

Question 1: What is the output of the following program?

```
#include <cstdio>
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s1, s2;
    const char *s;

    s2 = "Fred";
    s = s2.c_str();
    s1 = s;

    s1[1] = 'y';
    s2[1] = 'z';

    cout << s << " "
         << s1 << " "
         << s2 << endl;
    return 0;
}
```

Multiple Choice Answers:

In TurningPoint, simply put the letter of the answer.

(They are in alphabetical order, BTW)

- A. Fred Fyed Fyed
- B. Fred Fyed Fzed
- C. Fred Fzed Fyed
- D. Fred Fzed Fzed
- E. Fyed Fyed Fyed
- F. Fyed Fyed Fzed
- G. Fyed Fzed Fyed
- H. Fyed Fzed Fzed
- I. Fzed Fyed Fyed
- J. Fzed Fyed Fzed
- K. Fzed Fzed Fyed
- L. Fzed Fzed Fzed

Question 2: What is the output of the following program? (Note, it will be a single word composed of numbers and hyphens).

```
#include <cstdio>
#include <iostream>
using namespace std;

int main()
{
    int x, y;
    string s;

    x = 5;
    y = 10;
    s = "15 Fred";

    sscanf(s.c_str(), "%d %d", &x, &y);
    cout << x << "-" << y << "-";

    s = "Fred 20";

    sscanf(s.c_str(), "%d %d", &x, &y);
    cout << x << "-" << y << endl;

    return 0;
}
```

Answers to clicker questions

Question 1

The answer is J:

```
Fzed Fyed Fzed
```

Why? You start with **s2** being "Fred", and **s** pointing to the underlying string:

```
s2: "Fred"
    ^
    |
s ---
```

When you assign **s1** to **s**, it makes a copy -- C++ strings always make copies:

```
s2: "Fred"
    ^
    |
s ---
s1: "Fred"
```

Now, each string changes its second character -- **s1** changes it to 'y' and **s2** changes it to 'z':

```
s2: "Fzed"
    ^
    |
s ---
s1: "Fyed"
```

Thus, the output is "Fzed Fyed Fzed".

Question 2

The answer is:

```
15-10-15-10
```

Explanation: The first **sscanf()** correctly reads 15 into **x**, but fails reading **y**, because "Fred" is not an integer. So **y** remains at 10.

The second **sscanf()** fails reading **x**, and the **sscanf()** call exits at that point, so both **x** and **y** remain unchanged.