

1. What does the following print? (3 points each)

```
int a[5] = {10, 20, 30, 40, 50};
int *p = a+2;
```

	ANSWERS
cout << a[1];	20
cout << p[1];	40
cout << p - a;	2
cout << *p - *a;	20
cout << ++*++p;	41

2. Write a function *negate* that takes a `vector<double>` by reference and replaces each element with its negative. It might be called like this: (25 points)

```
vector<double> v;
v.push_back(2.5);
v.push_back(-4);
negate(v);
cout << v[0] << ", " << v[1]; // -2.5,4
```

```
// ANSWER
void negate(vector<double>& v)
{
    for (int i=0; i<int(v.size()); ++i)
        v[i] = -v[i];
}
```

3. Write the same function except that it is called by passing the begin and end iterators, like this: (25 points)

```
negate(v.begin(), v.end());
```

```
// ANSWER
void negate(vector<double>::iterator b,
            vector<double>::iterator e)
{
    while (b != e)
    {
        *b = -*b;
        ++b;
    }
}
```

4. Write a program that reads a list of words separated by white space until end of file. The output is a list of all words equal in length to the longest word. In the following example, user input is shown in **bold**. (35 points)

```
this is a test
^Z
this
test
```

```
// ANSWER
#include <iostream>
#include <vector>
#include <string>
using namespace std;

int main()
{
    // Read list of words into v
    // and find longest word
    vector<string> v;
    string s;
    int longest = 0;
    while (cin >> s)
    {
        v.push_back(s);
        if (int(s.size()) > longest)
            longest = int(s.size());
    }

    // Print longest words
    for (int i=0; i<int(v.size()); ++i)
        if (int(v[i].size()) == longest)
            cout << v[i] << "\n";

    return 0;
}
```