① 工学研究科　2024年度

② 入試区分

工学研究科博士前期（Ⅲ期）

③ 出題科目

情報処理

④ 出題の意図

本試験問題は、情報処理の理論的理解とプログラミング実践力を総合的に評価することを目的としている。［1］では、画像データ容量や稼働率、通信時間などの計算問題を通じて、情報の量的把握力、論理的思考、システムの信頼性や通信処理に関する基礎理解を問う構成となっている。また、2進数表現や有限小数の判定問題により、情報表現の基礎知識も確認する。［2］では、条件分岐、繰返し処理、再帰呼出しなどを含むプログラム作成を通して、アルゴリズム設計力と構造的プログラミング能力を評価し、実践的な問題解決力を測定する狙いがある。

【情報処理】［1］次の問い（1～4）に答えなさい。

(1) 横 1,600 画素，縦 1,200 画素で，24 ビットのカラー情報をもつ画像が撮影できるデジタルカメラがある。このカメラに 256M バイトの記録用メモリを使用するときの記録できる最大の枚数を求めよ。ここで，画像は圧縮しないものとする。

(2) 2 進数で表現すると無限小数にならない 10 進小数を次の選択肢の中からすべて選べ。
［選択肢］　ア 0.375　　イ 0.45　　ウ 0.625　　エ 0.75

(3) 稼働率が 0.9 である機器 A と稼働率が 0.7 の機器 B が次の図のように接続されているシステムが 2 つある。システム X とシステム Y の稼働率を求めよ。ただし，並列に接続した場合は，少なくともどちらかが動作していればサービスを提供可能とする。また，稼働率は小数点以下第 3 位を四捨五入すること。

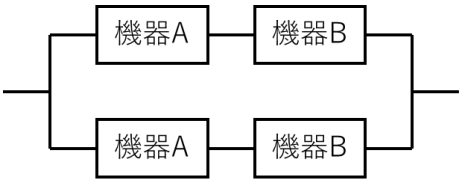

図 1 システム X

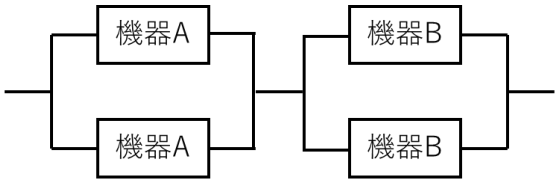

図 2 システム Y

(4) 設置場所が異なるクライアントとサーバ間で，次の条件で通信を行う場合の応答時間（秒）を求めよ。ここで，クライアントの送信処理の始まりから受信処理の終了までを応答時間とし，距離による遅延は考慮しないものとする。
［条件］

クライアントとサーバ間の回線速度	8M ビット／秒
伝送効率	60%
電文長	上り 1M バイト，下り 2M バイト
クライアントの処理時間	送信，受信合せて 0.4 秒
サーバの処理時間	送信，受信合せて 0.4 秒

【情報処理】［2］次の問い（1〜4）に示したプログラムコードを作成しなさい。ここで，解答には，Python，Java，C 言語のどれかを使用し，解答欄中に使用したプログラミング言語を明記すること。

(1) 西暦 year が閏（うるう）年かどうかを判定する関数 is_leap_year を作成しなさい。閏年の条件を次に示す。

(1) 西暦年が 4 で割り切れる年は閏年である

(2) (1)の例外として，西暦年が 100 で割り切れて 400 で割り切れない年は平年である

(2) 要素がすべて整数からなる行列 matrix（行数 r，列数 c）のすべての要素の和を求める関数 matrix_sum を作成しなさい。ただし，2 重の繰返し文を用いること。

(3) 整数 n の桁数を求める関数 count_digits を作成しなさい。ただし，n を文字列に変換してはいけません。

(4) 再帰呼出しを用いて整数 n の階乗を求める関数 factorial を作成しなさい。

＊＊＊＊＊　解答　＊＊＊＊

［1］

(1) 1 枚の画像：5.76MB　256÷5.76=44.444…　　最大 44 枚

(2) 無限小数は 0.45 のみ。従って、イ以外がすべて有限小数

(3) システム X の稼働率: 0.86、システム Y の稼働率: 0.90

(4)

① クライアントとサーバの処理時間:0.4+0.4=0.8 秒

② 上りと下りの通信時間: (1M+2M)×8÷(8M×0.6)=5 秒　（解）5.8 秒

［2］

(1) の解答例

```python
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False
```

(2) の解答例

```java
int matrix_sum(int matrix[][c], int r, int c) {
    int total_sum = 0;

    // 2重の繰り返し文を使用して行列の要素を合計する
    for (int i = 0; i < r; ++i) {
        for (int j = 0; j < c; ++j) {
            total_sum += matrix[i][j];
        }
    }

    return total_sum;
}
```

(3)の解答例

```python
def count_digits(n):
    # 整数を絶対値に変換
    n = abs(n)

    # カウンターの初期化
    count = 0

    # 10で割り続けて桁数を数える
    while n > 0:
        n = n // 10
        count += 1

    return count
```

(4)の解答例

```python
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n-1)
```