

令和9年度
第1回専攻科入学者選抜学力検査
問題用紙

生産システム工学専攻

専門科目

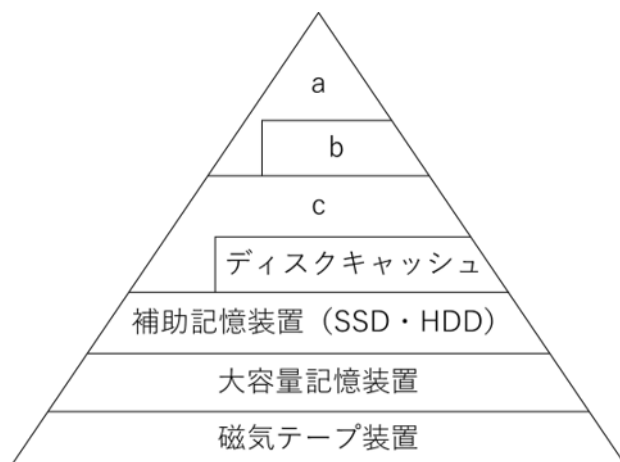
情報工学に関する専門科目

問1 情報工学基礎

※問題中で数値の基数は右下カッコの添え字に示すものとする。例えば10進数の3は $3_{(10)}$ 、この数値を2進数で表したものは $11_{(2)}$ と表す。

【1-1】

以下の図は、記憶装置をアクセス速度の速い順に表示したものである。空欄 a～c に入る最も適切な記憶装置の名称を答えよ。



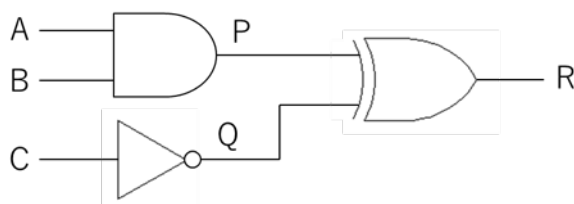
【1-2】

次の問に答えよ。

- (1) $111001_{(2)}$ を10進数に変換せよ。
- (2) $46.625_{(10)}$ を2進数に変換せよ。
- (3) $D2_{(16)}$ を2進数に変換せよ。
- (4) $10100_{(2)}$ の2の補数を求めよ。
- (5) 2の補数形式で表現された8ビットの2進数「11110100」を、右に2ビット算出シフトした結果を答えよ。
- (6) 6ビットの2進数で表現できる正負の範囲を、10進数で答えよ。

【1-3】

次の回路において、各入力 A, B, C の値がそれぞれ 0, 1, 0 のとき ($A=0$, $B=1$, $C=0$)、各出力 P, Q, R の値を答えよ。



問2 情報工学

【2-1】

以下のコードについて各問に答えよ。

```
def isPrime(n):
    """ isPrime(n)

    素数判定

    Args:
        n (int): 正の整数 (1 以上の整数)

    Returns:
        bool: n が素数の場合, True を返す.
              n が素数でない場合 (1 および合成数), False を返す.

    Raises:
        TypeError: n が正の整数でない場合,
                  "適切な引数ではありません" というメッセージの
                  TypeError オブジェクトを返す.
    """
    try:
        if type(n) is int and ア :
            if n == 1:
                return False

            range_max = int(pow(n, 0.5)) + 1
            for i in range( イ , range_max):
                if n % i == 0:
                    return ウ

            return エ
        else:
            raise TypeError("適切な引数ではありません")
    except Exception as e:
        return e

if __name__ == '__main__':
    # テストケース
    n = 99991
    ret = isPrime(n)
    print(f"{n}: {ret}")
```

- (1) 空欄ア～エに入る適切なコードを記述せよ。
- (2) isPrime(n) が True を返すテストケースを作成せよ。
- (3) isPrime(n) が False を返すテストケースを作成せよ。
- (4) True でも False でもなく、例外オブジェクトが返されるテストケースを作成せよ。
- (2) ～ (4) 共通の注意事項：解答は n の値のみでよい。ただし、n は 99991 以外とする。

【2-2】

以下のコードについて各問に答えよ。

```
def fib_dp(n):
    """ fib_dp(n)

    動的計画法でフィボナッチ数列を求める.
    初項 (第 1 項) =1, 第 2 項=1 とする.

    Args:
        n (int): 1 以上の整数

    Returns:
        フィボナッチ数列の第 n 項

    Note:
        引数の型のチェックは行わない.
    """
    if n <= 1:
        return n
    dp = [0, 1]
    for i in range(2, n+1):
        dp.append(dp[i-1] + dp[i-2])
    return dp[n]

if __name__ == '__main__':
    # テストケース
    print(fib_dp(5))    # 5
    print(fib_dp(10))   # 55
```

(1) `fib_dp(6)`を実行したとき, `for` ループが終了した直後のリスト `dp` の中身を答えよ。

(2) 動的計画法の説明として最も適切なものを選び。

- ア 問題を再帰的に分割し, 各部分問題を独立に解く手法である
- イ 一度計算した結果を保存しておき, 同じ計算を繰り返さないようにする手法である
- ウ 問題をランダムに分割し, 最も計算量が少ない解を選ぶ手法である
- エ 問題全体を一度に解くために, 大量のメモリを使用する手法である

(3) 以下の再帰実装と `fib_dp(n)`を比較したとき, `fib_dp(n)`の利点として最も適切なものを選び。

```
def fib_rec(n):
    if n <= 1:
        return n
    return fib_rec(n-1) + fib_rec(n-2)
```

- ア `fib_dp(n)`はコードが短く, 可読性が高い
- イ `fib_dp(n)`は再帰呼び出しを使用するため, スタックメモリを節約できる
- ウ `fib_dp(n)`は同じ値を重複して計算しないため, 大きな `n` に対して高速に動作する
- エ `fib_dp(n)`は浮動小数点数にも対応しており, 汎用性が高い

【2-3】

以下のコードについて各問に答えよ。scores.csv の内容および hist.png は次のページの通りである。

```
import pandas as pd
import matplotlib.pyplot as plt

def plot_histogram(filename, column, out_png):
    df = pd.read_csv(filename, encoding="utf-8-sig")
    col = df[column]

    plt.figure(figsize=(8, 6))
    plt.hist(col, bins=  , range=  )
    plt.xlabel(  )
    plt.ylabel(  )
    plt.xticks(range(0, 111, 10))
    plt.savefig(out_png)

    return {
        "min": round(float(col.min()), 6),
        "max": round(float(col.max()), 6),
        "mean": round(float(col.mean()), 6),
    }

if __name__ == '__main__':
    # テストケース
    result = plot_histogram("scores.csv", "得点", "hist.png")
    print(result)
```

- (1) 空欄ア～エに入る適切なコードまたは文字列を記述せよ。
- (2) plot_histogram()の戻り値として出力される result の内容を記述せよ。
- (3) pd.read_csv(filename, encoding="utf-8-sig") において、encoding="utf-8-sig"を指定する理由として最も適切なものを選べ。

ア CSV ファイルの読み込み速度を向上させるため

イ 日本語の文字化けを防ぐために、Shift-JIS で CSV ファイルを読み込むため

ウ pd.read_csv()のデフォルトのエンコーディングが ASCII であるため、日本語を含むファイルを読み込めるようにするため

エ Excel で CSV を保存したときに付加される BOM (バイトオーダーマーク) を自動的に除去して正しく読み込むため

(とじしろ)

番号,得点

1,72
2,45
3,88
4,63
5,91
6,55
7,78
8,34
9,67
10,82
11,59
12,95
13,48
14,71
15,60
16,85
17,39
18,76
19,53
20,100

