

1. 次のプログラムを書きなさい。

- A. 3 個の要素を持つ **String** 型配列 **str** を作成する
- B. **int** の配列の配列 **m** を作成し、{1,2,3}を要素に持つよう初期化する
- C. **int** 型の配列 **n** を宣言し{10,20,30}という要素を持つよう無名配列を代入する

【解答欄】 @10×3=30 点

A	<code>String[] str = new String[3];</code>
B	<code>int[] m = {1, 2, 3};</code>
C	<code>int[] n = new int[]{10, 20, 30};</code>

《解説》

str を作成し null で初期化するのは、配列要素をすべて null で初期化するという意味です。String 型の配列を作成すると null が初期値ですから、わざわざ代入する必要はありません。なお、str 自体に null を入れないようにしてください。

{...} のように、要素のリストを示す書き方の場合、常に [] 内は空白です。要素数を書いてはいけないことに注意してください。

2. 次の問いに答えなさい。

- A. 要素に "A" "B" "C" "D" "E" を持つ配列変数 **s** の宣言文を書きなさい。
- B. **s[2]** はなんという文字か解答欄に書きなさい。
- C. 次は、『**s[2]**と**"x"**が等しければ、**"x"**という文字列を表示する』という **if** 文です。
空欄になっている部分の条件式を解答欄に書きなさい。

```
if( ){
    System.out.println("X");
}
```

【解答欄】 @10×3=30 点

A.	<code>String[] s = {"A", "B", "C", "D", "E"};</code>
B.	<code>"C"</code>
C.	<code>s[2].equals("x")</code>

《解説》

String 型配列は要素のひとつずつが String です。
同値の比較（同じかどうか）には、基本データ型のように == ではなく
equals() メソッドを使います。配列要素が String 型なので、s[2].equals("x")
または、"X".equals(s[2]) のようにして比較します。

3. 次の配列変数の用法で正しいものをすべて答えなさい（複数ある）。

- A. `int[] n, m;`
- B. `double m=5; double[] x = new double[m];`
- C. `int[] n; n = {1,2,3,4,5};`
- D. `short[5] n = {1,2,3,4,5};`
- E. `double[] x; x=new double[5] {1.0,2.0,3.0,4.0,5.0};`
- F. `String s[] = new String[]{"a", "b", "c"};`
- G. `int []n=new int[3], m; n=m;`

【解答欄】

A, F

20 点

《解説》

B=配列要素数を `double` で指定しているが整数型しか使えない、C=初期化リストでの初期化は、宣言時しかできない、D=配列要素数を指定している、E=配列要素数を指定している、G=m は宣言しただけなので初期化されていない。そのため `n=m` とできない。

4. 次のプログラムを実行した時の結果として正しいものはどれか答えなさい。

```
String[] str;  
str[0] = "OK";
```

- A. `null` が表示される
- B. 何も表示されない
- C. コンパイルエラー
- D. 実行時例外が発生する

【解答欄】

C

10 点

《解説》

`str` は宣言しただけで初期化されていない（値がない）ので、`str[0]` のように使うことはできない。コンパイルエラーになる。

5. 次のプログラムを実行したときの結果として正しいものはどれか答えなさい。

```
int[] val={1,2,3,4,5}, data=null;  
data[0] = val[0];
```

- A. 12345
- B. 00000
- C. なにも表示されない
- D. コンパイルエラーになる
- E. 実行時例外になる

【解答欄】

E

10 点

《解説》

`data` には `null` が入っているので、`data[0]` のように書いてもコンパイルエラーにはならないが、実際に使ってしまうと、実行した時に実行時例外が発生して停止する。