q から要素を取り出す関数である．プログラム中の空欄 （ア） ～ （エ） を埋めて関数を完成させなさい．

(3) 関数 shortestPath1 は引数として与えられた二頂点間の最短経路の長さを求める関数である．shortestPath1(4, 1) を呼び出したときの，関数の戻り値および関数呼び出し終了時の配列 visited の内容を書きなさい．なお配列は次の形式で答えること（例：[1, 2, 3, 4, 5]）．

(4) 関数 shortestPath2 および getPath では，配列 visited の内容を変更して二頂点間の最短経路を求めている．ここでマクロ ROOT は出発点を表す特別な値である．また関数 shortestPath2 は shortestPath1 をもとにしており，その修正箇所はコメント「/* CHANGED */」で示されている．関数 getPath は，配列 visited の内容をもとに，頂点 t までの最短経路を表示する関数である．空欄 （オ） および （カ） を埋めて関数を完成させなさい．

(5) 関数 shortestPath2(3, 0) を呼び出したときの，関数呼び出し終了時の配列 visited の内容を書きなさい．また，shortestPath2(3, 0) の呼び出しの後に関数 getPath(0) を呼び出したときの出力を書きなさい．

リスト 1　shortestPath

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdbool.h>
4 #define NON -1
5 #define G_SIZE 5
```

次ページに続く

```
#define Q_SIZE 100
#define ROOT -2
int G[G_SIZE][G_SIZE] =
    {{0,1,0,0,0},{0,0,0,0,0},{1,0,0,1,0},{0,1,0,0,1},{0,0,1,0,0}};

int visited[G_SIZE];
int Q[Q_SIZE]; int Qf, Qr;
void initQ() {
    for (int i = 0; i < Q_SIZE; i++) Q[i] = 0;
    Qf = 0; Qr = 0;
}
void enqueue(int e) {
    if (    (ア)    ) {
        fprintf(stderr, "Error: queue overflow.\n"); exit(1);
    }
    Q[    (イ)    ] = e;
};
int dequeue() {
    if (    (ウ)    ) {
        fprintf(stderr, "Error: queue underflow.\n"); exit(1);
    }
    return Q[    (エ)    ];
}
bool emptyQ() {
    if (Qf == Qr) return true; else return false;
}
int shortestPath1(int s, int t) {
    int depth = 0;
    for (int i = 0; i < G_SIZE; i++) visited[i] = NON;
    visited[s] = 0;
    initQ();
    enqueue(s);
    while (!emptyQ()) {
        int v = dequeue();
        depth = visited[v];
        if (v == t) return depth;
        for (int i = 0; i < G_SIZE; i++) {
```

次ページに続く

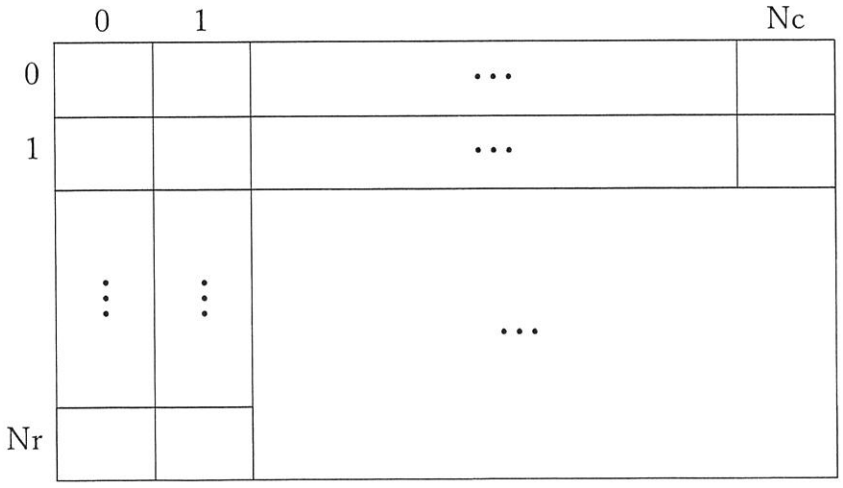
```
41              if (G[v][i] == 1 && visited[i] == NON) {
42                  visited[i] = depth + 1;
43                  enqueue(i);
44              }
45          }
46      }
47      return NON;
48  }
49  int shortestPath2(int s, int t) {
50      for (int i = 0; i < G_SIZE; i++) visited[i] = NON;
51      visited[s] = ROOT; /* CHANGED */
52      initQ();
53      enqueue(s);
54      while (!emptyQ()) {
55          int v = dequeue();
56          if (v == t) return 1; /* CHANGED */
57          for (int i = 0; i < G_SIZE; i++) {
58              if (G[v][i] == 1 && visited[i] == NON) {
59                  visited[i] = v; /* CHANGED */
60                  enqueue(i);
61              }
62          }
63      }
64      return NON;
65  }
66  void getPath(int t) {
67      initQ();
68      int v = t;
69      [   (オ)   ];
70      while (visited[v] != ROOT) {
71          [   (カ)   ];
72          enqueue(v);
73      }
74      printf("Path: ");
75      for (int i = Qr - 1; i >= 0; i--)
76          printf("%d ", Q[i]);
77      printf("\n");
78  }
```

問題4　情報基礎 2

文字列から一部の文字を，順番を変えずに抜き出して作られる文字列を部分列と呼ぶ．たとえば ABCDEF という文字列の部分列には A や CDE や BF や ACEF がある．2 つの文字列から作られる共通の部分列で最長のもの（最長共通部分列）の長さを求める C 言語プログラム P を考える．たとえば，プログラム P は ABCDEFGH と DBACAEAF という文字列を受け取ると，最長共通部分列は ACEF および BCEF であるので，4 を返す．

(1) ABCDEF と AEDCFGDF の最長共通部分列を答えなさい．

プログラム P は文字数 Nr の文字列 X と文字数 Nc の文字列 Y を受け取り，X と Y の最長共通部分列の長さを求める．プログラム P は下図に示すような縦の長さ（行数）が Nr＋1 で横の長さ（列数）が Nc＋1 である 2 次元の表を作り，この表の要素を埋める方法で答えを求める．この表の各行を上から 0 行目，1 行目，...，Nr 行目，各列を左から 0 列目，1 列目，...，Nc 列目と呼ぶ．r 行目 c 列目には，X の r 文字目までの文字列と，Y の c 文字目までの文字列の最長共通部分列の長さを格納する．したがって，この表の Nr 行目 Nc 列目の要素が X と Y の最長共通部分列の長さである．プログラム P ではこの表は 2 次元配列 mat によって表される．なお，文字列の片方または両方が空文字（長さ 0 の文字列）であるときには，それらの最長共通部分列の長さは 0 である．



次ページに続く

(2) プログラムPの先頭近くにある以下の部分では，表の0行目と0列目を0で初期化する．空欄（ア），（イ）を埋めなさい．ただし，関数strlenは文字列を指すポインタを受け取り，その文字列の長さ（文字列終端を示す文字は含まない）を返す．

```
char *str1 = （1つ目の文字列の先頭アドレス）;
char *str2 = （2つ目の文字列の先頭アドレス）;
unsigned int i, j;
unsigned int len1 = strlen(str1);
unsigned int len2 = strlen(str2);
for (i = 0; i <= len1; i++) {
  [          （ア）          ]
}
for (j = 0; j <= len2; j++) {
  [          （イ）          ]
}
```

文字列Xと文字列Yの最長共通部分列の長さは，Xの末尾の1文字を削除してできる文字列とYの末尾の1文字を削除してできる文字列の最長共通部分列の長さが既知ならば，その値を利用して求められる．

(3) プログラムPでは以下のfor文によって表の全要素を埋める．各要素は1つ上の要素，1つ左の要素，1つ左上の要素を利用して求められる．空欄（ウ），（エ），（オ），（カ）を埋めなさい．ただし，関数maxは2つの整数を受け取り，その最大値を返す．

```
for (i = 1; i <= len1; i++) {
  for (j = 1; j <= len2; j++) {
    char nextchar1 = str1[i - 1];
    char nextchar2 = str2[j - 1];
    if ( [          （ウ）          ] ) {
      mat[i][j] = [          （エ）          ];
    } else {
      mat[i][j] = max( [          （オ）          ] , [          （カ）          ] );
    }
  }
}
printf("Answer: %d¥n",mat[len1][len2]); /* 最長共通部分列の長さ */
```

次ページに続く

プログラムPを拡張して，XとYのどの部分の文字が最長共通部分列を構成するかについての情報も表示するようにした．最長共通部分列を構成する文字の組み合わせは複数存在しうるが，このプログラムはそれらのうちの1つだけを表示する．

(4)拡張されたプログラムでは以下に示すwhile文で最長共通部分列を構成する文字を求める．配列matの要素は(3)で示したプログラムによって埋められているとする．while文の実行を終えると，char型の配列res1とres2にはそれぞれ，受け取った文字列のうち最長共通部分列に含まれない文字を'-'（ハイフン）に置き換えた文字列が格納される．空欄（キ），（ク），（ケ）を埋めなさい．

なお，このwhile文では，XとYの末尾の文字から順に，その文字が最長共通部分列を構成するかどうかにしたがい，res1とres2を文字列の後ろの文字から順に埋める．表の各要素について，1つ上や1つ左の要素と同じ値であることは，その要素の場所に相当する文字が末尾に追加されても最長共通部分列が変化しないことを意味する．一方，異なる値であることは，最長共通部分列が変化することを意味する．

次ページに続く

```
unsigned int row = len1, col = len2;
while (1) {
  if (col == 0) {
    for (k = row - 1; k >= 0; k--) {
      res1[k] = '-';
    }
   break;
  }
  if (row == 0) {
    for (k = col - 1; k >= 0; k--) {
      res2[k] = '-';
    }
    break;
  }
  if (mat[row][col] == [          (キ)          ]) {
    row--;
    res1[row] = '-';
  } else if (mat[row][col] == [          (ク)          ]) {
    col--;
    res2[col] = '-';
  } else if (mat[row][col] == [          (ケ)          ]) {
    row--;
    col--;
    res1[row] = str1[row];
    res2[col] = str2[col];
  }
}
printf("Selected characters 1: %s¥n", res1);
printf("Selected characters 2: %s¥n", res2);
```

(5)受け取る文字列の長さが長くなると，このプログラムの時間計算量はどう変化するかを，Nr と Nc を用いて説明しなさい．

問題5　物理学 1

燃料を除いた本体の質量が M のロケットが地上に設置されている．このロケットには点火前に質量 m の燃料が搭載されている．その燃料に時刻 $t = 0$ で点火すると，単位時間あたりの噴射質量が μ，ロケットに対する相対速度の大きさが v の噴射が発生し，ロケットは鉛直上向きに上昇を始める．重力加速度の大きさを g とし，M, m, μ, v, g は一定とする．また，空気の抵抗は無視する．以下の問いに答えなさい．

(1)　点火後から燃料を使い切るまでの間のある時刻 t におけるロケットの加速度の大きさを M, m, μ, v, g, t で表しなさい．

(2)　(1)の時刻 t における速さを M, m, μ, v, g, t で表しなさい．

(3)　(1)の時刻 t における地上からの高さを M, m, μ, v, g, t で表しなさい．

(4)　燃料を使い切った後もロケットは上昇を続ける．ロケットの最高到達点の高さを M, m, μ, v, g で表しなさい．

(5)　最高到達点の高さをなるべく高くするためには，噴射質量 μ を大きくする方が良いのか，もしくは小さくする方が良いのかを答え，その理由を(4)の結果を用いて説明しなさい．

問題６　物理学２

次の物理単位を、MKSA 単位系（[m], [kg], [s], [A]）で答えなさい。

(1) 力　[N]
(2) エネルギー　[J]
(3) 電荷　[C]
(4) 電位　[V]
(5) 静電容量　[F]
(6) 磁束　[Wb]
(7) 磁束密度　[T]
(8) インダクタンス　[H]