DRY RUNS PRACTICE

Qno 1:

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

int dry(int);

int main()
{
	int b=10, result;



	result = dry(b);
	cout << result;

	return 0;
}

int dry(int n)
{
		if (n<2)
			return n;
		else
			return dry(n - 2) + dry(n - 1);


}
```

Qno 2:

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

int dry(int, int);

int main()
{
	int b = 5, k = 2, result;



	result = dry(b, k);
	cout << result;

	return 0;
}

int dry(int b, int p)
{
	if (p != 0)
		return (b *= dry(b, p - 1));
	else
		return 1;
}
```

Qno 3:

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

int dryrun(int n1, int n2);

int main()
{
	int n1 = 5, n2 = 20;

	cout << "OUTPUT " << dryrun(n1, n2);

	return 0;
}

int dryrun(int n1, int n2)
{
	if (n2 != 0)
		return dryrun(n2, n1 % n2);
	else
		return n1;
}
```

Qno 4:

```
#include<iostream>
using namespace std;
int main()
{
	float arr[] = { 12.4,2.3,4.5,6.7 };
	cout << sizeof(arr) / sizeof(arr[0]);
	return 0;
}
```

Qno 5:

What are the values assigned to these index(arr[0], arr[1], arr[2], arr[3]) of the following code:

```
#include<iostream>
using namespace std;
int main()
{

	int arr[4];
	for (int i = 0; i<4; i++)
		if (i % 2 == 0)
			arr[i] = 2 * i;
		else
			arr[i] = i;

	return 0;
}
```

Qno 6:

```
#include<iostream>
using namespace std;
int main()
{
	int a[5] = { 5,1,15,20,25 };
	int x, y, z;
	x = ++a[1];
	y = a[1]++;
	z = a[x++];
	cout << x << "  " << y << "  " << z;

	return 0;
}
```