

[No. 1] SD カードの規格のひとつである SDXC について、簡単に説明せよ。

[No. 2] 次の Java プログラムの説明、実行結果例及びプログラムを読み、プログラムの入力値として、「10 100 406 100 122 126 210 200 500 509」を与えたときの変換後の文字列を記述せよ。

[プログラムの説明]

次の Java プログラムは、入力値として与えた数列を変換ルールに従って文字に変換するプログラムである。

[実行結果例]

数値列入力（半角スペース区切り） ⇒ 18 100 313 120 110 108  
入力された数値は 6 個です。

-----  
変換前 | 18 100 313 120 110 108  
-----

変換後 | Sanuki  
-----

[プログラム]

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class StrConvert {

    public static void main(String[] args) throws IOException {

        // 変換用 3×26 の二次元配列
        String[][] StrConvArray = {
            {"A","b","C","d","E","f","G","h","I","j",
             "K","l","M","n","O","p","Q","r","S","t","U","v","W","x","Y","z"},
            {"a","B","c","D","e","F","g","H","i","J",
             "k","L","m","N","o","P","q","R","s","T","u","V","w","X","y","Z"},
            {"1","2","3","4","5","6","7","8","9","0",
             "-",".",":",";","/"," ","_","`","~","!","@","#","$","%","&","'","(",
             ")","*","+",",","<",">","=",">"}
        };

        // 標準入力から入力を受け取る inputReader を作成
        BufferedReader inputReader = new BufferedReader(new InputStreamReader(System.in));

        System.out.print("数値列入力（半角スペース区切り） ⇒ ");
```

```

// 入力された文字列を半角スペース区切りで分割し文字列の配列に格納
String inputString = inputReader.readLine();
String[] convnumstr = inputString.split(" ",0);

int convnumCount = convnumstr.length;

// 文字列の配列から数値の配列に変換
int[] convnum = new int[convnumCount];
for ( int i = 0; i < convnumCount; i++ ){
    convnum[i] = Integer.parseInt(convnumstr[i]);
}

// 入力された数値の件数表示
System.out.print("入力された数値は ");
System.out.print(String.format("%d", convnumCount));
System.out.println("個です。");

// ソート前の数列を表示
System.out.println("");
System.out.println("-----");
System.out.print("変換前 | ");
for ( int i = 0; i < convnumCount; i++ ){
    System.out.print(String.format("%d ", convnum[i]));
}
System.out.println("");
System.out.println("-----");

// 変換結果を格納する配列
String[] convResultStr = new String[convnumCount];

// 変換処理 一文字ずつ処理
for ( int i = 0; i < convnumCount; i++ ){
    int idx_x = (convnum[i] / 100) % 3;
    int idx_y = (convnum[i] % 100) % 26;

    convResultStr[i] = StrConvArray[idx_x][idx_y];
}

// 変換後の文字列を表示
System.out.print("変換後 | ");

for ( int i = 0; i < convnumCount; i++ ){
    System.out.print(String.format("%s", convResultStr[i]));
}

System.out.println("");
System.out.println("-----");
}
}

```

[利用している演算子や関数の説明]

- 配列の添え字は0から指定する。

例： `int[] intArray = {1, 2, 3};` の場合、`intArray[0]` は 1 を示す。

- 算術演算子である「/」は、割り算の商を返す。
- 算術演算子である「%」は、割り算の余りを返す。

[No. 3] 香川県内で発生した事故と、その事故の当事者を、リレーショナルデータベースで管理している。次の条件を読んで設問に答えよ。

- 発生した事故については「事故テーブル」に登録する。
- 各事故の当事者を「事故当事者テーブル」に登録する。当事者は各事故に対し、必ず一人以上存在するものとする。
- 「発生場所種別テーブル」は、発生場所種別を示すコードと、発生場所種別が登録されているコードテーブルである。
- 「乗車区分テーブル」は、乗車区分を示すコードと、乗車区分が登録されているコードテーブルである。
- 「事故テーブル」と「事故当事者テーブル」の結合には、「nen」、「station」及び「idx」を用いる。
- それぞれの表の格納レコード（データ）の一部を下記に示す。

事故テーブル (tbl\_jiko)

列名	nen	station	idx	md	tm	place_cd
説明	発成年	発生警察署	一連番号	発生日	発生時刻	発生場所種別 CD
型	文字列	文字列	文字列	文字列	文字列	文字列
データ	2019	高松北	0001	10/1	10:00	01
	2019	高松南	0001	10/2	11:00	02
	2019	高松北	0002	11/3	15:00	02
	2020	高松北	0001	1/4	9:00	01
	2020	高松西	0001	2/4	13:00	03
	2020	丸亀	0001	2/5	14:00	03
	2020	丸亀	0002	3/3	16:00	02
	～	～	～	～	～	～

事故当事者テーブル (tbl\_tojisha)

列名	nen	station	idx	hito_idx	simei	age	kbn_cd
説明	発成年	発生警察署	一連番号	当事者番号	氏名	年齢	乗車区分 CD
型	文字列	文字列	文字列	文字列	文字列	数値	文字列
データ	2019	高松北	0001	01	香川	60	01
	2019	高松北	0001	02	高松	50	01
	2019	高松北	0001	03	徳島	40	02
	2019	高松北	0002	01	高知	25	01
	～	～	～	～	～	～	～

発生場所種別テーブル(tbl\_place)

列名	place_cd	place
説明	発生場所種別 CD	発生場所種別
型	文字列	文字列
データ	01	道路上
	02	交差点
	03	駐車場
	～	～

乗車区分テーブル(tbl\_kbn)

列名	kbn_cd	kbn
説明	乗車区分 CD	乗車区分
型	文字列	文字列
データ	01	自動車
	02	自転車
	03	徒歩
	～	～

[設問 1] 2019 年に高松北警察署で発生した事故について、事故ごとに、年齢が 50 歳以上 60 歳以下の当事者数を抽出する SQL 文を作成したい。空欄を埋め次の SQL 文を完成させよ。

```

SELECT
  A.nen, A.station, A.idx,  (*)
FROM
  tbl_jiko A, tbl_tojisha B
WHERE
  A.nen = '2019'
  AND A.nen = B.nen
  AND A.station = B.station
  AND A.idx = B.idx
  AND B.age 
GROUP BY A.nen, A.station, A.idx
    
```

[設問 2] 設問 1 の SQL 文の「GROUP BY A.nen, A.station, A.idx」の後に次の SQL 文を追加することで、当事者数が 3 人以上の事故のみを抽出する SQL 文に変更したい。追加する SQL 文の空欄を埋めよ。

>= 3

[設問3] すべての事故について、発生場所種別が「道路上」以外のデータを抽出し、事故ごとに発生場所種別(place)を表示する SQL 文を作成したい。次の SQL 文を完成させよ。

```
SELECT
  A.nen, A.station, A.idx, B.place
FROM
  [ ] ④
WHERE
  A.place_cd = B.place_cd
  AND A.[ ] ⑤ '01'
```